Globally convergent Newton-type methods for multiobjective optimization

M. L. N. Gonçalves * F. S. Lima * L. F. Prudente *

February 10, 2021

Abstract: We propose two Newton-type methods for solving (possibly) nonconvex unconstrained multiobjective optimization problems. The first is directly inspired by the Newton method designed to solve convex problems, whereas the second uses second-order information of the objective functions with ingredients of the steepest descent method. One of the key points of our approaches is to impose some safeguard strategies on the search directions. These strategies are associated to the conditions that prevent, at each iteration, the search direction to be *too close* to orthogonality with the multiobjective steepest descent direction and require a proportionality between the lengths of such directions. In order to fulfill the demanded safeguard conditions on the search directions of Newton-type methods, we adopt the technique in which the Hessians are modified, if necessary, by adding multiples of the identity. For our first Newton-type method, it is also shown that, under convexity assumptions, the local superlinear rate of convergence (or quadratic, in the case where the Hessians of the objectives are Lipschitz continuous) to a local efficient point of the given problem is recovered. The global convergences of the aforementioned methods are based, first, on presenting and establishing the global convergence of a general algorithm and, then, showing that the new methods fall in this general algorithm. Numerical experiments illustrating the practical advantages of the proposed Newton-type schemes are presented.

Keywords: Multiple objective programming, Newton method, global convergence, numerical experiments

AMS subject classifications: 49M15, 65K05, 90C26, 90C29

1 Introduction

Multiobjective optimization is concerned with problems in which some objective functions have to be minimized simultaneously. In proper problems of this setting, the objective functions are conflicting, meaning that no single point minimizes all objectives at once. Hence, the concept of optimality has to be replaced by the concept of *Pareto optimality* or *efficiency*. A point is called *Pareto optimal* if there does not exist a different point with smaller or equal objective function

*Instituto de Matemática e Estatística, Universidade Federal de Goiás, CEP 74001-970 - Goiânia, GO, Brazil, E-mails: maxlng@ufg.br, fernandoslima@discente.ufg.br, lfprudente@ufg.br. This work was funded by FAPEG (Grants PRONEM-201710267000532, PPP03/15-201810267001725) and CNPq (Grants 302666/2017-6, 408123/2018-4, 424860/2018-0).

values, such that there is a decrease in at least one objective. In other words, it is impossible to improve one objective without going worse in another.

In the last two decades, the extension of numerical methods of scalar-valued to multiobjective-valued optimization has been the subject of intense research. Examples of methods in this direction include: steepest descent [13,19], conditional gradient [3], Newton [12,41], quasi-Newton [2, 35], conjugate gradient [16, 27], projected gradient [14, 17], and proximal methods [5]. As main characteristics, these methods are based on some convergence theory and do not transform the problem at hand into a parameterized scalar problem and then solve it, being attractive alternatives to scalarization [15] and heuristic approaches [24]. Usually, in practical problems, there exists a (possibly infinite) number of Pareto optimal points which, at first, are considered equally good. In order to allow the decision maker to carry out the best (subjective) choice, it is important to formulate methods capable of finding a representative set of the *Pareto frontier* or, equivalently, of all efficient points. To estimate the Pareto frontier of a multiobjective optimization problem, a *multi-start* strategy is often adopted: we run the algorithm from several different starting points and collect the efficient points found. Therefore, in view of this application, the property of global convergence is highly desirable.

The Newton method for multiobjective optimization was originally proposed in [12]. In the latter reference, using convexity assumptions on the objective functions, the authors showed that the main characteristics of the scalar Newton method are preserved, i.e., the method is capable of achieving local superlinear convergence rate (or quadratic, in the case where the Hessians of the objective functions are Lipschitz continuous) to Pareto optimal points. Such convexity assumptions are necessary to ensure that the Hessians of the objective functions are positive definite and hence that the search directions of the Newton method are well defined and yield descent. In [41], under similar assumptions, the authors studied the Newton method using the majorizing function technique. These works substantially cover the convergence theory of the Newton method for convex multiobjective problems. However, its applicability to general multiobjective problems remains open. Therefore, the main intent of the present work is to fill this open research topic, i.e., to propose and study globally convergent Newton-type methods for solving (possibly) nonconvex multiobjective optimization problems.

Let us now recall the limitations of the Newton-type methods when applied to nonconvex scalar problems. In [30], by means of examples, it was shown that the Newton method can diverge when applied to an unconstrained scalar-valued problem with the following characteristics: the objective function is strongly convex along each search direction (although it is not by itself), the level sets of the objective functions are compact, and the line searches are exact. Similar results related to the quasi-Newton BFGS method were obtained in [7,8,29,30]. These remarkable works show that some safeguard strategies must be imposed on the directions of Newton-type methods to obtain global convergence in the nonconvex case. A well-known safeguard strategy in the scalar setting imposes two conditions on the search directions, see [4]. The primary condition requires that the search directions are never *too close* to orthogonality with the gradient of the objective function. In other words, the angle between each search direction and the steepest descent direction must be less than and bounded away from $\pi/2$. While the second condition demands that the length of a search direction should be proportional to the length of the gradient of the objective function. Here, we propose a corresponding safeguard strategy for multiojective optimization problems which, in particular, prevents, at each iteration, the search direction and the multiobjective steepest descent direction from becoming *almost* orthogonal and requires a proportionality between the lengths of such directions. Mimicking the scalar case (see [4]), the proposed safeguard strategy, combined with a nonmonotone Armijo-like line search, generates

globally convergent methods for nonconvex multiobjective optimization problems. Nonmonotone line searches algorithms in the multiobjective setting were previously considered, for example, in [11, 32, 37].

We propose two Newton-type methods that deal with nonconvex multiobjective problems. The first one (called, *Newton method with safeguarded directions*) is directly inspired by the Newton method designed to solve convex problems [12]. We adopt, based on the scalar case, the technique in which the Hessians of the objective functions are modified, if necessary, by adding multiples of the identity to make them positive definite and to ensure that the corresponding search directions satisfy the proposed safeguard strategy. Under local convexity assumptions, we prove that the proposed scheme with suitable algorithmic parameters is reduced to the algorithm of [12], enjoying its local convergence properties. This means that we obtain a globally convergent version of the Newton method and that, under suitable conditions, the rate of convergence is superlinear (or quadratic, in the case where the Hessians of the objectives are Lipschitz continuous). A possible drawback of this method is the need to guarantee the positiveness of the Hessians approximations of all objective functions. In principle, this makes the computational cost of an iteration proportional to the number of objectives. Thus, we propose a second method (called, *Newton-Gradient algorithm with safeguarded directions*) that requires the positiveness of only one matrix per iteration. The latter fact turns this second method an attractive alternative to deal with problems that have many objectives. As the name suggests, this method uses second-order information of the objective functions with ingredients of the steepest descent method [13]. Similarly to the first method, we also use the technique of modifying the Hessians of the objective functions, if necessary, in order to fulfill the requirements on the search directions. The global convergence analyses of the aforementioned Newton-type methods are based, first, on presenting and establishing the global convergence of a general algorithm and, then, showing that the new methods fall in this general algorithm. Numerical experiments illustrating the effectiveness of the introduced globalization techniques are presented and comparisons of the proposed Newton-type methods with the classic Newton and steepest descent methods are provided.

This paper is organized as follows. Section 2 presents the problem of interest in this paper as well as some definitions and basic results. Section 3 introduces a general algorithm for solving multiobjective optimization problems and presents its global convergence. The search directions for this algorithm are deliberately left open, we only require that they satisfy some safeguard conditions. Section 4 presents and analyzes two proposed Newton-type methods. In particular, it is shown that the methods fit the general algorithm of section 3 and, as a consequence, their global convergences are obtained. We also discuss some aspects with respect to local convergence properties. Numerical experiments are presented in section 5 and some final remarks are provided in section 6.

2 Preliminaries

Denote by $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ the set of real numbers, the set of nonnegative real numbers, and the set of positive real numbers, respectively. Moreover, $\mathbb{R}^n$ and $\mathbb{R}^{n\times p}$ stand for the set of n dimensional real column vectors and the set of $n \times p$ real matrices, respectively. Given two matrices $A, B \in \mathbb{R}^{n\times n}$, $A \preceq B$ means that $B - A$ is positive semidefinite. For any vectors $x, y \in \mathbb{R}^n$, $\langle x, y\rangle$ denotes their usual inner product and $\|x\| := \sqrt{\langle x, x\rangle}$ denotes the Euclidean norm of x. We denote by $B[x, \delta]$ the closed ball of radius δ with center $x \in \mathbb{R}^n$. The cardinality of a set C is denoted by $|C|$. If $K = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$, with $k_j < k_{j+1}$ for all $j \in \mathbb{N}$, then we

denote $K \subset \mathbb{N}$.
∞

In this paper, we are interested in the following unconstrained multiobjective optimization problem:

$$\min_{x \in \mathbb{R}^n} F(x), \tag{1}$$

where $F\colon \mathbb{R}^n \to \mathbb{R}^m$ is twice continuously differentiable. In the multiobjective optimization setting, the concept of optimality is replaced by the concept of *Pareto-optimality* or *efficiency*. A point $x^* \in \mathbb{R}^n$ is called *Pareto optimal* or *efficient* if and only if there is no $x \in \mathbb{R}^n$ such that $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$, where the inequality sign $\leq$ between vectors is to be understood in a componentwise sense. In its turn, a point $x^* \in \mathbb{R}^n$ is called *weakly Pareto optimal* or *weakly efficient* if and only if there is no $x \in \mathbb{R}^n$ such that $F(x) < F(x^*)$. It is said that $x^* \in \mathbb{R}^n$ is a *local Pareto optimal* (resp. *local weak Pareto optimal*) if there exists a neighborhood $V \subset \mathbb{R}^n$ of x^* such that the point x^* is Pareto optimal (resp. weak Pareto optimal) for F restricted to V. A necessary condition for local Pareto-optimality of $x^* \in \mathbb{R}^n$ is

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

where $JF(x) \in \mathbb{R}^{m \times n}$ denotes the Jacobian of F at x and $\text{Image}(JF(x))$ is the image set of $JF(x)$. A point $x^* \in \mathbb{R}^n$ that satisfies condition (2) is called *Pareto critical* or *stationary*. Therefore, if x is not Pareto critical, there exists $v \in \mathbb{R}^n$ such that $JF(x)v \in -(\mathbb{R}^m_{++})$. Every such vector v is a *descent direction* for F at x, i.e., there exists $\varepsilon > 0$ such that $F(x + tv) < F(x)$ for any $t \in]0, \varepsilon]$, see [26]. Given $x, y \in \mathbb{R}^n$, it is said that x dominates y when $F(y) - F(x) \in \mathbb{R}^m_+ \backslash \{0\}$. The function $F\colon \mathbb{R}^n \to \mathbb{R}^m$ is said to be convex (resp. strongly convex) if its components $F_j\colon \mathbb{R}^n \to \mathbb{R}$ are convex (resp. strongly convex), for all $j = 1, \ldots, m$.

Define $f\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$f(x, d) := \max_{j=1,\ldots,m} \langle \nabla F_j(x), d \rangle. \tag{3}$$

Function f gives a characterization of descent directions of F at x, because d is a descent direction for F at x if and only if $f(x, d) < 0$, see [13, 19]. The following result gives some other useful properties of f.

Lemma 1. *Let $F\colon \mathbb{R}^n \to \mathbb{R}^m$ be a continuously differentiable function and consider $f\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as in (3). Then, the following statements hold:*

(a) for any $x \in \mathbb{R}^n$ and $\alpha \geq 0$, we have $f(x, \alpha d) = \alpha f(x, d)$;

(b) the mapping $(x, d) \mapsto f(x, d)$ is continuous.

Proof. Item (a) follows trivially from the definition of f in (3) and, for the proof of item (b), see [19]. □

We next recall the extensions of the steepest descent and the Newton-type directions for multiobjective optimization. We refer to [12,13,18,19,35,39] for further reading on these subjects.

2.1 The Multiobjective Steepest Descent Direction

For a given point $x \in \mathbb{R}^n$, consider the scalar-valued problem:

$$\min_{d \in \mathbb{R}^n} f(x, d) + \frac{1}{2}\|d\|^2. \tag{4}$$

Since $f(x,\cdot)$ is a real closed convex function, it follows that (4) has always a unique optimal solution. Denote by $d_{SD}(x)$ the solution of (4) and by $\theta_{SD}(x)$ its optimal value, i.e.,

$$d_{SD}(x) := \arg\min\left\{ f(x,d) + \frac{\|d\|^2}{2} \mid d \in \mathbb{R}^n \right\}, \tag{5}$$

and

$$\theta_{SD}(x) := f(x, d_{SD}(x)) + \frac{1}{2}\|d_{SD}(x)\|^2. \tag{6}$$

Direction $d_{SD}(x)$ extends the notion of the steepest descent direction to the multiobjective optimization case. Note that in the single-objective minimization case where $F\colon \mathbb{R}^n \to \mathbb{R}$, we obtain $f(x,d) = \langle \nabla F(x), d\rangle$, $d_{SD}(x) = -\nabla F(x)$ and $\theta_{SD}(x) = -\|\nabla F(x)\|^2/2$. As is well-known, the direction $d_{SD}(x)$ and the optimum value $\theta_{SD}(x)$ can be used, in particular, to characterize stationary points of (1).

Lemma 2. *Let $d_{SD} : \mathbb{R}^n \to \mathbb{R}^n$ and $\theta_{SD} : \mathbb{R}^n \to \mathbb{R}$ be given by (5) and (6), respectively. Then, we have:*

(a) if x is Pareto critical, then $d_{SD}(x) = 0$ and $\theta_{SD}(x) = 0$;

(b) if x is not Pareto critical, then $d_{SD}(x) \neq 0$, $\theta_{SD}(x) < 0$, $f(x, d_{SD}(x)) < -(1/2)\|d_{SD}(x)\|^2 < 0$, and $d_{SD}(x)$ is a descent direction for F at x;

(c) the mappings $d_{SD}(\cdot)$ and $\theta_{SD}(\cdot)$ are continuous.

Proof. See [19, Lemma 3.3]. □

Problem (4) can be reformulated as

$$\begin{array}{ll} \min\limits_{(t,d)\in\mathbb{R}\times\mathbb{R}^n} & t + \frac{1}{2}\|d\|^2 \\ \text{s. t.} & \langle \nabla F_j(x), d\rangle \leq t, \quad \forall j = 1,\ldots,m, \end{array} \tag{7}$$

which is a convex quadratic problem with linear inequality constraints. Since problem (7) has a unique solution $(f(x, d_{SD}(x)), d_{SD}(x))$ and its constraints are linear, there exists a multiplier $\lambda^{SD}(x) \in \mathbb{R}^m$ such that the triple $(t, d, \lambda) := (f(x, d_{SD}(x)), d_{SD}(x), \lambda^{SD}(x)) \in \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^m$ satisfies the Karush-Kuhn-Tucker conditions of problem (7) given by:

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

$$\lambda_j \geq 0, \quad \langle \nabla F_j(x), d\rangle \leq t, \quad \lambda_j \left[\langle \nabla F_j(x), d\rangle - t\right] = 0, \quad \forall j = 1,\ldots,m.$$

Therefore, in particular, we have

$$d_{SD}(x) = -\sum_{j=1}^m \lambda_j^{SD}(x) \nabla F_j(x), \tag{8}$$

$$\sum_{j=1}^m \lambda_j^{SD}(x) = 1, \quad \lambda_j^{SD}(x) \geq 0, \quad \forall j = 1,\ldots,m, \tag{9}$$

$$\theta_{SD}(x) = -\frac{1}{2}\|d_{SD}(x)\|^2, \tag{10}$$

and

$$f(x, d_{SD}(x)) = -\|d_{SD}(x)\|^2. \tag{11}$$

We end this section by stating a useful result whose proof can be found in [39, Corollary 2.3].

Lemma 3. *For any $x \in \mathbb{R}^n$, $-d_{SD}(x)$ is the minimal norm element of the set*

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

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

2.2 The Multiobjective Newton-type Direction

For a given point $x \in \mathbb{R}^n$, let $B_j(x)$ be some approximation of $\nabla^2 F_j(x)$, $j = 1, \ldots, m$. Assume that $B_j(x)$, $j = 1, \ldots, m$, is a positive definite matrix and consider the scalar-valued problem:

$$\min_{d \in \mathbb{R}^n} \max_{j=1,\ldots,m} \langle \nabla F_j(x), d \rangle + \frac{1}{2} \langle B_j(x) d, d \rangle. \tag{12}$$

Since the objective function of (12) is strongly convex, this problem always has a unique solution. The Newton-type direction $d_N(x)$ is defined to be the solution of problem (12) and its optimum value will be denoted by $\theta_N(x)$, i.e.,

$$d_N(x) := \arg\min_{d \in \mathbb{R}^n} \max_{j=1,\ldots,m} \langle \nabla F_j(x), d \rangle + \frac{1}{2} \langle B_j(x) d, d \rangle, \tag{13}$$

and

$$\theta_N(x) := \max_{j=1,\ldots,m} \langle \nabla F_j(x), d_N(x) \rangle + \frac{1}{2} \langle B_j(x) d_N(x), d_N(x) \rangle. \tag{14}$$

Direction $d_N(x)$ and the optimum value $\theta_N(x)$ can also be used to characterize Pareto critical points of problem (1), as stated in the next lemma.

Lemma 4. *Let $d_N : \mathbb{R}^n \to \mathbb{R}^n$ and $\theta_N : \mathbb{R}^n \to \mathbb{R}$ given by (13) and (14), respectively. Then, we have:*

(a) if x is Pareto critical, then $d_N(x) = 0$ and $\theta_N(x) = 0$;

(b) if x is not Pareto critical, then $d_N(x) \neq 0$, $f(x, d_N(x)) < \theta_N(x) < 0$, and $d_N(x)$ is a descent direction for F at x;

(c) the mapping $\theta_N(\cdot)$ is continuous.

Proof. The proof follows the same ideas of the ones in [12, Lemma 3.2] and [18, Proposition 4.1] by replacing the $\nabla^2 F_j(x)$ by $B_j(x)$ for every $j = 1, \ldots, m$. See also [35, Lemma 2]. □

Problem (13) is equivalent to

$$\begin{aligned} \min_{(t,d) \in \mathbb{R} \times \mathbb{R}^n} \quad & t \\ \text{s. t.} \quad & \langle \nabla F_j(x), d \rangle + \frac{1}{2} \langle B_j(x) d, d \rangle \leq t, \quad \forall j = 1, \ldots, m, \end{aligned} \tag{15}$$

which is a smooth optimization problem with linear objective function and quadratic constraints. The unique solution of (15) is $(t,d) := (\theta_N(x), d_N(x))$. Moreover, this problem is convex and has a Slater point (for example, $(1,0) \in \mathbb{R} \times \mathbb{R}^n$). Thus, it follows that there exists a multiplier $\lambda^N(x) \in \mathbb{R}^m$ such that the triple $(t,d,\lambda) := (\theta_N(x), d_N(x), \lambda^N(x)) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ satisfies the Karush-Kuhn-Tucker conditions of (15) given by:

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

and, for all $j = 1, \ldots, m$,

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

Hence, we obtain

$$d_N(x) = -\left[\sum_{j=1}^{m} \lambda_j^N(x) B_j(x)\right]^{-1} \sum_{j=1}^{m} \lambda_j^N(x) \nabla F_j(x), \tag{16}$$

$$\sum_{j=1}^{m} \lambda_j^N(x) = 1, \quad \lambda_j^N(x) \geq 0, \quad \forall j = 1, \ldots, m, \tag{17}$$

and

$$\theta_N(x) = -\frac{1}{2} d_N(x)^T \left[\sum_{j=1}^{m} \lambda_j^N(x) B_j(x)\right] d_N(x). \tag{18}$$

We end this section by noting that when $B_j(x)$ is taken as $\nabla^2 F_j(x)$ for all $j = 1, \ldots, m$, then the direction $d_N(x)$ in (13) corresponds to the multiobjective Newton direction, see [12].

3 A Globally Convergent General Algorithm for Multiobjective Optimization

This section describes a general algorithm for solving (1) and presents its global convergence analysis. The algorithm imposes a safeguard strategy on the search directions which, combined with an average-type nonmonotone line search, turns it a globalized scheme. As reported in [11, 32], nonmonotone line searches are also efficient in the multiobjective setting.

Algorithm 1. *(A General Algorithm for Multiobjective Optimization)*

Step 0. *Let* $x^0 \in \mathbb{R}^n$, $\Gamma_1 \in]0,1]$, $\Gamma_2 > 0$, $\sigma \in]0,1[$, $0 < \omega_1 < \omega_2 < 1$, *and* $\eta \in [0,1[$ *be given. Set* $C^0 = F(x^0)$, $q_0 = 1$, *and initialize* $k \leftarrow 0$.

Step 1. *Compute* $d(x^k) \in \mathbb{R}^n$ *satisfying*

$$f(x^k, d(x^k)) \leq -\Gamma_1 \|d_{SD}(x^k)\| \|d(x^k)\|, \tag{19}$$

$$\|d(x^k)\| \geq \Gamma_2 \|d_{SD}(x^k)\|, \tag{20}$$

where $d_{SD}(x^k)$ *is as in* (5).

Step 2. *If x^k is Pareto critical, then* **stop**.

Step 3. *Set $t_k = 1$.*

Step 3.1 *If*

$$F_j(x^k + t_k d(x^k)) \leq C_j^k + \sigma t_k f(x^k, d(x^k)), \quad \forall j = 1, \ldots, m, \tag{21}$$

then set $x^{k+1} = x^k + t_k d(x^k)$ and go to Step 4.

Step 3.2 *Find $t_{\text{trial}} \in [\omega_1 t_k, \omega_2 t_k]$, set $t_k \leftarrow t_{\text{trial}}$, and go to Step 3.1.*

Step 4. *Updade q_{k+1} and C^{k+1} as follows:*

$$q_{k+1} = \eta q_k + 1 \quad \text{and} \quad C^{k+1} = \frac{\eta q_k}{q_{k+1}} C^k + \frac{1}{q_{k+1}} F(x^{k+1}). \tag{22}$$

Step 5. *Set $k \leftarrow k + 1$, and go to Step 1.*

Condition (19) implies that $d(x^k)$ is a descent direction for F at x^k. Moreover, by (8) and (9), we have $-\langle d_{SD}(x^k), d(x^k)\rangle \leq f(x^k, d(x^k))$, which, together with (19), gives

$$-\langle d_{SD}(x^k), d(x^k)\rangle \leq -\Gamma_1 \|d_{SD}(x^k)\| \|d(x^k)\|.$$

This means that condition (19) also implies that the angle between $d(x^k)$ and the steepest descent direction $d_{SD}(x^k)$ is smaller or equal to a fixed angle smaller than $\pi/2$, defined by the parameter Γ_1. In its turn, condition (20) essentially says that the length of $d(x^k)$ should be proportional to the length of $d_{SD}(x^k)$. As will be seen, the safeguard conditions (19) and (20) will play a key role in the convergence analysis of Algorithm 1. The way to obtain $d(x^k)$ satisfying (19) and (20) will depend on the particular instance of the method. For example, in the steepest descent method, we trivially have that $d(x^k) := d_{SD}(x^k)$ satisfies such conditions with $\Gamma_1 = \Gamma_2 = 1$. In section 4, we will present some Newton-type methods which can be seen as instances of Algorithm 1. In particular, it will be specified how the search direction $d(x^k)$ at Step 1 can be obtained. Concerning the line search, the step size t_k must be such that it satisfies the nonmonotone Armijo-like condition (21). The algorithmic parameter η controls the degree of nonmonotonicity in C^k. If $\eta = 0$, we obtain $C^k = F(x^k)$ and, as a consequence, the nonmonotone line search reduces to the monotone one. For $\eta = 1$, it turns out that C^k corresponds to the average of function values $F(x^0), \ldots, F(x^k)$. Parameters ω_1 and ω_2 are related to the backtracking strategy used in Step 3. If a step size t fails to satisfy (21), a new smaller trial step size t_{trial} is chosen such that $t_{\text{trial}} \in [\omega_1 t, \omega_2 t]$. This safeguarded choice allows the inner loop at Step 3 to have a finite termination and prevents the successful step size t_k from being artificially small.

We next recall a useful property about the vector C^k defined in (22), whose proof can be found in [32, Lemmas 6].

Lemma 5. *For each iteration k of Algorithm 1, we have $F(x^k) \leq C^k \leq A_k$, where $A_k = \frac{1}{k+1} \sum_{i=0}^{k} F(x^i)$.*

The following theorem establishes the well-definiteness of Algorithm 1, meaning that if the algorithm does not stop in iteration k at Step 2, then x^{k+1} is computed in finite time.

Theorem 6. *Algorithm 1 is well defined.*

Proof. Assume that x^k is a noncritical iterate generated by Algorithm 1. Then, similarly to [32, Proposition 1], it is possible to show that there exists $\bar{t} > 0$ such that (21) holds if $t_k \in]0, \bar{t}]$. Therefore, since $\omega_2 < 1$, the backtracking process at Step 3 finishes in a finite number of inner iterations and x^{k+1} is computed. □

Algorithm 1 successfully stops if a critical point is found. Otherwise, by Theorem 6, a sequence $\{x^k\}$ of noncritical iterates is generated. We proceed with the convergence analysis assuming that the algorithm iterates infinitely. The following lemma establishes that, for each $j = 1, \ldots, m$, the sequence $\{C_j^k\}$ is nonincreasing. This result coincides with the one presented in [32, Lemma 7]. However, we include its proof, due to its simplicity and usefulness in the forthcoming theorem.

Lemma 7. *Let $\{x^k\}$ be the sequence generated by Algorithm 1. Then, for each $j = 1, \ldots, m$, $\{C_j^k\}$ is a nonincreasing sequence.*

Proof. By (21) and (22), for each $j = 1, \ldots, m$ and all k, we obtain

$$\begin{aligned} C_j^{k+1} &= \frac{\eta q_k}{q_{k+1}} C_j^k + \frac{1}{q_{k+1}} F_j(x^{k+1}) \leq \frac{\eta q_k}{q_{k+1}} C_j^k + \frac{1}{q_{k+1}} [C_j^k + \sigma t_k f(x^k, d(x^k))] \\ &= C_j^k + \frac{\sigma t_k}{q_{k+1}} f(x^k, d(x^k)) \leq C_j^k, \end{aligned}$$

where the last inequality holds because $f(x^k, d(x^k)) < 0$. □

We next establish the global convergence of the sequence $\{x^k\}$ generated by Algorithm 1. Specifically, we show that any limit point of $\{x^k\}$ is Pareto critical. Note that the existence of limit points is not guaranteed at all. However, we point out that a sufficient condition for the existence of limit points is the boundedness of the set level $\{x \in \mathbb{R}^n \mid F(x) \leq F(x^0)\}$.

Theorem 8. *Let $\{x^k\}$ be the sequence generated by Algorithm 1. If x^* is a limit point of $\{x^k\}$, then x^* is Pareto critical.*

Proof. Let $K = \{k_0, k_1, k_2, ...\} \underset{\infty}{\subset} \mathbb{N}$ be such that $\lim_{i \to \infty} x^{k_i} = x^*$. Since $k_{i+1} \geq k_i + 1$, in view of Lemmas 5 and 7, we obtain, for all $i \in \mathbb{N}$,

$$F_j(x^{k_{i+1}}) \leq C_j^{k_{i+1}} \leq C_j^{k_i+1} \leq C_j^{k_i} + \frac{\sigma t_{k_i}}{q_{k_i+1}} f(x^{k_i}, d(x^{k_i})) \leq C_j^{k_i}, \quad \forall j = 1, \ldots, m, \tag{23}$$

where the third inequality is due to (21) and (22), and the last one holds because $f(x^{k_i}, d(x^{k_i})) < 0$. By continuity arguments, we have $\lim_{i \to \infty} F(x^{k_i}) = F(x^*)$ which, in particular, implies that $\{F(x^{k_i})\}$ is bounded. As a consequence, by (23), we obtain, for each $j = 1, \ldots, m$, that $\{C_j^{k_i}\}$ is a monotone bounded sequence and hence it admits limit. For any $j = 1, \ldots, m$, it follows, by taking limits in (23), that $\lim_{i \to \infty} (t_{k_i}/q_{k_i+1}) f(x^{k_i}, d(x^{k_i})) = 0$, which, combined with (19), yields

$$\lim_{i \to \infty} \frac{t_{k_i}}{q_{k_i+1}} \|d(x^{k_i})\| \|d_{SD}(x^{k_i})\| = 0. \tag{24}$$

We claim that

$$\lim_{i \to \infty} t_{k_i} \|d(x^{k_i})\| \|d_{SD}(x^{k_i})\| = 0. \tag{25}$$

If $\eta = 0$, then $q_{k_i+1} = 1$ for all $i \in \mathbb{N}$, and (25) trivially holds. Assuming that $\eta \in]0, 1[$, by (22), we have

$$q_{k_i+1} = 1 + \sum_{\ell=0}^{k_i} \eta^{\ell+1} \leq \sum_{\ell=0}^{\infty} \eta^{\ell} = \frac{1}{1-\eta}.$$

Therefore, $\{1/q_{k_i+1}\}$ is bounded from below and (25) follows from (24).

Now, by (25), there exists $K_1 \underset{\infty}{\subseteq} K$ such that: (a) $\lim_{k\in K_1} \|d_{SD}(x^k)\| = 0$ or (b) $\lim_{k\in K_1} t_k\|d(x^k)\| = 0$.

Case (a): in this case, it follows from Lemma 2(c) that $d_{SD}(x^*) = 0$ and hence x^* is Pareto critical.

Case (b): in this case, there exists a subsequence of indices $K_2 \underset{\infty}{\subseteq} K_1$ such that: (b1) $\lim_{k\in K_2} \|d(x^k)\| = 0$ or (b2) $\lim_{k\in K_2} t_k = 0$.

Case (b1): in this case, taking limits on both sides of (20) for $k \in K_2$ and considering Lemma 2(c), we also have $d_{SD}(x^*) = 0$, concluding that x^* is Pareto critical.

Case (b2): without loss of generality, assume that $t_k < 1$ for all $k \in K_2$. Hence, for all $k \in K_2$, by Step 3, there exist $\bar{t}_k \in]0, t_k/\omega_1]$ and $i_k \in \{1, \ldots, m\}$ such that

$$F_{i_k}(x^k + \bar{t}_k d(x^k)) > C^k_{i_k} + \sigma\bar{t}_k f(x^k, d(x^k)) \geq F_{i_k}(x^k) + \sigma\bar{t}_k f(x^k, d(x^k)), \tag{26}$$

where the second inequality follows from Lemma 5. Let us define $s_k := \bar{t}_k d(x^k)$. Since $\bar{t}_k \in]0, t_k/\omega_1]$ and $\lim_{k\in K_2} t_k\|d(x^k)\| = 0$, we obtain $\lim_{k\in K_2} \|s_k\| = 0$. By the mean value theorem, (26), and Lemma 1(a), there exists $\varepsilon_k \in]0, 1[$ such that

$$\langle \nabla F_{i_k}(x^k + \varepsilon_k s_k), s_k\rangle = F_{i_k}(x^k + s_k) - F_{i_k}(x^k) > \sigma f(x^k, s_k).$$

Thus, dividing both sides of above inequality by $\|s_k\|$, using the definition of f in (3) and Lemma 1(a), we obtain

$$f\left(x^k + \varepsilon_k s_k, \frac{s_k}{\|s_k\|}\right) > \sigma f\left(x^k, \frac{s_k}{\|s_k\|}\right). \tag{27}$$

On the other hand, by using Lemma 1(a) and the definition of s_k, it follows from (19) that $f(x^k, s_k) \leq -\Gamma_1\|d_{SD}(x^k)\|\|s_k\|$, or, equivalently,

$$f\left(x^k, \frac{s_k}{\|s_k\|}\right) \leq -\Gamma_1\|d_{SD}(x^k)\|. \tag{28}$$

Since $\{s_k/\|s_k\|\}_{k\in K_2}$ is bounded, there exist $K_3 \underset{\infty}{\subseteq} K_2$ and $s \in \mathbb{R}^n$ such that $\lim_{k\in K_3}\{s_k/\|s_k\|\} = s$. Taking limits for $k \in K_3$ on both sides of (27) and (28), we obtain $(1 - \sigma)f(x^*, s) \geq 0$ and $f(x^*, s) \leq -\Gamma_1\|d_{SD}(x^*)\|$, respectively. Since $\sigma \in]0, 1[$, it follows from the last two inequalities that $d_{SD}(x^*) = 0$. Therefore, x^* is Pareto critical and the proof is complete. □

4 Newton-type Methods for Multiobjective Optimization

The main goal of this section is to propose and analyze two Newton-type methods for solving (possibly) nonconvex multiobjective problems, which can be seen as particular cases of Algorithm 1. The first algorithm is directly inspired by the Newton method designed to solve convex problems [12]. In turn, the second one uses second-order information of the objective functions with ingredients of the steepest descent method. In both, in order to adjust the corresponding directions to the requirements of Algorithm 1, we adopt, from the scalar optimization, the technique in which the Hessians are modified, if necessary, by adding multiples of the identity.

4.1 Newton Method with Safeguarded Directions

Before we sketch the algorithm, let us derive some useful properties involving the steepest descent and the Newton-type directions. Let $x \in \mathbb{R}^n$ be given and assume that $B_j(x)$ is some positive definite approximation of $\nabla^2 F_j(x)$ for each $j = 1, \ldots, m$. Hereafter, considering a dual solution $\lambda^N(x) \in \mathbb{R}^m$ of (15), we denote:

$$d_\lambda(x) := \sum_{j=1}^{m} \lambda_j^N \nabla F_j(x). \tag{29}$$

The above direction will play an important role in order to check if the the safeguard conditions (19) and (20) hold for the directions generated by the method of this section.

Lemma 9. *Let $x \in \mathbb{R}^n$ be given and assume that $B_j(x)$ is positive definite for each $j = 1, \ldots, m$. Let $\lambda_{\min} > 0$ and $\lambda_{\max} > 0$ be such that $\lambda_{\min} I \preceq B_j(x) \preceq \lambda_{\max} I$, for all $j = 1, \ldots, m$. Consider $d_{SD}(x)$, $\theta_{SD}(x)$, $d_N(x)$, $\theta_N(x)$, and $d_\lambda(x)$ as in* (5), (6), (13), (14), *and* (29), *respectively. Then, the following inequalities hold:*

(a) $\frac{\lambda_{\min}}{2}\|d_N(x)\|^2 \leq |\theta_N(x)| \leq \frac{\lambda_{\max}}{2}\|d_N(x)\|^2$;

(b) $\frac{1}{\lambda_{\max}}|\theta_{SD}(x)| \leq |\theta_N(x)| \leq \frac{1}{\lambda_{\min}}|\theta_{SD}(x)|$;

(c) $\frac{1}{\lambda_{\max}}\|d_{SD}(x)\| \leq \|d_N(x)\| \leq \frac{1}{\lambda_{\min}}\|d_{SD}(x)\|$;

(d) $f(x, d_N(x)) \leq -\frac{\lambda_{\min}}{2\lambda_{\max}}\|d_{SD}(x)\|\|d_N(x)\|$;

(e) $\frac{\lambda_{\min}}{\lambda_{\max}}\|d_\lambda(x)\| \leq \|d_{SD}(x)\| \leq \|d_\lambda(x)\|$;

(f) $f(x, d_N(x)) \leq -\frac{\lambda_{\min}}{2\lambda_{\max}}\|d_\lambda(x)\|\|d_N(x)\|$.

Proof. (a) Define

$$B(x) := \sum_{j=1}^{m} \lambda_j^N B_j(x). \tag{30}$$

Then, by (17) and the definitions of $\lambda_{\min}$ and $\lambda_{\max}$, we have $\lambda_{\min} I \preceq B(x) \preceq \lambda_{\max} I$. Therefore, considering the characterization (18) and remembering that $\theta_N(x) \leq 0$, item (a) holds trivially. (b) For all $d \in R^n$ and $j = 1, \ldots, m$, we have $\lambda_{\min}\|d\|^2 \leq \langle B_j(x)d, d\rangle \leq \lambda_{\max}\|d\|^2$. Therefore, by (13) and (14), we obtain

$$\begin{aligned} \min_{d\in\mathbb{R}^n} \max_{j=1,\ldots,m} \langle \nabla F_j(x), d\rangle + \frac{\lambda_{\min}}{2}\|d\|^2 &\leq \theta_N(x) \\ &\leq \min_{d\in\mathbb{R}^n} \max_{j=1,\ldots,m} \langle \nabla F_j(x), d\rangle + \frac{\lambda_{\max}}{2}\|d\|^2. \end{aligned} \tag{31}$$

Now, for every $c > 0$, it follows from (6) that

$$\begin{aligned} \min_{d\in\mathbb{R}^n} \max_{j=1,\ldots,m} \langle \nabla F_j(x), d\rangle + \frac{c}{2}\|d\|^2 &= \frac{1}{c}\left[\min_{d\in\mathbb{R}^n} \max_{j=1,\ldots,m} \langle \nabla F_j(x), cd\rangle + \frac{1}{2}\|cd\|^2\right] \\ &= \frac{1}{c}\theta_{SD}(x). \end{aligned}$$

Hence, (31) implies that $(1/\lambda_{\min})\theta_{SD}(x) \leq \theta_N(x) \leq (1/\lambda_{\max})\theta_{SD}(x)$. Thus, since $\theta_{SD}(x)$ and $\theta_N(x)$ are nonpositive, the proof of item (b) follows from the last inequalities.
(c) It follows from items (a) and (b) that

$$\frac{1}{\lambda_{\max}}|\theta_{SD}(x)| \leq |\theta_N(x)| \leq \frac{\lambda_{\max}}{2}\|d_N(x)\|^2$$

and

$$\frac{\lambda_{\min}}{2}\|d_N(x)\|^2 \leq |\theta_N(x)| \leq \frac{1}{\lambda_{\min}}|\theta_{SD}(x)|.$$

Therefore, using (10), we obtain the desired inequalities.
(d) By Lemma 4(b) and the first inequality of (b), we have $f(x, d_N(x)) \leq -|\theta_N(x)| \leq -(1/\lambda_{\max})|\theta_{SD}(x)|$. On the other hand, by (10) and the second inequality of (c), it follows that

$$-|\theta_{SD}(x)| = -\frac{\|d_{SD}(x)\|^2}{2} \leq -\frac{\lambda_{\min}}{2}\|d_{SD}(x)\|\|d_N(x)\|.$$

Therefore, combining the two inequalities above, we obtain item (d).
(e) It follows from (16), (29), and the definition of $B(x)$ in (30) that $d_N(x) = -B(x)^{-1}d_\lambda(x)$. Hence, by the second inequality of (c), we have

$$\begin{aligned}\frac{1}{\lambda_{\min}^2}\|d_{SD}(x)\|^2 \geq \|d_N(x)\|^2 = \|B(x)^{-1}d_\lambda(x)\|^2 &= \langle B(x)^{-2}d_\lambda(x), d_\lambda(x)\rangle \\ &\geq \frac{1}{\lambda_{\max}^2}\|d_\lambda(x)\|^2,\end{aligned}$$

which implies the first inequality in (e). The second inequality follows from the definition of $d_\lambda(x)$ in (29), (17) and Lemma 3.
(f) From Lemma 4(b), (18), (30), and the fact that $d_N(x) = -B(x)^{-1}d_\lambda(x)$, we have

$$\begin{aligned}f(x, d_N(x)) \leq \theta_N(x) = -\frac{1}{2}d_N(x)^T B(x)d_N(x) &= -\frac{1}{2}d_\lambda(x)^T B(x)^{-1}d_\lambda(x) \\ &\leq -\frac{1}{2\lambda_{\max}}\|d_\lambda(x)\|^2.\end{aligned}$$

Thus, using the second inequalities of (e) and (c), we find $f(x, d_N(x)) \leq -[\lambda_{\min}/(2\lambda_{\max})]\|d_\lambda(x)\|\|d_N(x)\|$, concluding the proof. □

Let us now formally describe the Newton algorithm with safeguarded directions.

Algorithm 2. *(A Newton Algorithm with Safeguarded Directions)*

Step 0. *Let $x^0 \in \mathbb{R}^n$, $\Gamma_1 \in]0, 1/2[$, $\Gamma_2 > 0$, $\mu_{ini} > 0$, $\sigma \in]0, 1/2[$, $0 < \omega_1 < \omega_2 < 1$, and $\eta \in [0, 1[$ be given. Set $C^0 = F(x^0)$, $q_0 = 1$, and $\mu \leftarrow \mu_{ini}$. Initialize $k \leftarrow 0$.*

Step 1. *For each $j = 1, \ldots, m$, proceed as follows: if $\nabla^2 F_j(x^k)$ is positive definite, then define $B_j(x^k) := \nabla^2 F_j(x^k)$; otherwise, find $\rho_j > 0$ such that $\nabla^2 F_j(x^k) + \rho_j I$ is positive definite and set*

$$B_j(x^k) := \nabla^2 F_j(x^k) + \rho_j I.$$

Step 2. *Solve problem* (15) *to obtain $\left(\theta_N(x^k), d_N(x^k), \lambda^N(x^k)\right) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ satisfying* (16)–(18).

Step 3. *If* $\theta_N(x^k) = 0$, *then* **stop**.

Step 4. *Compute* $d_\lambda(x^k)$ *as in* (29). *If* $f(x^k, d_N(x^k)) > -\Gamma_1 \|d_\lambda(x^k)\| \|d_N(x^k)\|$, *then set*

$$B_j(x^k) \leftarrow B_j(x^k) + \mu I, \quad \forall j = 1, \ldots, m,$$

$\mu \leftarrow 2\mu$, *and return to Step 2.*

Step 5. *If* $\|d_N(x^k)\| < \Gamma_2 \|d_\lambda(x^k)\|$, *then* $d_N(x^k) \leftarrow \Gamma_2 \frac{\|d_\lambda(x^k)\|}{\|d_N(x^k)\|} d_N(x^k)$.

Step 6. *Compute* t_k *as in Step 3 of Algorithm 1 and set* $x^{k+1} = x^k + t_k d(x^k)$. *Update* q_{k+1} *and* C^{k+1} *as in Step 4 of Algorithm 1. Set* $\mu \leftarrow \mu_{ini}$ *and* $k \leftarrow k + 1$, *and go to Step 1.*

Some comments are in order. First, at Step 1 if the initial approximation $\nabla^2 F_j(x^k)$ is not definite positive, then a suitable multiple of the identity $\rho_j I$ is added to $\nabla^2 F_j(x^k)$ to obtain $\nabla^2 F_j(x^k) + \rho_j I$ definite positive. In practice, one can verify whether a matrix is definite positive by trying to perform its Cholesky factorization. Thus, in principle, the computational cost involved in Step 1 is proportional to the number of objectives m. Second, the matrices $B_j{}'s$ are always definite positive at Step 2 and hence problem (15) has a unique solution. Third, although we are interested in showing that Algorithm 2 is a particular case of Algorithm 1, we see that Algorithm 2 does not explicitly use the steepest descent direction $d_{SD}(x^k)$. Instead of (19) and (20), direction $d_N(x^k)$ must satisfy

$$f(x^k, d_N(x^k)) \leq -\Gamma_1 \|d_\lambda(x^k)\| \|d_N(x^k)\| \text{ and } \|d_N(x^k)\| \geq \Gamma_2 \|d_\lambda(x^k)\|. \tag{32}$$

Once problem (15) is solved, the computational cost of calculating $d_\lambda(x^k)$ is negligible. Lemma 9 contains the tools to link the safeguard conditions (19) and (20) with the conditions in (32). Fourth, in view of Lemma 9(f), if the algorithmic parameter Γ_1 is choose small enough, then we expect the first condition of (32) to be satisfied at Step 4. If this is not the case, an (increasing) multiple of the identity μI is added to $B_j{}'s$ and the algorithm returns to Step 2 to compute a new search direction. On the other hand, as μ increases, the length of $d_N(x^k)$ decreases. Thus, if $\|d_N(x^k)\| < \Gamma_2 \|d_\lambda(x^k)\|$, then the length of $d_N(x^k)$ is adjusted in Step 5.

The following lemma is connected to the well-definiteness of Algorithm 2. It shows that $d_N(x^k)$ eventually satisfies the first condition of (32).

Lemma 10. *Assume that* $x^k \in \mathbb{R}^n$ *is not Pareto critical and, for each* $j = 1, \ldots, m$, *let* $\bar{B}_j(x^k)$ *be the positive definite Hessian approximation given at the end of Step 1 of Algorithm 2. For* $\delta > 0$, *consider* $B_j^\delta(x^k) := \bar{B}_j(x^k) + \delta I$, *denote by* $\left(\theta_N^\delta(x^k), d_N^\delta(x^k), \lambda^{N\delta}(x^k)\right) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ *a corresponding primal-dual solution of* (15), *and define* $d_\lambda^\delta(x^k) := \sum_{j=1}^m \lambda_j^{N\delta} \nabla F_j(x^k)$. *Then, there exists* $\bar{\delta} > 0$ *such that*

$$f(x^k, d_N^\delta(x^k)) \leq -\Gamma_1 \|d_\lambda^\delta(x^k)\| \|d_N^\delta(x^k)\|, \quad \forall \delta \geq \bar{\delta}.$$

Proof. Let $\lambda_{\min}^\delta, \lambda_{\max}^\delta, \bar{\lambda}_{\min}, \bar{\lambda}_{\max} \in \mathbb{R}_{++}$ be such that $\lambda_{\min} I \preceq B_j^\delta(x^k) \preceq \lambda_{\max} I$ and $\lambda_{\min} I \preceq \bar{B}_j(x^k) \preceq \lambda_{\max} I$, for all $j = 1, \ldots, m$. By the definition of $B_j^\delta(x^k)$, we have $\lambda_{\min}^\delta = \bar{\lambda}_{\min} + \delta$ and $\lambda_{\max}^\delta = \bar{\lambda}_{\max} + \delta$. Hence, we obtain

$$\frac{\lambda_{\min}^\delta}{2\lambda_{\max}^\delta} = \frac{\bar{\lambda}_{\min} + \delta}{2(\bar{\lambda}_{\max} + \delta)} \xrightarrow{\delta \to \infty} \frac{1}{2} > \Gamma_1. \tag{33}$$

Using Lemma 9(f), we get $f(x^k, d_N^\delta(x^k)) \leq -[\lambda_{\min}^\delta / (2\lambda_{\max}^\delta)] \|d_\lambda^\delta(x^k)\| \|d_N^\delta(x^k)\|$. Therefore, the statement of the lemma follows combining the last inequality and (33). □

We next show that Algorithm 2 is well defined. This implies that, if Algorithm 2 does not stop at a Pareto critical iterate, then it generates an infinite sequence $\{x^k\}$.

Theorem 11. *Algorithm 2 is well defined and stops at x^k if and only if x^k is Pareto critical.*

Proof. We first observe that if $\nabla^2 F_j(x^k)$ is not definite positive then, for ρ_j large enough, $\nabla^2 F_j(x^k) + \rho_j I$ is definite positive, fulfilling the requirement of Step 1 (for example, ρ_j can be chosen as the smallest element of $\{2^\ell\}_{\ell \geq 0}$ satisfying that condition). Thus, at the end of Step 1, $B_j(x_k)$, $j = 1, \ldots, m$, are positive definite matrices and subproblem (15) at Step 2 is solvable. Now, since by Lemma 4 x^k is Pareto critical if and only if $\theta_N(x^k) = 0$, it follows that Algorithm 2 stops at Step 3 if and only if x^k is Pareto critical. If x^k is not Pareto critical, since the algorithm uses increasing values for μ at Step 4, Lemma 10 implies that Algorithm 2 proceeds to Step 5 performing a finite number of (inner) steps. We conclude that Algorithm 2 is well defined by observing that it uses the same line search scheme as Algorithm 1. □

The following theorem establishes the global convergence of Algorithm 2 in the sense that every limit point of the generated sequence $\{x^k\}$ is Pareto critical. This will be done by showing that Algorithm 2 is a particular case of Algorithm 1.

Theorem 12. *Algorithm 2 is an instance of Algorithm 1. As a consequence, any limit point of the sequence $\{x^k\}$ generated by Algorithm 2 is Pareto critical.*

Proof. By the definition of Algorithm 2, the search direction $d_N(x^k) \in \mathbb{R}^n$ is such that the inequalities in (32) hold for all k. Thus, using Lemma 9(e), we obtain $f(x^k, d_N(x^k)) \leq -\Gamma_1 \|d_{SD}(x^k)\| \|d_N(x^k)\|$ and $\|d_N(x^k)\| \geq \Gamma_2 \|d_{SD}(x^k)\|$, concluding that $d_N(x^k)$ satisfies the safeguard conditions (19) and (20) for all k. Since Algorithm 2 uses the same line search scheme as Algorithm 1, the proof is concluded. □

We now discuss some local converge properties related to Algorithm 2. The following theorem establishes that, under local convexity assumptions, if an iterate x^k is *close* to a local Pareto optimal point $\bar{x}$, then the sequence $\{x^k\}$ generated by Algorithm 2 converges *quickly* to some Pareto optimal solution in a vicinity of $\bar{x}$. This means that the proposed globalized version enjoys the local converge properties of the Newton method showed in [12]. Before stating the theorem, we note that if $\bar{x}$ is such that $\nabla^2 F_j(\bar{x})$ is positive definite for all $j = 1, \ldots, m$, then there exist a neighborhood $V_{\bar{x}} \subset \mathbb{R}^n$ of $\bar{x}$ and $\lambda_{\min}, \lambda_{\max} \in \mathbb{R}_{++}$, such that

$$\lambda_{\min} I \preceq \nabla^2 F_j(x) \preceq \lambda_{\max} I \quad \text{for all} \quad x \in V_{\bar{x}}, \quad j = 1, \ldots, m. \tag{34}$$

Theorem 13. *Let $\bar{x} \in \mathbb{R}^n$ be a local Pareto optimal point and assume that $\nabla^2 F_j(\bar{x})$ is positive definite for all $j = 1, \ldots, m$. Let $V_{\bar{x}} \subset \mathbb{R}^n$ and $\lambda_{\min}, \lambda_{\max} \in \mathbb{R}_{++}$ as in (34). Consider the application of Algorithm 2 and assume that the algorithmic parameters Γ_1 and Γ_2 are chosen such that $0 < \Gamma_1 \leq \lambda_{\min}/(2\lambda_{\max})$ and $0 < \Gamma_2 \leq 1/\lambda_{\max}$. Then, there exist $0 < \delta \leq r$ such that if $x^0 \in B[\bar{x}, \delta] \subset V_{\bar{x}}$ then, for all k, we have:*

1. $d_N(x^k) = -\left[\sum_{j=1}^m \lambda_j^N(x^k) \nabla^2 F_j(x^k)\right]^{-1} \sum_{j=1}^m \lambda_j^N(x^k) \nabla F_j(x^k)$, *which means that no corrections are needed in the Newton directions;*
2. $t_k = 1$;
3. $x^k \in B[x^0, r] \subset V_{\bar{x}}$.

Moreover, $\{x^k\}$ converges superlinearly to some local Pareto optimal point $x^* \in B[x^0, r]$. In addition, if $\nabla^2 F_j$ are Lipschitz continuous on $V_{\bar{x}}$ for all $j = 1, \ldots, m$, then the convergence is quadratic.

Proof. Consider an iteration k for which $x^k \in V_{\bar{x}}$. Then, at the end of Step 1, by (34), $B_j(x^k) = \nabla^2 F_j(x^k)$, for all $j = 1, \ldots, m$, at the end of Step 1 and Algorithm 2 computes $d_N(x^k)$ at Step 2 as in the first item of the theorem. Using now (34), Lemma 9(f), and the definitions of Γ_1 and Γ_2, we have $f(x^k, d_N(x^k)) \leq -\Gamma_1 \|d_\lambda(x^k)\| \|d_N(x^k)\|$ and $\|d_N(x^k)\| \geq \Gamma_2 \|d_\lambda(x^k)\|$, implying that Algorithm 2 proceeds to the line search phase without changing $d_N(x^k)$. Therefore, from [12, Theorem 5.1] (see also, [12, Corollary 5.2]), we obtain

$$F_j(x^k + d(x^k)) \leq F_j(x^k) + \bar{\sigma}\theta_N(x^k), \quad \forall j = 1, \ldots, m, \tag{35}$$

where $\bar{\sigma} := 2\sigma < 1$. On the other hand, by (16) and (18), we have

$$\theta_N(x^k) = \frac{1}{2} \sum_{j=1}^{m} \lambda_j^N(x^k) \langle \nabla F_j(x^k), d_N(x^k) \rangle,$$

which, together with (3) and (17), yields $\theta_N(x^k) \leq f(x^k, d_N(x^k))/2$. Combining the last inequality with (35) and using Lemma 5, we conclude that

$$F_j(x^k + d(x^k)) \leq C_j^k + \sigma f(x^k, d_N(x^k)), \quad \forall j = 1, \ldots, m.$$

Thus, $t_k = 1$ satisfies the nonmonotone descent condition (21). As consequence, again from [12, Theorem 5.1], we have $x^{k+1} \in B[x^0, r] \subset V_{\bar{x}}$. Hence, by inductive arguments, items 1, 2, and 3 hold for all k. This mean that the sequence $\{x^k\}$ generated by Algorithm 2 coincides with the one generated by the Newton method of [12]. Therefore, the convergence rate results are obtained directly from [12, Theorem 5.1, Corollary 5.2, Theorem 6.1, Corollary 6.2]. □

Under some assumptions of strict convexity over F (which is satisfied, for example, if F is strongly convex), the next result shows that Algorithm 2 with suitable algorithmic parameters converges superlinearly (or quadratically) to an efficient point.

Corollary 14. *Assume that $\nabla^2 F_j(x)$ is positive definite for all $j = 1, \ldots, m$ and all $x \in \mathbb{R}^n$. If the starting point x^0 is such that the level set $\{x \in \mathbb{R}^n \mid F(x) \leq F(x^0)\}$ is bounded, then Algorithm 2 generates a sequence $\{x^k\}$ that converges superlinearly to some Pareto optimal point $\bar{x}$, provided that the algorithmic parameters Γ_1 and Γ_2 are chosen as in Theorem 13. In addition, if $\nabla^2 F_j$ are Lipschitz continuous for all $j = 1, \ldots, m$, then the convergence is quadratic.*

Proof. Since x^k belongs to the bounded level set $\{x \in \mathbb{R}^n \mid F(x) \leq F(x^0)\}$ for all k, it turns out that $\{x^k\}$ has a limit point $\bar{x}$. Hence, in view of Theorem 12 and [12, Theorem 3.1], we have that $\bar{x}$ is Pareto optimal. Moreover, there exists k_0 such that x^{k_0} belongs to the convergence region of Theorem 13. If Γ_1 and Γ_2 are small enough, then Theorem 13 is applicable and hence the whole sequence converges superlinearly (or quadratically) to $\bar{x}$. □

4.2 Newton-Gradient Method with Safeguarded Directions

As mentioned earlier, the method proposed in this section uses second-order information of the objective functions with ingredients of the steepest descent method. We proceed by discussing its main ideas. For a given $x \in \mathbb{R}^n$, consider problem (7) and let $d_{SD}(x)$ and $\lambda^{SD}(x)$ be as in (8)

and (9), respectively. Define the scalarized function $g : \mathbb{R}^n \to \mathbb{R}$ by $g(y) := \sum_{j=1}^m \lambda_j^{SD}(x) F_j(y)$. In view of (8), $d_{SD}(x)$ corresponds to the steepest descent direction of g at x. Thus, an iteration of the multiobjective steepest descent method can be seen as an iteration of the scalar gradient method for g. When a new iterate is computed, the KKT multipliers of (7) change and a new scalarized function g is considered. The class of methods proposed in this section uses, at each iteration, a Newton-type search direction for the scalarized function g. For such directions to deal with the possibility that they do not yield descents, we use the same technique as in section 4.1 (i.e., we modify the Hessians, if necessary, by adding multiples of the identity). We next describe formally the method.

Algorithm 3. *(A Newton-Gradient Algorithm with Safeguarded Direction)*

Step 0. *Let* $x^0 \in \mathbb{R}^n$, $\Gamma_1 \in]0,1]$, $\Gamma_2 > 0$, $\sigma \in]0,1/2[$, $0 < \omega_1 < \omega_2 < 1$, *and* $\eta \in [0,1[$ *be given. Set* $C^0 = F(x^0)$, $q_0 = 1$, $\mu \leftarrow 0$. *Initialize* $k \leftarrow 0$.

Step 1. *Solve problem* (7) *to obtain* $\left(\theta_{SD}(x^k), d_{SD}(x^k), \lambda^{SD}(x^k)\right) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ *satisfying* (8)–(10).

Step 2. *If* $\theta_{SD}(x^k) = 0$, *then* **stop**.

Step 3. *Compute* $B_k := \sum_{j=1}^m \lambda_j^{SD}(x^k) \nabla^2 F_j(x^k)$. *If* B_k *is not positive definite, then find* $\mu > 0$ *such that* $B_k + \mu I$ *is positive definite and set* $B_k := \left[\sum_{j=1}^m \lambda_j^{SD}(x^k) \nabla^2 F_j(x^k)\right] + \mu I$.

Step 4. *Obtain* $d_{NG}(x^k)$ *by solving the linear system* $B_k d_{NG}(x^k) = d_{SD}(x^k)$.

Step 5. *If* $f(x^k, d_{NG}(x^k)) > -\Gamma_1 \|d_{SD}(x^k)\| \|d_{NG}(x^k)\|$, *then set*

$$\mu \leftarrow \max\{2\mu, 1\}, \quad B_k := \left[\sum_{j=1}^m \lambda_j^{SD}(x^k) \nabla^2 F_j(x^k)\right] + \mu I,$$

and return to Step 4.

Step 6. *If* $\|d_{NG}(x^k)\| < \Gamma_2 \|d_{SD}(x^k)\|$, *then* $d_{NG}(x^k) \leftarrow \Gamma_2 \frac{\|d_{SD}(x^k)\|}{\|d_{NG}(x^k)\|} d_{NG}(x^k)$.

Step 7. *Compute* t_k *as in Step 3 of Algorithm 1 and set* $x^{k+1} = x^k + t_k d(x^k)$. *Update* q_{k+1} *and* C^{k+1} *as in Step 4 of Algorithm 1. Set* $\mu \leftarrow 0$ *and* $k \leftarrow k+1$, *and go to Step 1.*

The main feature of Algorithm 3 is that it requires the positiveness of only one matrix. Therefore, remembering that we need to guarantee the positiveness of m matrices in Algorithm 2, Algorithm 3 provides an attractive alternative to deal with problems that have many objectives. Furthermore, subproblem (7) is solved only once per iteration at Step 1, in contrast to Algorithm 2 for which subproblem (15) is solved whenever the first condition of (32) does not hold.

Let us discuss some specific points of Algorithm 3. At Step 3, similarly to Algorithm 2, we consider adding a suitable multiple of the identity to get B_k definite positive. Now, since in an iteration k the algorithm uses increasing values for μ (see Step 5), it follows that B_k is always definite positive at Step 4 and hence $d_{NG}(x^k)$ is well defined and can be obtained by using the Cholesky factorization to solve the linear system. At each iteration, $d_{NG}(x^k)$ is required to satisfy:

$$f(x^k, d_{NG}(x^k)) \leq -\Gamma_1 \|d_{SD}(x^k)\| \|d_{NG}(x^k)\| \text{ and } \|d_{NG}(x^k)\| \geq \Gamma_2 \|d_{SD}(x^k)\|, \tag{36}$$

which correspond directly to the safeguard conditions (19) and (20) of Algorithm 1. At Step 5, if $d_{NG}(x^k)$ does not satisfy the first condition of (36), then the diagonal of B_k is increased and the algorithm returns to Step 4 to compute a new direction. In the following lemma, we will show that this condition is eventually satisfied, i.e., the first condition of (36) holds for sufficiently large values of μ. In particular, we will prove that the direction of $d_{NG}(x^k)$ tends to the direction of $d_{SD}(x^k)$ as μ grows. Finally, if necessary, as in Algorithm 2, the length of $d_{NG}(x^k)$ is adjusted at Step 6 to satisfy the second inequality of (36).

Lemma 15. *Assume that $x^k \in \mathbb{R}^n$ is not Pareto critical. For $\delta > 0$, define*

$$B_k^\delta := [\sum_{j=1}^{m} \lambda_j^{SD}(x^k)\nabla^2 F_j(x^k)] + \delta I,$$

and consider $d_{NG}^\delta(x^k) \in \mathbb{R}^n$ such that $B_k^\delta d_{NG}^\delta(x^k) = d_{SD}(x^k)$. Then, there exists $\bar{\delta} > 0$ such that

$$f(x^k, d_{NG}^\delta(x^k)) \leq -\Gamma_1 \|d_{SD}(x^k)\| \|d_{NG}^\delta(x^k)\|, \quad \forall \delta \geq \bar{\delta}.$$

Proof. Without loss of generality, we assume that δ is large enough such that B_k^δ is definite positive and hence $d_{NG}^\delta(x^k)$ is well defined. Now, since x^k is not Pareto critical, then $d_{SD}(x^k) \neq 0$ (see Lemma 2(b)), which, in turn, implies that $d_{NG}^\delta(x^k) \neq 0$. Let us denote $B := \sum_{j=1}^m \lambda_j^{SD}(x^k)\nabla^2 F_j(x^k)$. Thus, by the definition of $d_{NG}^\delta(x^k)$, we have

$$d_{NG}^\delta(x^k) = \frac{1}{\delta}\left(\frac{B}{\delta} + I\right)^{-1} d_{SD}(x^k).$$

Therefore,

$$\frac{d_{NG}^\delta(x^k)}{\|d_{NG}^\delta(x^k)\|} = \frac{\left(\frac{B}{\delta} + I\right)^{-1} d_{SD}(x^k)}{\| \left(\frac{B}{\delta} + I\right)^{-1} d_{SD}(x^k)\|} \xrightarrow{\delta \to \infty} \frac{d_{SD}(x^k)}{\|d_{SD}(x^k)\|}.$$

Hence, using Lemma 1(a), the continuity of $f(x^k, \cdot)$ and (11), we obtain

$$\begin{aligned}\frac{f(x^k, d_{NG}^\delta(x^k))}{\|d_{NG}^\delta(x^k)\|} = f\left(x^k, \frac{d_{NG}^\delta(x^k)}{\|d_{NG}^\delta(x^k)\|}\right) &\xrightarrow{\delta \to \infty} f\left(x^k, \frac{d_{SD}(x^k)}{\|d_{SD}(x^k)\|}\right) \\ &= \frac{f(x^k, d_{SD}(x^k))}{\|d_{SD}(x^k)\|} = -\|d_{SD}(x^k)\|.\end{aligned}$$

Thus, as $\Gamma_1 \in]0, 1]$, there exists $\bar{\delta} > 0$ such that $f(x^k, d_{NG}^\delta(x^k))/\|d_{NG}^\delta(x^k)\| \leq -\Gamma_1 \|d_{SD}(x^k)\|$, for all $\delta \geq \bar{\delta}$, concluding the proof. □

The following theorem establishes the well-definiteness of Algorithm 3 which, in particular, implies that a sequence of points $\{x^k\}$ is obtained if the algorithm does not generate a Pareto critical iterate.

Theorem 16. *Algorithm 3 is well-defined and stops at x^k if and only if x^k is Pareto critical.*

Proof. Using Lemma 2, we conclude that x^k is Pareto critical if and only if $\theta_{SD}(x^k) = 0$. Therefore, Algorithm 3 stops at Step 2 if and only if a Pareto critical point is found. If Algorithm 3 does not stop at iteration k then, similarly to Lemma 11, it is possible to find a suitable value

of μ such that B_k is definite positive at the end of Step 3. Thus, the linear system at Step 4 has unique solution and $d_{NG}(x^k)$ is well defined. Since the algorithm uses increasing values for μ at Step 5, it follows from Lemma 15 that Algorithm 3 proceeds to Step 6 performing a finite number of (inner) steps. Now, the well-definiteness of Algorithm 3 is obtained by observing that it uses the same line search scheme as Algorithm 1. □

We now show that Algorithm 3 is also an instance of Algorithm 1 and, as a consequence, its global convergence is obtained from Theorem 8.

Theorem 17. *Algorithm 3 is an instance of Algorithm 1. As a consequence, any limit point of the sequence $\{x^k\}$ generated by Algorithm 3 is Pareto critical.*

Proof. It follows straightforwardly from the definition of Algorithm 3 that (36) holds for for all k. Therefore, $d_{NG}(x^k)$ satisfies the safeguard conditions (19) and (20) for all k. Since Algorithm 3 uses the same line search scheme as Algorithm 1, the proof is concluded. □

We end this section by showing that a convergence result along the lines of Theorem 13 cannot be expected for Algorithm 3. The following example shows a strongly convex function for which any neighborhood of an arbitrary Pareto optimal point contains points for which the corresponding *pure* Newton-Gradient directions do not yield descent. Therefore, for Algorithm 3, corrections in the search directions are eventually necessary even for strongly convex problems and iterates arbitrarily close to efficient points.

Example 1. *Let $F : \mathbb{R}^2 \to \mathbb{R}^2$, $F(x_1, x_2) = (F_1(x_1, x_2), F_2(x_1, x_2))^T$, defined by*

$$F_1(x_1, x_2) = \frac{x_1^2}{2} + \frac{x_2^2}{2} \quad \textit{and} \quad F_2(x_1, x_2) = \frac{(x_1 - 2)^2}{2} + \frac{(2x_2 - 2)^2}{2}.$$

Note that F is strongly convex. The set of Pareto optimal points is given by $\{(2\alpha, 4\alpha/(1+3\alpha))^T \in \mathbb{R}^2 \mid 0 \leq \alpha \leq 1\}$, which corresponds to the solid curve in Figure 1. Define

$$U := \left\{ \begin{pmatrix} 2\alpha + \beta \\ 4\alpha/(1+3\alpha) \end{pmatrix} \in \mathbb{R}^2 \mid 0 < \alpha < 1; \right.$$
$$\left. 0 < \beta < \frac{2(\sqrt{27\alpha^3 + 27\alpha^2 + 45\alpha + 1} - 9\alpha^2 - 1)}{3(1+3\alpha)} \right\}. \tag{37}$$

The set U corresponds to the shaded region in Figure 1. Consider the application of Algorithm 3 with any starting point $x^0 \in U$. By solving (7), we obtain $d_{SD}(x^0) = (-4\beta, 2\beta(1+3\alpha))^T/[(1+3\alpha)^2 + 4]$. Direct calculations show that

$$\begin{aligned} B_0 &:= \lambda_1^{SD}(x^0)\nabla^2 F_1(x^0) + \lambda_2^{SD}(x^0)\nabla^2 F_2(x^0) \\ &= \begin{pmatrix} 1 & 0 \\ 0 & \dfrac{(1+3\alpha)(12\alpha + 3\beta + 9\alpha\beta + 18\alpha^2 + 10)}{2[(1+3\alpha)^2 + 4]} \end{pmatrix}, \end{aligned}$$

which is positive definite, as expected. Hence, by solving the linear system at Step 4, we obtain

$$d_{NG}(x^0) = \left(\frac{-4\beta}{(1+3\alpha)^2 + 4}, \frac{4\beta}{12\alpha + 3\beta + 9\alpha\beta + 18\alpha^2 + 10} \right)^T.$$

Using (3)*, after some algebraic manipulations, we have*

$$f(x^0, d_{NG}(x^0)) \geq \langle \nabla F_2(x^0), d_{NG}(x^0) \rangle$$
$$= \frac{16\beta(\alpha-1)}{(1+3\alpha)(12\alpha+3\beta+9\alpha\beta+18\alpha^2+10)} - \frac{4\beta(2\alpha+\beta-2)}{(1+3\alpha)^2+4} > 0,$$

where the last inequality is due to the definitions of α *and* β *in* (37)*. This concludes that* $d_{NG}(x^0)$ *is not a descent direction of* F *at* x^0*. Now consider an arbitrary Pareto optimal point* $\bar{x}$ *of* F*. As suggested in Figure 1, for any* $\delta > 0$*, we have* $B[\bar{x}, \delta] \cap U \neq \varnothing$*. As a consequence, any neighborhood of* $\bar{x}$ *contains points* $x \in U$ *for which the corresponding Newton-Gradient directions* $d_{NG}(x) = -\left[\sum_{j=1}^{2} \lambda_j^{SD}(x) \nabla^2 F_j(x)\right]^{-1} d_{SD}(x)$ *do not yield descent.*

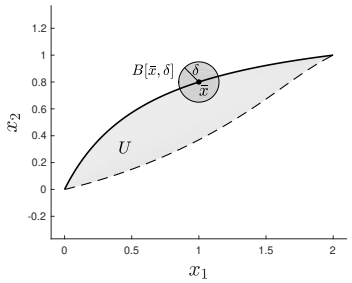

Figure 1: The solid curve corresponds to the set of efficient points of F. If $x \in U$, then the corresponding Newton-Gradient direction $d_{NG}(x) = -\left[\sum_{j=1}^{2} \lambda_j^{SD}(x) \nabla^2 F_j(x)\right]^{-1} d_{SD}(x)$ does not yield descent. Any neighborhood of an arbitrary efficient point $\bar{x}$ contains points belonging to U.

5 Numerical Experiments

Aiming to verify the effectiveness of the proposed globalization techniques, we present some computational results comparing the performances of Algorithms 2 and 3 with the Newton method without safeguards (Algorithm 2 with $B_j(x^k) = \nabla^2 F_j(x^k)$ at Step 1, for all $j = 1, \ldots, m$, and going from Step 3 directly to Step 6) and the steepest descent method (Algorithm 1 using $d_{SD}(x^k)$ at Step 1). Essentially, the last two methods correspond to the nonmonotonous versions of the Newton algorithm of [12] and the steepest descent algorithm of [13], respectively.

The chosen test problems consist of 44 convex and nonconvex unconstrained multiobjective problems found in the literature. Table 1 shows the main characteristics of the problems. The first two columns contain the problem name and the corresponding reference. Columns "n" and "m" inform the number of variables and the number of objectives, respectively. "C" indicates whether the problem is convex or not. For each problem, we considered starting points belonging to a box $\{x \in \mathbb{R}^n \mid l \leq x \leq u\}$, where the bounds $l, u \in \mathbb{R}^n$ are reported in the last two columns of the table. We emphasize that the boxes reported in Table 1 were used only to define the starting points, but were not considered by the algorithms themselves.

Problem	Source	n	m	Convex	l^T	u^T
AP1	[2]	2	3	Y	$(-10,-10)$	$(10,10)$
AP2	[2]	1	2	Y	-100	100
AP3	[2]	2	2	N	$(-100,-100)$	$(100,100)$
AP4	[2]	3	3	Y	$(-10,-10,-10)$	$(10,10,10)$
BK1	[21]	2	2	Y	$(-5,-5)$	$(10,10)$
DD1[a]	[9]	5	2	N	$(-20,\ldots,-20)$	$(20,\ldots,20)$
DGO1	[21]	1	2	N	-10	13
Far1	[21]	2	2	N	$(-1,-1)$	$(1,1)$
FDS	[12]	5	3	Y	$(-2,\ldots,-2)$	$(2,\ldots,2)$
FF1	[21]	2	2	N	$(-1,-1)$	$(1,1)$
Hil1	[20]	2	2	N	$(0,0)$	$(1,1)$
IKK1	[21]	2	3	Y	$(-50,-50)$	$(50,50)$
JOS1	[22]	100	2	Y	$(-100,\ldots,-100)$	$(100,\ldots,100)$
KW2	[23]	2	2	N	$(-3,-3)$	$(3,3)$
LE1	[21]	2	2	N	$(-5,-5)$	$(10,10)$
Lov1	[25]	2	2	Y	$(-10,-10)$	$(10,10)$
Lov3	[25]	2	2	N	$(-20,-20)$	$(20,20)$
Lov4	[25]	2	2	N	$(-20,-20)$	$(20,20)$
Lov5	[25]	3	2	N	$(-2,-2,-2)$	$(2,2,2)$
MGH16[b]	[33]	4	5	N	$(-25,-5,-5,-1)$	$(25,5,5,1)$
MGH26[b]	[33]	4	4	N	$(-1,-1,-1-1)$	$(1,1,1,1)$
MGH33[b]	[33]	10	10	Y	$(-1,\ldots,-1)$	$(1,\ldots,1)$
MHHM2	[21]	2	3	Y	$(0,0)$	$(1,1)$
MLF2	[21]	2	2	N	$(-100,-100)$	$(100,100)$
MMR1[c]	[31]	2	2	N	$(0.1,0)$	$(1,1)$
MMR3	[31]	2	2	N	$(-1,-1)$	$(1,1)$
MOP2	[21]	2	2	N	$(-1,-1)$	$(1,1)$
MOP3	[21]	2	2	N	$(-\pi,-\pi)$	(π,π)
MOP5	[21]	2	3	N	$(-1,-1)$	$(1,1)$
MOP7	[21]	2	3	Y	$(-400,-400)$	$(400,400)$
PNR	[36]	2	2	Y	$(-2,-2)$	$(2,2)$
QV1	[21]	10	2	N	$(-5,\ldots,-5)$	$(5,\ldots,5)$
SK1	[21]	1	2	N	-100	100
SK2	[21]	4	2	N	$(-10,-10,-10,-10)$	$(10,10,10,10)$
SLCDT1	[38]	2	2	N	$(-1.5,-1.5)$	$(1.5,1.5)$
SLCDT2	[38]	10	3	Y	$(-1,\ldots,-1)$	$(1,\ldots,1)$
SP1	[21]	2	2	Y	$(-100,-100)$	$(100,100)$
SSFYY2	[21]	1	2	N	-100	100
Toi4[b]	[40]	4	2	Y	$(-2,-2,-2,-2)$	$(5,5,5,5)$
Toi8[b]	[40]	3	3	Y	$(-1,-1,-1,-1)$	$(1,1,1,1)$
Toi9[b]	[40]	4	4	N	$(-1,-1,-1,-1)$	$(1,1,1,1)$
Toi10[b]	[40]	4	3	N	$(-2,-2,-2,-2)$	$(2,2,2,2)$
VU1	[21]	2	2	N	$(-3,-3)$	$(3,3)$
ZLT1	[21]	10	5	Y	$(-1000,\ldots,-1000)$	$(1000,\ldots,1000)$

[a] This is a modified version of DD1 problem that can be found in [12, 32].
[b] This is an adaptation of a single-objective optimization problem to the multiobjective setting that can be found in [32].
[c] This is a modified version of MMR1 problem that can be found in [27].

Table 1: List of test problems.

The numerical results will be shown using performance profiles graphics [10], which are useful tools for comparing several methods on a large set of test problems. Let $\mathcal{S}$ be the set of solvers, $\mathcal{P}$ be the set of problems, and $t_{p,s} > 0$ be the performance of the solver $s \in \mathcal{S}$ on the problem $p \in \mathcal{P}$, where lower values of $t_{p,s}$ mean better performances. Define the performance ratio $r_{p,s} := t_{p,s}/\min\{t_{p,s} \mid s \in \mathcal{S}\}$. Then, the performance profile is obtained by plotting, for all $s \in \mathcal{S}$, the cumulative distribution function $\rho_s : [1, \infty[\to [0, 1]$ for the performance ratio $r_{p,s}$ given by $\rho_s(\tau) := (1/|\mathcal{P}|)\left|\{p \in \mathcal{P} \mid r_{p,s} \leq \tau\}\right|$.

5.1 Some Implementation Details

We implemented the algorithms in Fortran 90. The codes, as well as the formulation of

each considered test problem, are freely available at `https://lfprudente.ime.ufg.br/`. Due to numerical issues, we considered a scaled version of problem (1) given by

$$\min_{x\in\mathbb{R}^n} \left(\gamma_1 F_1(x), \ldots, \gamma_m F_m(x)\right), \tag{38}$$

where the scaling factors are computed as $\gamma_j := 1/\max(1, \|\nabla F_j(x^0)\|_\infty)$, $j = 1, \ldots, m$, where x^0 is the starting point. We point out that problems (1) and (38) are equivalent, in the sense that they have the same Pareto critical points.

In Algorithms 2 and 3, we test if a matrix A is positive definite by trying to find its Cholesky factorization. A positive answer is obtained when this factorization can be completed. Otherwise, a suitable $\mu > 0$ such that $A + \mu I$ is positive definite is obtained by a trial and error process: denoting by A_{ii} a diagonal element of A, we first try to use $\mu = -\min_i A_{ii} + 1$ if $\min_i A_{ii} \leq 0$ or $\mu = 1$ if $\min_i A_{ii} > 0$, and successively double its value in case of new Cholesky factorization failures. In our implementation, Cholesky factorizations are computed using the LAPACK [1] routine `dpotrf`. For Algorithm 2, if an objective F_j is strongly convex, then the algorithm proceeds in Step 1 with $\nabla^2 F_j$ without performing a Cholesky factorization.

For computing $d_{SD}(x)$ and $d_N(x)$, we solve subproblems (7) and (15), respectively, using Algencan [4], an augmented Lagrangian code for general nonlinear programming. Regarding the nonmonotone Armijo-type line search, without attempting to go into details, we mention that it was coded based on (quadratic) polynomial interpolations of the coordinate functions. The reader can find in [28] a careful discussion about line search strategies for vector optimization problems. Concerning the algorithmic parameters, we set $\sigma = 10^{-4}$, $\Gamma_1 = 10^{-6}$, $\Gamma_2 = 0.1$, and $\eta = 0.85$. The value for parameter η that controls the level of nonmonotonicity was chosen based on the suggestion of [32]. All runs were stopped at a point x declaring convergence whenever $\left|\theta(x)\right| \leq 5\times\texttt{eps}^{1/2}$, where $\texttt{eps} = 2^{-52} \approx 2.22\times10^{-16}$ is the machine precision and $\theta(x)$ corresponds to $\theta_N(x)$ or $\theta_{SD}(x)$, depending on the considered algorithm. We use $\theta_N(x)$ for Algorithm 2 and for the Newton method without safeguards, and $\theta_{SD}(x)$ for Algorithm 3 and for the steepest descent method. The maximum number of allowed iterations was set to 2000. We also stopped the execution of the algorithms if an error occurs in the solutions of the subproblems. The last two stopping criteria correspond to failures. Furthermore, for the Newton method without safeguards, a run that generated a non-descent direction was also counted as a failure.

5.2 Efficiency and Robustness Using CPU Time as the Performance Measurement

Since we are interested in estimating the Pareto frontier of a given problem, as mentioned in section 1, a strategy often used is to execute the algorithm at hand from several different starting points. Thus, for each problem, we considered 300 starting points from a uniform random distribution belonging to the corresponding boxes. Each instance was considered an independent problem and was solved by all algorithms. At this stage, a run was considered successful if an approximate critical point was found, regardless of the objective functions values. Although the problems are usually solved very quickly, we used the CPU time as the performance measurement. This means that, in the performance profile, $t_{p,s}$ is the computing time required to solve problem p by solver s. In the performance profile graphic, $\rho_s(1)$ is the the fraction of problems for which solver s was the most efficient over all the methods. Thus, the results in Figure 2 show that the Newton method without safeguards was the most efficient (44.4%), followed by Algorithm 3 (41.4%), Algorithm 2 (22.4%), and the steepest descent method (12.8%). These results are not surprising, given that, in general, an iteration of the Newton method without safeguards is

computationally cheaper than that of Algorithms 2 and 3, and that the steepest descent method does not use second-order information. On the other hand, as we are mainly interested in verifying the influence of the proposed globalization techniques, the most important result for us is related to robustness, i.e., verifying the capacity of the methods to solve the greatest possible number of problems. In a performance profile, robustness can be accessed on the extreme right of the graphic. As can be seen in Figure 2, Algorithms 2 and 3 are the most robust (99.3% and 98.3%, respectively), followed by the steepest descent method (86.8%) and the Newton method without safeguards (80.5%). We emphasize that in 7 of the 27 nonconvex problems (namely, Far1, Hil1, KW2, LE1, QV1, SK2, and VU1), the Newton method without safeguards solved less than half of the considered instances. In particular, for problems LE1 and QV1, it failed for all starting points. In turn, the steepest descent method failed in a considerable number of problems due to its numerical limitations, although it theoretically enjoys global convergence properties. These results indicate that the introduced safeguard strategies applied to second-order methods are useful in obtaining more robust schemes.

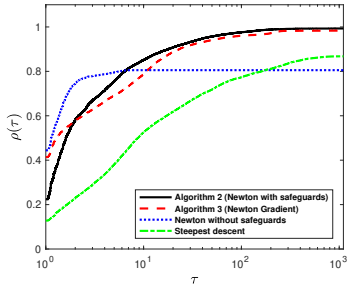

Figure 2: Performance profile using CPU time as the performance measurement considering 300 starting points for each test problem.

5.3 Pareto Frontiers

In the following, we compare the ability of the algorithms to properly generate Pareto frontiers. This will be done using the so-called *Purity* and (Γ and Δ) *Spread* metrics, which we briefly explain here for completeness. We refer the reader to [6] for a detailed explanation of these metrics. Let $PF_{p,s}$ be an approximation to the Pareto front obtained by solver s for problem p. Here, $PF_{p,s}$ was obtained by running a particular solver s from the same starting points considered in section 5.2 and then removing the dominated points. Now, let PF_p be an approximation to the Pareto front obtained by first considering $\cup_{s\in\mathcal{S}} PF_{p,s}$ and then removing the dominated points.

- Purity metric: The Purity metric measures the number of nondominated points belonging to PF_p that a solver is able to compute. Given a solver s and a problem p, it is defined by the ratio $\bar{t}_{p,s} := \left|PF_{p,s} \cap PF_p\right| / \left|PF_p\right|$. To analyze the Purity metric using the performance profile, we set $t_{p,s} := 1/\bar{t}_{p,s}$ and, thus, lower values of $t_{p,s}$ indicate better performances. If $\bar{t}_{p,s} = 0$, then we define $t_{p,s} := \infty$. As recommended in [6], we compared the algorithms in pairs when using the Purity metric.

- Spread metrics: A Spread metric seeks to measure the ability of a given solver to obtain *well-distributed* points along the Pareto frontier. Given a solver s and a problem p, consider that $PF_{p,s} \cap PF_p$ is formed by $x_1, \ldots, x_N$ and assume that these points are conveniently sorted by each objective function j such that $F_j(x_i) \leq F_j(x_{i+1})$, $i = 1, ..., N$. Let x_0 and x_{N+1} be the the points corresponding to the lowest and highest values, respectively, of F_j

obtained from PF_p. The metrics Γ and Δ are defined by

$$\Gamma_{p,s} := \max_{j\in\{1,\ldots,m\}} \max_{i\in\{0,\ldots,N\}} \delta_{i,j}$$

and

$$\Delta_{p,s} := \max_{j\in\{1,\ldots,m\}} \left(\frac{\delta_{0,j} + \delta_{N,j} + \sum_{i=1}^{N} |\delta_{i,j} - \bar{\delta}_j|}{\delta_{0,j} + \delta_{N,j} + (N-1)\bar{\delta}_j} \right),$$

where $\delta_{i,j} := \left|F_j(x_{i+1}) - F_j(x_i)\right|$ and $\bar{\delta}_j$, $j = 1, \ldots, m$, is the average of the distances $\delta_{i,j}$, $i = 1, ..., N$. In the performance profile, we set $t_{p,s} := \Gamma_{p,s}$ or $t_{p,s} := \Delta_{p,s}$, depending on the chosen metric.

Figure 3 shows the Purity performance profiles comparing the algorithms in pairs. It is easy to see that Algorithms 2 and 3 outperformed the Newton method without safeguards and the steepest descent method, respectively. This undoubtedly is related to the greater robustness of Algorithms 2 and 3, as shown in section 5.2. However, the superior performance of Algorithm 3 over Algorithm 2 may be surprising. Algorithm 2 and the steepest descent method achieved similar performances with respect to the Purity metric, not depicted here for the sake of brevity.

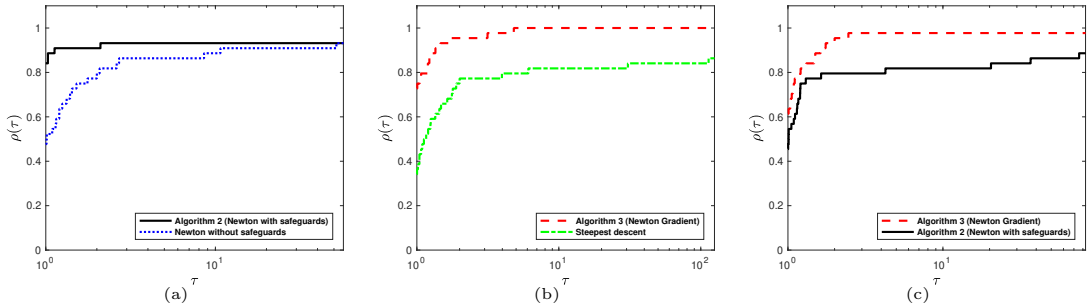

Figure 3: Purity performance profiles comparing: (a) Algorithm 2 and the Newton method without safeguards; (b) Algorithm 3 and the steepest descent method; (c) Algorithms 2 and 3.

The performance profiles of the Spread metrics Γ and Δ are shown in Figure 4. As can be seen, Algorithms 2 and 3 are slightly better than the others solvers for the Γ metric, whereas no significant difference is noticed for the Δ metric.

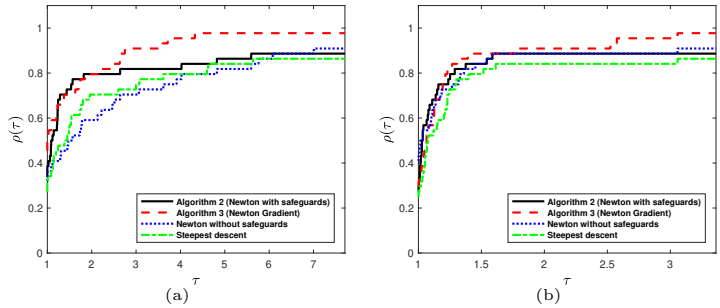

Figure 4: Spread performance profiles: (a) Γ metric; (b) Δ metric.

5.4 Larger Problem Instances

Finally, we illustrate the applicability of Algorithm 3 for solving problems with a larger number of objectives. MGH26, Toi9, and Toi10 are customizable problems in such a way that the

dimension n and the number of objectives m can be chosen by the user. For comparison purposes, we also tested Algorithm 2 on these problems. We considered some choices of n and m and a randomly generated starting point for each instance. Table 2 reports for each solver the number of iterations (it), the number of Hessians evaluations (nhev), the number of Cholesky factorizations (chol), and the CPU time (time). These results allow us to conclude that Algorithm 3 found approximately stationary points using less computational resources than Algorithm 2. Since Algorithm 3 needs to guarantee the positiveness of a single matrix, it required considerably less Cholesky factorizations than Algorithm 2 (which needs to guarantee the positiveness of m matrices). This explains the significant difference on the time required by these methods to stop.

			Algorithm 3				Algorithm 2			
Problem	n	m	it	nhev	chol	time	it	nhev	chol	time
MGH26	200	200	5	434	5	0.63	9	2000	2000	28.03
	400	400	5	916	5	4.13	9	4000	4000	301.91
	500	500	5	1142	5	8.59	9	5000	5000	612.09
Toi9	100	100	15	1057	16	0.51	6	700	700	4.08
	200	200	18	2574	25	3.28	7	1600	1600	60.53
	400	400	19	5617	26	38.37	9	4000	4000	792.26
Toi10	50	49	130	4729	154	2.38	711	34888	34888	72.80
	100	99	270	19761	390	28.77	214	21285	21500	447.43
	200	199	83	12527	93	64.27	89	17910	18090	2416.98

Table 2: Performance of Algorithms 2 and 3 on larger instances of problems MGH26, Toi9, and Toi10.

6 Final Remarks

In the present work, we introduced some new tools that can be used to globalize methods designed to solve nonconvex unconstrained multiobjective optimization problems. As a result, we first proposed a globally convergent version of the Newton method, extending its applicability to a larger class of multiobjective optimization problems. In particular, under local convexity assumptions, using the results of [12], it was proven that the sequence generated by the Newton algorithm converges superlinearly (or quadratically, in the case where the Hessians of the objectives are Lipschitz continuous) to a local efficient point of the given problem. We observe that, although the safeguard conditions (19) and (20) use the steepest descent direction, the proposed Newton scheme does not require its calculation. We further introduced the so-called Newton-Gradient method that uses the Lagrange multipliers arising from the steepest descent problem (7) with second-order information of the objective functions. The global convergence of the latter method was also established. Regarding the numerical experiments, it was observed that the modifications introduced in the classic Newton and steepest descent methods, described in Algorithms 2 and 3, led to an improvement in the robustness and in the ability to obtain *good* representations of the Pareto frontiers (measured by the Purity and Spread metrics) of the methods. In addition, in some problems with a large number of objectives, the Newton-Gradient method showed to be promising.

Recently in [27], the authors extended the concepts of Wolfe and Zoutendjik conditions for multiobjective optimization. In particular, under mild assumptions, they showed that the Zoutendjik condition holds, i.e.,

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

for a general line search method in which the step sizes satisfy the Wolfe conditions. It is

easy to see that the safeguard condition (19) together with the Zoutendjik condition implies that $\|d_{SD}(x^k)\|$ converges to zero. This means that, as in the scalar case (see [34]), globally convergent methods for multiobjective optimization can be obtained by taking the search directions as in (19) and using a line search satisfying the Wolfe conditions. Therefore, we expect that the techniques proposed here can also be valuable for improving the design and implementation of other algorithms for multiobjective optimization. In particular, it would be interesting to investigate rank-one and rank-two quasi-Newton methods.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, third edition, 1999.

[2] M. A. Ansary and G. Panda. A modified Quasi-Newton method for vector optimization problem. *Optimization*, 64(11):2289–2306, 2015.

[3] P. B. Assunção, O. P. Ferreira, and L. F. Prudente. Conditional gradient method for multiobjective optimization. *Comput. Optim. Appl.*, 2021.

[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] 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.

[7] Y.-H. Dai. Convergence properties of the BFGS algorithm. *SIAM J. Optim.*, 13(3):693–701, 2002.

[8] Y.-H. Dai. A perfect example for the BFGS method. *Math. Program.*, 138(1-2):501–530, 2013.

[9] 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.

[10] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2):201–213, 2002.

[11] 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.

[12] J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. Newton's method for multiobjective optimization. *SIAM J. Optim.*, 20(2):602–626, 2009.

[13] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Math. Metohd. Oper. Res.*, 51(3):479–494, 2000.

[14] E. H. Fukuda and L. M. Graña Drummond. Inexact projected gradient method for vector optimization. *Comput. Optim. Appl.*, 54(3):473–493, 2013.

[15] A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *J. Math. Anal. Appl.*, 22(3):618–630, 1968.

[16] 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.

[17] L. M. Graña Drummond and A. N. Iusem. A projected gradient method for vector optimization problems. *Comput. Optim. Appl.*, 28(1):5–29, 2004.

[18] 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.

[19] L. M. Graña Drummond and B. F. Svaiter. A steepest descent method for vector optimization. *J. Comput. Appl. Math.*, 175(2):395 – 414, 2005.

[20] C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *J. Optimiz. Theory App.*, 110(3):557–583, 2001.

[21] 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.

[22] Y. Jin, M. Olhofer, and B. Sendhoff. Dynamic weighted aggregation for evolutionary multiobjective 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.

[23] 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.

[24] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.*, 10(3):263–282, 2002.

[25] A. Lovison. Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis. *SIAM J. Optim.*, 21(2):463–490, 2011.

[26] D. T. Luc. Theory of vector optimization. Lectures Notes in economics and mathematical systems, vol. 319, 1989.

[27] L. R. Lucambio Pérez and L. F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM J. Optim.*, 28(3):2690–2720, 2018.

[28] L. R. Lucambio Pérez and L. F. Prudente. A Wolfe line search algorithm for vector optimization. *ACM Trans. Math. Softw.*, 45(4):23, 2019.

[29] W. F. Mascarenhas. The BFGS method with exact line searches fails for non-convex objective functions. *Math. Program.*, 99(1):49–61, 2004.

[30] W. F. Mascarenhas. On the divergence of line search methods. *Comput. Appl. Math.*, 26(1):129–169, 2007.

[31] 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.

[32] K. Mita, E. H. Fukuda, and N. Yamashita. Nonmonotone line searches for unconstrained multiobjective optimization problems. *J. Global Optim.*, 75:63–90, 2019.

[33] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *ACM Trans. Math. Softw.*, 7(1):17–41, Mar. 1981.

[34] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[35] Z. Povalej. Quasi-Newton method for multiobjective optimization. *J. Comput. Appl. Math.*, 255:765 – 777, 2014.

[36] 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 Berlin Heidelberg.

[37] S. Qu, Y. Ji, J. Jiang, and Q. Zhang. Nonmonotone gradient methods for vector optimization with a portfolio optimization application. *Eur. J. Oper. Res.*, 263(2):356 – 366, 2017.

[38] 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, Aug 2008.

[39] B. F. Svaiter. The multiobjective steepest descent direction is not Lipschitz continuous, but is Hölder continuous. *Oper. Res. Lett.*, 46(4):430 – 433, 2018.

[40] 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.

[41] 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.