



**HAL**  
open science

# New computable algorithms for smooth multiobjective optimization problems

Sorin-Mihai Grad, Tibor Illés, Petra Renáta Rigó

► **To cite this version:**

Sorin-Mihai Grad, Tibor Illés, Petra Renáta Rigó. New computable algorithms for smooth multiobjective optimization problems. 2024. hal-04680633

**HAL Id: hal-04680633**

**<https://hal.science/hal-04680633v1>**

Preprint submitted on 28 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# New computable algorithms for smooth multiobjective optimization problems

Sorin-Mihai Grad\*    Tibor Illés<sup>†</sup>    Petra Renáta Rigó<sup>‡</sup>

August 28, 2024

## Abstract

We propose new practical algorithms for solving smooth multiobjective optimization problems based on determining joint decreasing directions via suitable linear programming problems. The presented iterative method is specialized for unconstrained, sign constrained and linearly constrained multiobjective optimization problems. In all cases we show that the objective function values sequence is decreasing with respect to the considered nonnegative orthant while the iterates are feasible. Furthermore, we prove that every accumulation point of the sequence generated by the algorithm, if any, is a substationary point to the considered multiobjective optimization problem, and, under convexity assumptions, it is actually a weakly Pareto efficient (also known as weakly Pareto-optimal) point. Different to similar algorithms from the literature, the ones proposed in this work involve joint decreasing directions that are easily computable in polynomial time by solving linear programming problems. The computational performance of our algorithms has been illustrated on convex unconstrained and convex linearly constrained multiobjective optimization problems.

**Keywords.** Continuous Optimization, multiple objective programming, joint decreasing direction, linear programming, (weakly) Pareto efficient solution

## 1 Introduction

Multiobjective optimization problems consist in minimizing several objective functions at the same time, such problems arising in different fields, such as engineering [9], statistics [6], management science [10], etc. The most used solution notions for such problems are

---

\*Corresponding Author. Unité de Mathématiques Appliquées, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France. E-mail: sorin-mihai.grad@ensta-paris.fr, ORCID-ID: 0000-0002-1139-7504

<sup>†</sup>Corvinus Centre for Operations Research, Corvinus University of Budapest, Hungary. E-mail: tiber.illes@uni-corvinus.hu.

<sup>‡</sup>Corvinus Centre for Operations Research, Corvinus University of Budapest, Hungary. E-mail: petra.rigo@uni-corvinus.hu.

the Pareto and weakly Pareto efficient ones, see [4,23]. A point is called Pareto efficient if there does not exist another point with the same or smaller objective function value (with respect to the corresponding nonnegative orthant), such that there is a strict decrease in at least one objective function value. A point is called weakly Pareto efficient if there does not exist another point with strict decrease in all its objective values. These notions have also been extended to cases when other ordering cones are used instead of the nonnegative orthant and even to problems with infinitely-dimensional image space [4].

One of the main approaches to handle multiobjective optimization problems is the scalarization technique, see [11,23,25]. However, the scalar optimization problem attached to a multiobjective one often turns out to be unbounded (see, for instance [5]), providing thus no valuable input in solving the original problem. This motivated the introduction of algorithms capable of directly solving multiobjective problems, most of which being direct extensions of iterative methods for tackling scalar optimization problems, such as the steepest descent method [19], the Barzilai-Borwein descent method [7], the conjugate gradient method [24], different versions of the projected gradient method [2,14,17], Newton's method [12], and even the proximal point algorithm [5,16] in the nonsmooth case. All these algorithms have a common feature in that there is a decrease (with respect to the partial ordering induced by the underlying cone) in the vector objective value at each iteration. Convergence results for the sequences produced by these methods were provided under different assumptions. While the determination of the descent direction plays a key role in these algorithms, in most of the contributions to this direction this step has only a theoretical value and, consequently, no numerical results were provided. Moreover, inexact versions of some of these methods were proposed in the literature in order to facilitate their implementation. Our aim is to propose new algorithms for solving different types of multiobjective optimization problems, where the descent directions are easily computable in polynomial time.

These new algorithms for solving multiobjective optimization problems are based on an idea by Illés and Lovics [20] from 2018. They proposed some techniques for determining joint decreasing directions for both unconstrained and linearly constrained multiobjective optimization problems. Afterward, a subdivision technique was employed for approximating the whole weakly Pareto efficient set of linearly constrained convex multiobjective optimization problems, and these theoretical achievements were illustrated by computational results obtained while solving the classical Markowitz model.

We propose a generic algorithm for solving multiobjective optimization problems where joint decreasing directions are used in order to guarantee descending values of the objective vector function. We discuss afterward the differences between it and other similar algorithms considered in the literature. In particular, we show that joint decreasing directions for these problems can be computed in polynomial time using results from linear programming, making thus our method practical in the sense that numerical results can be obtained easily for it. Then, we specialize the proposed algorithm for unconstrained, sign constrained and linearly constrained multiobjective optimization problems. In order to illustrate the applicability of the proposed methods, we provide numerical results. From the test examples available in the literature we chose a convex unconstrained multiobjective optimization problem to illustrate how our algorithm works. After that, we modified the selected test problem by adding linear constraints in such a way, that some

of the weakly Pareto efficient points became infeasible for the constrained problem. The computational results in the constrained case show the effect of the linear constraints both on the running time and on the obtained weakly Pareto efficient set.

The paper is organized as follows. In Section 2 some basic concepts and definitions related to multiobjective/vector optimization problems are presented. In Section 3 we propose a generic scheme based on the computation of joint decreasing directions for solving multiobjective optimization problems, and we point out some differences between it and some of the similar algorithms from the literature. In Section 4, the new algorithm for unconstrained multiobjective optimization problems is defined. Section 5 is devoted to the algorithm for multiobjective optimization problems with sign constraints, while in Section 6, the new algorithm variant for linearly constrained multiobjective optimization problem is discussed. In all cases we show that the objective vector function values sequence is decreasing with respect to the natural partial ordering of  $\mathbb{R}^m$ . Furthermore, we prove that every accumulation point of the generated sequence, if any, is a substationary point, and, under convexity hypotheses imposed on the objective vector function and on the feasible set, a weakly Pareto efficient one. In Section 7 we provide numerical results in order to show the efficacy and efficiency of the new algorithms. Section 8 contains concluding remarks and ideas for future research.

We use the following notations throughout the paper. Let  $\mathbb{R}_{\oplus}^m$  and  $\mathbb{R}_+^m$  be the nonnegative and the positive orthant in the  $m$ -dimensional Euclidean space ( $m \geq 2$ ), respectively. We consider the partial ordering induced by  $\mathbb{R}_+^m$ , i.e. for  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ ,  $\mathbf{x} \leq \mathbf{y}$  if  $\mathbf{y} - \mathbf{x} \in \mathbb{R}_{\oplus}^m$  and  $\mathbf{x} < \mathbf{y}$  if  $\mathbf{y} - \mathbf{x} \in \mathbb{R}_+^m$ . We write  $\mathbf{x} \leq \mathbf{y}$  when  $\mathbf{x} \leq \mathbf{y}$  and  $\mathbf{x} \neq \mathbf{y}$ . We also use analogously for  $m$ -dimensional vectors the notations  $\geq$ ,  $\geq$  and  $>$ . For real numbers we use the standard notations  $<$ ,  $>$ ,  $\leq$ , and  $\geq$ . Vectors are denoted by lowercase boldface Latin letters,  $\mathbf{e}$  is the  $n$ -dimensional all-one vector. Furthermore,  $\|\cdot\|$  denotes the standard Euclidean norm. Let  $U \subseteq \mathbb{R}^n$  open and  $F = (F_1, \dots, F_m)^\top : U \rightarrow \mathbb{R}^m$  continuously differentiable, i.e.  $F \in C^1(U, \mathbb{R}^m)$ . The Jacobian of  $F$  at  $x \in U$  is denoted by  $JF(x) \in \mathbb{R}^{m \times n}$  and the Hessian of  $F_j$  ( $j = 1, \dots, m$ ) at  $x \in U$  is  $\nabla^2 F_j(x)$ . For simplicity, when  $F_j$  is convex (on  $U$ ),  $j = 1, \dots, m$ , instead of calling  $F$   $\mathbb{R}_{\oplus}^m$ -convex (on  $U$ ) we say that it is convex (on  $U$ ).

## 2 Basic results in multiobjective optimization

In this section we introduce the general multiobjective optimization problem and the concepts of (weakly) Pareto efficient solutions. Let  $\mathcal{F} \subseteq \mathbb{R}^n$  be closed with nonempty interior and  $F = (F_1, \dots, F_m)^\top : \mathcal{F} \rightarrow \mathbb{R}^m$ . The general multiobjective optimization problem we consider is

$$\left. \begin{array}{l} \text{Min } F(\mathbf{x}) \\ \mathbf{x} \in \mathcal{F}. \end{array} \right\} \quad (GMOP)$$

From now on we assume  $F$  to be continuously differentiable on  $\text{int } \mathcal{F}$ . We call  $(GMOP)$  convex when  $F$  and  $\mathcal{F}$  are convex. We say that  $\bar{\mathbf{x}} \in \mathcal{F}$  is a *substationary point* to  $(GMOP)$  if there exists a  $\mathbf{w} \in \mathcal{S}_m := \{\mathbf{w} \in \mathbb{R}_{\oplus}^m : \mathbf{e}^\top \mathbf{w} = 1\}$ , which satisfies  $\mathbf{w}^\top [JF(\bar{\mathbf{x}})] = \mathbf{0}$ . This notion is defined by means of a necessary condition for the weak Pareto efficiency of a point to  $(GMOP)$  and its origin can be traced back to [22], arguably the first paper on multiobjective optimization.

We say that  $\bar{\mathbf{x}} \in \mathcal{F}$  is a *stationary (critical) point to (GMOP)* if  $\mathcal{R}(JF(\bar{\mathbf{x}})) \cap (-\mathbb{R}_+^m) = \emptyset$ , where  $\mathcal{R}(\cdot)$  denotes the image set of the considered mapping. Hence,  $\bar{\mathbf{x}} \in \mathcal{F}$  is stationary to (GMOP) if and only if for all  $\mathbf{v} \in \mathbb{R}^n$  we have  $JF(\bar{\mathbf{x}})\mathbf{v} \not\leq 0$ , that is for every  $\mathbf{v} \in \mathbb{R}^n$  there exists a  $j = j(\mathbf{v})$  such that

$$(JF(\bar{\mathbf{x}})\mathbf{v})_j = \nabla F_j(\bar{\mathbf{x}})^\top \mathbf{v} \geq 0,$$

and this implies

$$\max_{i=1,\dots,m} \nabla F_i(\bar{\mathbf{x}})^\top \mathbf{v} \geq 0 \quad \forall \mathbf{v} \in \mathbb{R}^n.$$

**Definition 2.1** We say that  $\mathbf{x}^* \in \mathcal{F}$  is a

- *weakly Pareto efficient solution to (GMOP) if no feasible solution  $\mathbf{x} \in \mathcal{F}$  exists satisfying  $F(\mathbf{x}) < F(\mathbf{x}^*)$ ;*
- *Pareto efficient solution to (GMOP) if no feasible solution  $\mathbf{x} \in \mathcal{F}$  exists satisfying  $F(\mathbf{x}) \leq F(\mathbf{x}^*)$ .*

Pareto solutions to (GMOP) are also weakly Pareto ones to it, while the opposite implication does not hold in general. For a discussion on how to guarantee the latter, see [5, Section 4]. Since 1952 several methods have been proposed for finding one of the (weakly) Pareto efficient solutions to (GMOP). Markowitz considered the following weighted optimization problem

$$\left. \begin{array}{l} \min \mathbf{w}^\top F(\mathbf{x}) \\ \mathbf{x} \in \mathcal{F} \end{array} \right\} \quad (WOP(\mathbf{w})),$$

where  $\mathbf{w} \in \mathcal{S}_m$ . The next statement shows that (WOP( $\mathbf{w}$ )) can be used for determining the weakly Pareto efficient solutions to (GMOP), see [4, 8, 23, 25].

**Theorem 2.1** Let (GMOP) and the corresponding (WOP( $\mathbf{w}$ )) be given for a  $\mathbf{w} \in \mathcal{S}_m$ . Assume that  $\mathbf{x}^* \in \mathcal{F}$  is an optimal solution to (WOP( $\mathbf{w}$ )), then  $\mathbf{x}^*$  is a weakly Pareto efficient solution to (GMOP).

For the completeness of the discussion we mention the following theorem, which shows that in the convex case each Pareto efficient solution to (GMOP) can be determined by solving (WOP( $\mathbf{w}$ )) defined by means of a suitable weights vector  $\mathbf{w} \in \mathcal{S}_m \cap \mathbb{R}_+^m$ , see [8, 23, 25].

**Theorem 2.2** Let (GMOP) be a convex multiobjective optimization problem, and assume that  $\mathbf{x}^* \in \mathcal{F}$  is a Pareto efficient solution to (GMOP). Then there is a weight vector  $\mathbf{w} \in \mathcal{S}_m \cap \mathbb{R}_+^m$  and corresponding (WOP( $\mathbf{w}$ )) whose optimal solution is  $\mathbf{x}^*$ .

**Remark 1** A point  $\bar{\mathbf{x}} \in \mathcal{F}$  is *substationary to (GMOP)* if and only if there exists a  $\mathbf{w} \in \mathcal{S}_m$  such that  $\mathbf{w}^\top \nabla F(\bar{\mathbf{x}}) = \mathbf{0}$ . This is a necessary condition for the local optimality of  $\bar{\mathbf{x}}$  to (WOP( $\mathbf{w}$ )) and becomes sufficient for its (global) optimality when  $F$  is convex (on  $\mathcal{F}$ ) and  $\mathcal{F}$  is convex. Hence, by Theorem 2.1, when (GMOP) is convex, a substationary point to it is weakly Pareto efficient to it.

Now we present the definition of joint decreasing directions which will be used in this paper.

**Definition 2.2** (Illés and Lovics [20], see also [26]) *Given (GMOP) and a feasible point  $\mathbf{x} \in \mathcal{F}$ , a vector  $\mathbf{v} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$  is said to be a*

- *joint decreasing direction of  $F$  at point  $\mathbf{x}$  if there exists  $h_1 > 0$ , such that, for every  $h \in (0, h_1]$  one has  $F(\mathbf{x} + h\mathbf{v}) < F(\mathbf{x})$ ;*
- *feasible joint decreasing direction of  $F$  if it is a joint decreasing direction of  $F$  at  $\mathbf{x}$  and there exists  $h_2 > 0$ , such that, for every  $h \in (0, h_2]$ ,  $\mathbf{x} + h\mathbf{v} \in \mathcal{F}$  holds as well.*

**Remark 2** *A vector  $\mathbf{v} \in \mathbb{R}^n$  is a joint decreasing direction of  $F$  if and only if  $[JF(\mathbf{x})] \mathbf{v} < \mathbf{0}$ .*

Let  $\mathcal{D}_F(\mathbf{x}) = \{\mathbf{v} \in \mathbb{R}^n : [JF(\mathbf{x})] \mathbf{v} < \mathbf{0}\}$  be the set of all joint decreasing directions of  $F$  at  $\mathbf{x}$ . A method for determining them relies on employing the quadratic programming problem

$$\min_{\mathbf{w} \in \mathcal{S}_m} \mathbf{w}^\top \left( JF(\mathbf{x}) [JF(\mathbf{x})]^\top \right) \mathbf{w} \quad (QOP(\mathbf{x})).$$

**Theorem 2.3** (Schäffler et al. [26], see [20] for a more direct proof) *Let (GMOP), a point  $\mathbf{x} \in \mathbb{R}^n$ , and the associated (QOP( $\mathbf{x}$ )) be given. Let  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be a continuously differentiable (and convex) vector function, and  $\mathbf{w}^* \in \mathbb{R}^m$  denote the optimal solution to (QOP( $\mathbf{x}$ )). We define the vector  $\mathbf{q} = [JF(\mathbf{x})]^\top \mathbf{w}^* \in \mathbb{R}^n$ . If  $\mathbf{q} = \mathbf{0}$ , then  $\mathbf{x}$  is a substationary (weakly Pareto efficient) point to (GMOP), otherwise  $-\mathbf{q}$  is a joint decreasing direction of  $F$  at point  $\mathbf{x}$ .*

A similar outcome can be reached by introducing, for  $\mathbf{x} \in \mathbb{R}^n$ , the following linear programming problem

$$\left. \begin{array}{l} \max_{(\mathbf{q}, q_0) \in \mathbb{R}^n \times \mathbb{R}} q_0 \\ [JF(\mathbf{x})] \mathbf{q} + q_0 \mathbf{e} \leq \mathbf{0} \\ 0 \leq q_0 \leq 1 \end{array} \right\} \quad (LP(\mathbf{x})).$$

**Theorem 2.4** (Illés and Lovics [20]) *Let (GMOP), a point  $\mathbf{x} \in \mathbb{R}^n$  and the associated (LP( $\mathbf{x}$ )) be given. Let  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be a continuously differentiable (and convex) vector function. Then (LP( $\mathbf{x}$ )) always has an optimal solution  $(\mathbf{q}^*, q_0^*)$ . There are two cases for the optimal value of (LP( $\mathbf{x}$ )), either  $q_0 = 0$  when  $\mathbf{x}$  is a substationary (weakly Pareto efficient) point to (GMOP), or  $q_0 = 1$  in which case  $\mathbf{q}^*$  is a joint decreasing direction of  $F$  at  $\mathbf{x}$ .*

---

**Algorithm 3.1** Generic scheme for solving GMOPs

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} = \mathbb{R}^n$ ,  $0 < \beta < 1$ ,  $k = 0$ ,  $J = \{\frac{1}{2^n} : n \in \mathbb{N}\}$   
[Step 2] **if**  $\mathcal{D}_F(\mathbf{x}^k) = \emptyset$  **then** STOP ( $\mathbf{x}^k$  is a substationary solution to (GMOP));  
          **otherwise** select a *joint decreasing direction*  $\mathbf{v}^k \in \mathcal{D}_F(\mathbf{x}^k)$ , compute  $\mathbf{u}^k = JF(\mathbf{x}^k)\mathbf{v}^k$   
[Step 3] choose  $t_k$  as the largest  $t \in J$  s.t.  $F(\mathbf{x}^k + t\mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t\mathbf{u}^k$   
[Step 4] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k\mathbf{v}^k$  and  $k = k + 1$ ; go to Step 2

---

### 3 Generic scheme for solving general multiobjective optimization problems

Based on the above discussion, we propose in Algorithm 3.1 a generic scheme for solving GMOPs that makes use of joint decreasing directions.

In the following subsections we recall several algorithms for solving GMOPs from the literature. The key difference between these methods is in the second step of Algorithm 3.1, namely the way how the descent directions are defined. In each case we will point out what is this step.

#### 3.1 Steepest descent algorithm

In this subsection we focus on the steepest descent algorithm introduced by Fliege and Svaiter [11] for solving unconstrained multicriteria optimization problems. It is proven that the method converges to a critical point to (GMOP). In [19], the algorithm is refined in order to solve smooth unconstrained multiobjective optimization problems, where the vector-minimization is considered with respect to a closed convex ordering cone. The authors prove that every accumulation point of the generated sequence is critical. The sketch of the steepest descent algorithm used in these papers is presented in Algorithm 3.2.

---

**Algorithm 3.2** Steepest descent algorithm

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} = \mathbb{R}^n$ ,  $0 < \beta < 1$ ,  $k = 0$ ,  $J = \{\frac{1}{2^n} : n \in \mathbb{N}\}$   
[Step 2] determine the steepest descent direction  $\mathbf{v}^k = \operatorname{argmin}_{\mathbf{v} \in \mathbb{R}^n} \{\varphi_{\mathbf{x}^k}(\mathbf{v}) + \frac{1}{2}\|\mathbf{v}\|^2\}$  and  $\theta_k = \varphi_{\mathbf{x}^k}(\mathbf{v}^k) + \frac{1}{2}\|\mathbf{v}^k\|^2$ , where  $\varphi_{\mathbf{x}}(\mathbf{v}) = \max_{j=1, \dots, m} \nabla F_j(\mathbf{x})^\top \mathbf{v}$ ; if  $\theta_k = 0$  : STOP  
[Step 3] choose  $t_k$  as the largest  $t \in J$  s.t.  $F(\mathbf{x}^k + t\mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t JF(\mathbf{x}^k)\mathbf{v}^k$   
[Step 4] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k\mathbf{v}^k$  and  $k = k + 1$ ; go to Step 2

---

Step 2 of Algorithm 3.1 and its counterpart in Algorithm 3.2 are different, as the first one delivers a joint decreasing direction, while the other a steepest descent direction obtained by solving the minimax quadratic optimization problem  $\inf_{\mathbf{v} \in \mathbb{R}^n} \{\varphi_{\mathbf{x}^k}(\mathbf{v}) + \frac{1}{2}\|\mathbf{v}\|^2\}$ . Moreover, Algorithm 3.2 requires the additional computation of  $\theta_k$ ,  $k \geq 0$ . Last but not least, the stopping criteria of the two algorithms differ as well.

## 3.2 Projected gradient method

A projected gradient algorithm for solving smooth multiobjective optimization problems consisting in vector-minimizing over a closed convex set  $\mathcal{F} \subseteq \mathbb{R}^n$  a vector function  $F : \mathcal{F} \rightarrow \mathbb{R}^m$  with respect to the partial ordering induced by a closed convex cone was proposed by Graña-Drummond and Iusem [17]. They showed that any accumulation point of the sequence generated by this method converges to a stationary point to the considered problem. In the convex case and under some mild assumptions they also show convergence of the sequence generated by the algorithm to a weakly Pareto efficient solution to the considered problem. Improvements and extensions of the method are proposed in [14,15]. The sketch of the method adapted to the present framework is presented in Algorithm 3.3.

---

### Algorithm 3.3 Projected gradient algorithm

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} \subseteq \mathbb{R}^n$ ,  $0 < \beta < 1$ ,  $\sigma > 0$ ,  $k = 0$ ,  $J = \left\{ \frac{1}{2^n} : n \in \mathbb{N} \right\}$   
[Step 2] determine the descent direction  $\mathbf{v}^k = \operatorname{argmin}_{\mathbf{v} \in \mathcal{F} - \mathbf{x}^k} \left\{ \sigma \varphi_{\mathbf{x}^k}(\mathbf{v}) + \frac{1}{2} \|\mathbf{v}\|^2 \right\}$  and  $\theta_k = \varphi_{\mathbf{x}^k}(\mathbf{v}^k) + \frac{1}{2} \|\mathbf{v}^k\|^2$ , where  $\varphi_{\mathbf{x}}(\mathbf{v}) = \max_{j=1, \dots, m} \nabla F_j(\mathbf{x})^\top \mathbf{v}$ ; if  $\theta_k = 0$  : STOP  
[Step 3] choose  $t_k$  as the largest  $t \in J$  s.t.  $F(\mathbf{x}^k + t\mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t J F(\mathbf{x}^k) \mathbf{v}^k$   
[Step 4] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{v}^k$  and  $k = k + 1$ ; go to Step 2

---

The differences between Algorithm 3.1 and Algorithm 3.3 are basically the same like between the first one and Algorithm 3.2, with the additional constraint that the steepest descent direction in Step 2 of Algorithm 3.3 is chosen so that the new iterate is feasible.

## 3.3 Conjugate gradient algorithm

Lucambio Pérez and Prudente [24] have recently introduced nonlinear conjugate gradient methods for finding critical points of vector functions with respect to a partial ordering induced by a closed convex pointed cone with nonempty interior. No convexity assumption is imposed on the objective vector function. Under inexact line search, the authors prove that the sequences generated by the method find critical points to (*GMOP*). The sketch of their basic algorithm is given in Algorithm 3.4, particularized for the framework considered in our paper.

---

### Algorithm 3.4 Conjugate gradient algorithm

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} \subseteq \mathbb{R}^n$ ,  $0 < \beta < 1$ ,  $(\sigma_k)_k > 0$ ,  $k = 0$ ,  $J = \left\{ \frac{1}{2^n} : n \in \mathbb{N} \right\}$   
[Step 2] determine the descent direction  $\mathbf{v}^k = \operatorname{argmin}_{\mathbf{v} \in \mathbb{R}^n} \left\{ \varphi_{\mathbf{x}^k}(\mathbf{v}) + \frac{1}{2} \|\mathbf{v}\|^2 \right\}$ , where  $\varphi_{\mathbf{x}}(\mathbf{v}) = \max_{j=1, \dots, m} \nabla F_j(\mathbf{x})^\top \mathbf{v}$ ; if  $\mathbf{v}^k = \mathbf{0}$  : STOP; define  $\mathbf{d}^k = \mathbf{d}^{k-1} + \sigma_k \mathbf{v}^k$  ( $\mathbf{d}^0 = \mathbf{v}^0$ )  
[Step 3] choose  $t_k$  as the largest  $t \in J$  s.t.  $F(\mathbf{x}^k + t\mathbf{d}^k) \leq F(\mathbf{x}^k) + \beta t J F(\mathbf{x}^k) \mathbf{d}^k$   
[Step 4] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{d}^k$  and  $k = k + 1$ ; go to Step 2

---

Here we can also see that when comparing Algorithm 3.1 and Algorithm 3.4 the key difference can be found in Step 2, where the descent direction is computed in each case in a specific way. Furthermore, the use of  $(\sigma_k)_k > 0$  can be also noticed, and the stopping criteria differ.

### 3.4 Newton's method

In this subsection we present the extension of Newton's method proposed by Fliege, Graña-Drummond and Svaiter [12] for solving GMOPs. The objective vector functions are supposed to be twice continuously differentiable and locally strongly convex. Under these hypotheses the method is locally superlinearly convergent to Pareto efficient points when the objective vector function is cone-convex. Moreover, in [18] an improvement is proposed, where quadratic convergence of the generated sequence is achieved. Below we present the scheme of multiobjective Newton's method in Algorithm 3.5.

---

#### Algorithm 3.5 Newton's method

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} \subseteq \mathbb{R}^n$ ,  $0 < \beta < 1$ ,  $k = 0$ ,  $J = \left\{ \frac{1}{2^n} : n \in \mathbb{N} \right\}$

[Step 2] determine the Newton direction  $\mathbf{s}^k = \operatorname{argmin}_{\mathbf{s} \in \mathbb{R}^n} \max_{j=1, \dots, m} \left\{ \nabla F_j(\mathbf{x}^k)^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \nabla^2 F_j(\mathbf{x}^k) \mathbf{s} \right\}$

and  $\theta_k = \inf_{\mathbf{s} \in \mathbb{R}^n} \max_{j=1, \dots, m} \left\{ \nabla F_j(\mathbf{x}^k)^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \nabla^2 F_j(\mathbf{x}^k) \mathbf{s} \right\}$ ; if  $\theta_k = 0$  : STOP

[Step 3] choose  $t_k$  as the largest  $t \in J$  s.t.  $\begin{cases} \mathbf{x}^k + t\mathbf{s}^k \in \mathcal{F} \\ F_j(\mathbf{x}^k + t\mathbf{s}^k) \leq F_j(\mathbf{x}^k) + \beta t\theta_k, \quad j = 1, \dots, m \end{cases}$

[Step 4] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{s}^k$  and  $k = k + 1$ ; go to Step 2

---

Even though Newton's algorithm is a second order method, while the other ones considered in this paper are first order ones, its general scheme is consistent with the others. Of course, Step 2 in Algorithm 3.5 is not the one in Algorithm 3.1 since Newton's direction is obtained by minimizing the max-ordering scalarization of the variations on the quadratic approximation of the objective vector function. Furthermore, Step 3 and the stopping criteria differ in this case, too.

To sum up, the algorithms presented in this section show that the determination of the descent directions and the solutions of the subproblems appearing in Step 2 (in each case) play a crucial role in these methods. However, out of the previously mentioned papers only in few cases one can read about numerical results or implementation techniques, and in those cases inexact versions of these methods were considered. Our aim is to introduce new algorithms based on Algorithm 3.1 where the subproblems are easily solvable in an efficient and exact manner. In the following section we propose a new algorithm for solving unconstrained multiobjective optimization problems. After that, we present other variants of the new algorithm for solving smooth multiobjective optimization problems with sign constraints and linear constraints, respectively.

## 4 New algorithm for solving unconstrained multiobjective optimization problems

In this section we present the new algorithm we propose for solving unconstrained multiobjective optimization problems which involves linear programming subproblems. The main steps of the method are given in Algorithm 4.1.

Following the result of Illés and Lovics [20] we examine the case (using notations from Step 2 of Algorithm 4.1)

$$[JF(\mathbf{x})]\mathbf{q} + q_0 \mathbf{e} \leq \mathbf{0}, \quad q_0 > 0. \quad (1)$$

---

**Algorithm 4.1** New algorithm for solving unconstrained multiobjective optimization problems

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} = \mathbb{R}^n$ ,  $0 < \beta < 1$ ,  $k = 0$ ,  $J = \left\{ \frac{1}{2^n} : n \in \mathbb{N} \right\}$

[Step 2] determine the *joint decreasing direction*  $\mathbf{q} \in \mathbb{R}^n$  by solving

$$\begin{aligned} \max_{(\mathbf{q}, q_0) \in \mathbb{R}^n \times \mathbb{R}} \quad & q_0 \\ JF(\mathbf{x}^k)\mathbf{q} + q_0 \mathbf{e} \leq & \mathbf{0} \\ 0 \leq q_0 \leq & 1 \end{aligned}$$

[Step 3] if  $q_0 = 0$  then STOP ( $\mathbf{x}^k$  is a substationary solution to (GMOP));  
otherwise  $\mathbf{v}^k := \mathbf{q}$ ;

[Step 4] choose  $t_k$  as the largest  $t \in J$  s.t.  $F(\mathbf{x}^k + t\mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t JF(\mathbf{x}^k)\mathbf{v}^k$

[Step 5] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{v}^k$  and  $k = k + 1$ ; go to Step 2

---

If system (1) has a solution  $(\mathbf{q}, q_0)$ , then  $((1/q_0)\mathbf{q}, 1)$  solves it, too, so the optimal value of the objective function in Step 2 of Algorithm 4.1 is 1, in which case  $q_0 = 1$  in Step 3. This means that

$$[JF(\mathbf{x})]\mathbf{q} \leq -\mathbf{e}$$

so  $\mathbf{q}$  is a *joint decreasing direction* of the vector function  $F$ .

If system (1) has no solution, then the optimal value of the objective function is 0 (as  $(\mathbf{0}, 0)$  is feasible to the considered linear programming problem), and from a variant of the *Farkas Lemma* (see [20]) we know that there exists a  $\mathbf{w} \in \mathbb{R}^m$  which satisfies the following equality system

$$\mathbf{w}^\top [JF(\mathbf{x})] = \mathbf{0}, \quad \mathbf{e}^\top \mathbf{w} = 1, \quad \mathbf{w} \geq \mathbf{0}. \quad (2)$$

This means that if the optimal value of the problem ( $LP(\mathbf{x})$ ) is 0, then  $\mathbf{x}$  is a *substationary point* to (GMOP).

**Lemma 4.1** *Let (GMOP) be given with  $\mathcal{F} = \mathbb{R}^n$  and the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  continuously differentiable. Assuming that Algorithm 4.1 starting at  $\mathbf{x}^0 \in \mathcal{F}$  produces for  $0 < \beta < 1$  the sequence of points  $\{\mathbf{x}^k\}$ , then*

$$F(\mathbf{x}^{k+1}) < F(\mathbf{x}^k), \quad \forall k \geq 0. \quad (3)$$

**Proof.** Let  $k \geq 0$ . Since  $\mathbf{v}^k$  is a joint decreasing direction computed in Step 2 of Algorithm 4.1, we have  $[JF(\mathbf{x}^k)]\mathbf{v}^k \leq -\mathbf{e}$ . Therefore, in Step 4, choose  $t_k$  as the largest  $t \in J$  such that

$$F(\mathbf{x}^k + t\mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t JF(\mathbf{x}^k)\mathbf{v}^k \leq F(\mathbf{x}^k) - \beta t \mathbf{e},$$

where  $t > 0$ . Selecting  $t_k$  and restructuring the previous inequality, we get

$$F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) \leq -\beta t_k \mathbf{e} < \mathbf{0}, \quad (4)$$

where  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{v}^k$  from Step 5 of Algorithm 4.1. Thus we proved (3). ■

In the following statement we show that the objective vector function values sequence  $\{F(\mathbf{x}^k)\}$  generated by Algorithm 4.1 is  $\mathbb{R}_\oplus^m$ -decreasing.

**Theorem 4.2** *Let (GMOP) be given with  $\mathcal{F} = \mathbb{R}^n$ , and the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  continuously differentiable (and convex). Assume that Algorithm 4.1 starting at  $\mathbf{x}^0 \in \mathcal{F}$  produces for  $0 < \beta < 1$  the sequence of points  $\{\mathbf{x}^k\}$ . Then, every accumulation point of  $\{\mathbf{x}^k\}$ , if any, is a substationary (weakly Pareto efficient) point to (GMOP).*

**Proof.** Let  $\bar{\mathbf{x}}$  be an accumulation point of the sequence  $\{\mathbf{x}^k\}$ . We should consider now a convergent subsequence of  $\{\mathbf{x}^k\}$ , but for simplicity we identify it with the whole sequence. Let  $k \geq 0$ . We have

$$\lim_{k \rightarrow \infty} \left( F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) \right) = F(\bar{\mathbf{x}}) - F(\bar{\mathbf{x}}) = 0 \quad (5)$$

and from (4) we get

$$F(\mathbf{x}^k) - F(\mathbf{x}^{k+1}) \geq \beta t_k \mathbf{e} > 0. \quad (6)$$

From (5) and (6) follows that

$$\lim_{k \rightarrow \infty} t_k = 0. \quad (7)$$

On the other hand,

$$0 = \lim_{k \rightarrow \infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| = \lim_{k \rightarrow \infty} \|\mathbf{x}^k + t_k \mathbf{v}^k - \mathbf{x}^k\| = \lim_{k \rightarrow \infty} \|t_k \mathbf{v}^k\|. \quad (8)$$

From (7) and (8) it follows that  $\{\mathbf{v}^k\}$  is bounded. Hence, there exists an accumulation point  $\bar{\mathbf{v}} \in \mathbb{R}^n$  of the sequence of joint decreasing directions  $\{\mathbf{v}^k\}$  generated by Algorithm 4.1. Therefore, there exists a subsequence  $\{\mathbf{v}^{k_j}\}$  such that

$$\lim_{k_j \rightarrow \infty} \mathbf{v}^{k_j} = \bar{\mathbf{v}}, \quad \text{and} \quad \lim_{k_j \rightarrow \infty} t_{k_j} = 0.$$

This means that for a fixed quite large  $j_0 \in \mathbb{N}$  there exists a positive integer  $\gamma > 1$  such that for all  $j > j_0$  we have

$$t_{k_j} < \frac{1}{2^\gamma}, \quad (9)$$

which means that the inequality in Step 4 of the Algorithm 4.1 is not satisfied for  $t = \frac{1}{2^\gamma}$ , hence

$$F\left(\mathbf{x}^{k_j} + \frac{1}{2^\gamma} \mathbf{v}^{k_j}\right) \not\leq F(\mathbf{x}^{k_j}) + \frac{\beta}{2^\gamma} JF(\mathbf{x}^{k_j}) \mathbf{v}^{k_j}. \quad (10)$$

Thus, for each  $j > j_0$  there exists  $i = i(k_j) \in \{1, \dots, m\}$  such that

$$F_i\left(\mathbf{x}^{k_j} + \frac{1}{2^\gamma} \mathbf{v}^{k_j}\right) > F_i(\mathbf{x}^{k_j}) + \frac{\beta}{2^\gamma} \nabla F_i(\mathbf{x}^{k_j})^\top \mathbf{v}^{k_j}. \quad (11)$$

Because the indices in  $\{1, \dots, m\}$  are finitely many, in the infinite sequence  $\{k_j\}$  at least one of them, say  $i_0$  should appear infinitely many times. The occurrence of  $i_0 \in \{1, 2, \dots, m\}$  defines the subsequence  $\{k_{j_l}\}$ . Then, for all  $l = 1, 2, \dots$  we have

$$F_{i_0}\left(\mathbf{x}^{k_{j_l}} + \frac{1}{2^\gamma} \mathbf{v}^{k_{j_l}}\right) > F_{i_0}(\mathbf{x}^{k_{j_l}}) + \frac{\beta}{2^\gamma} \nabla F_{i_0}(\mathbf{x}^{k_{j_l}})^\top \mathbf{v}^{k_{j_l}}. \quad (12)$$

We take the limit  $l \rightarrow \infty$  and we get

$$F_{i_0} \left( \bar{\mathbf{x}} + \frac{1}{2^\gamma} \bar{\mathbf{v}} \right) \geq F_{i_0}(\bar{\mathbf{x}}) + \frac{\beta}{2^\gamma} \nabla F_{i_0}(\bar{\mathbf{x}})^\top \bar{\mathbf{v}}. \quad (13)$$

This inequality holds for the positive integer  $\gamma$  defined by (9), and for  $i_0$ , hence  $JF(\bar{\mathbf{x}})\bar{\mathbf{v}} \not\leq 0$ , and it follows that

$$\max_{i=1, \dots, m} \nabla F_i(\bar{\mathbf{x}})^\top \bar{\mathbf{v}} \geq 0. \quad (14)$$

From (14) follows that there exists  $\bar{\mathbf{w}} \in \mathbb{R}^m$  that solves (2) for the given  $\bar{\mathbf{x}}$ . Using Theorem 2.4 we get that  $\bar{\mathbf{x}}$  is a substationary point to  $(GMOP)$ . Finally, in the convex case Remark 1 yields the assertion.  $\blacksquare$

In the following section we introduce a similar algorithm for solving multiobjective optimization problems with sign constraints.

## 5 Multiobjective optimization problems with sign constraints

Consider now the following multiobjective optimization problems with sign constraints

$$\left. \begin{array}{l} \text{Min } F(\mathbf{x}) \\ \mathbf{x} \geq \mathbf{0} \end{array} \right\} \quad (SMOP).$$

Let the vector  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \geq \mathbf{0}$  be given. Then, we can define the following two sets of indices

$$\begin{aligned} I_+(\mathbf{x}) &= \{i \in \{1, \dots, n\} : \mathbf{x}_i > 0\}, \\ I_0(\mathbf{x}) &= \{i \in \{1, \dots, n\} : \mathbf{x}_i = 0\}, \end{aligned}$$

denoted simpler by  $I_+$  and  $I_0$ , respectively, when it is clear to which vector we refer. Using them we partition the columns of the matrix  $JF(\mathbf{x})$  into two parts denoted by  $JF(\mathbf{x})_{I_0}$  and  $JF(\mathbf{x})_{I_+}$ . Consider the linear programming problem

$$\left. \begin{array}{l} \max_{(\mathbf{q}, q_0) \in \mathbb{R}^n \times \mathbb{R}} q_0 \\ [JF(\mathbf{x})]_{I_+} \mathbf{u} + [JF(\mathbf{x})]_{I_0} \mathbf{z} + q_0 \mathbf{e} \leq \mathbf{0} \\ \mathbf{z} \geq \mathbf{0} \\ 0 \leq q_0 \leq 1. \end{array} \right\} \quad (LPS(\mathbf{x}))$$

The following statement is the counterpart of Theorem 2.4 for multiobjective optimization problems with sign constraints, exhibiting the connection between  $(LPS(\mathbf{x}))$  and the substationarity of  $\mathbf{x} \in \mathbb{R}_{\oplus}^n$  to  $(SMOP)$  or the determination of a feasible joint decreasing direction of  $F$  on  $\mathbb{R}_{\oplus}^n$ .

**Theorem 5.1 (Illés and Lovics [20])** *Let  $(SMOP)$ ,  $\mathbf{x} \geq \mathbf{0}$  a feasible point of it and the associated  $(LPS(\mathbf{x}))$  be given. Let the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be continuously*

differentiable (and convex) on  $\mathcal{F} = \mathbb{R}_{\oplus}^n$ . Then,  $(LPS(\mathbf{x}))$  always has an optimal solution  $(\mathbf{u}^*, \mathbf{z}^*, q_0^*)$ . There are two possibilities for the optimal value of  $(LPS(\mathbf{x}))$ , either  $q_0^* = 0$  which means that  $\mathbf{x}$  is a stationary (weakly Pareto efficient) point to  $(SMOP)$ , or  $q_0^* = 1$  in which case  $\mathbf{q} = (\mathbf{u}^*, \mathbf{z}^*)^\top$  is a feasible joint decreasing direction of the vector function  $F$  on  $\mathbb{R}_{\oplus}^n$ .

The new method for solving  $(SMOP)$  is given in Algorithm 5.1.

---

**Algorithm 5.1** Algorithm for solving sign constrained multiobjective optimization problems

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} = \mathbb{R}_{\oplus}^n$ ,  $0 < \beta < 1$ ,  $k = 0$ ,  $J = \{\frac{1}{2^n} : n \in \mathbb{N}\}$

[Step 2] determine the (feasible) joint decreasing direction  $\mathbf{q} = \begin{pmatrix} \mathbf{u}^* \\ \mathbf{z}^* \end{pmatrix} = (q_1, \dots, q_n)^\top \in \mathbb{R}^n$  from

$$\begin{aligned} & \max_{(\mathbf{q}, q_0) \in \mathbb{R}^n \times \mathbb{R}} q_0 \\ & [JF(\mathbf{x}^k)]_{I_+} \mathbf{u} + [JF(\mathbf{x}^k)]_{I_0} \mathbf{z} + q_0 \mathbf{e} \leq \mathbf{0} \\ & 0 \leq q_0 \leq 1, \quad \mathbf{z} \geq \mathbf{0}, \quad \mathbf{q} = (\mathbf{u}, \mathbf{z})^\top \end{aligned}$$

[Step 3] if  $q_0 = 0$  then STOP ( $\mathbf{x}^k$  is a stationary solution to  $(SMOP)$ );

otherwise compute  $\sigma = \min\{\frac{x_i^k}{-q_i} : q_i < 0 \text{ and } i \in I_+\}$ ;

if  $\sigma \geq 1$  then  $\mathbf{v}^k := \mathbf{q}$ ;

otherwise  $\mathbf{v}^k := \frac{1}{\sigma} \mathbf{q}$

[Step 4] choose  $t_k$  as the largest  $t \in J$  s.t.

$$F(\mathbf{x}^k + t \mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t JF(\mathbf{x}^k) \mathbf{v}^k$$

[Step 5] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{v}^k$  and  $k = k + 1$ ; go to Step 2

---

**Lemma 5.2** Let  $(SMOP)$  be given with  $\mathcal{F} = \mathbb{R}_{\oplus}^n$  and the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be continuously differentiable on  $\mathcal{F}$ . Assume that Algorithm 5.1 starting at  $\mathbf{x}^0 \in \mathcal{F}$  produces the sequence of points  $\{\mathbf{x}^k\}$ . Then

$$F(\mathbf{x}^{k+1}) < F(\mathbf{x}^k), \quad \forall k \geq 0. \quad (15)$$

**Proof.** Since  $\mathbf{v}^k$  is a feasible joint decreasing direction computed in Step 2 of Algorithm 5.1, we have

$$[JF(\mathbf{x}^k)]_{I_+} \mathbf{u}^* + [JF(\mathbf{x}^k)]_{I_0} \mathbf{z}^* = JF(\mathbf{x}^k) \mathbf{v}^k \leq -\mathbf{e}, \quad \text{and} \quad \mathbf{z}^* \geq \mathbf{0}.$$

Therefore, in Step 4 of Algorithm 5.1 choose  $t_k$  as the largest  $t \in J = \{\frac{1}{2^n} : n \in \mathbb{N}\}$  such that

$$F(\mathbf{x}^k + t \mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t JF(\mathbf{x}^k) \mathbf{v}^k \leq F(\mathbf{x}^k) - \beta t \mathbf{e},$$

and  $\mathbf{x}^k + t \mathbf{v}^k \geq \mathbf{0}$ , where  $t > 0$ . Selecting  $t_k$  and restructuring the previous inequality, we get

$$F(\mathbf{x}^{k+1}) - F(\mathbf{x}^k) \leq -\beta t_k \mathbf{e} < \mathbf{0}, \quad (16)$$

where  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{v}^k$ . Thus we proved (15). ■

**Theorem 5.3** Let (SMOP) be given with  $\mathcal{F} = \mathbb{R}_{\oplus}^n$ , and the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  continuously differentiable (and convex) on  $\mathcal{F}$ . Assume that Algorithm 5.1 starting at  $\mathbf{x}^0 \in \mathcal{F}$  produces the sequence of points  $\{\mathbf{x}^k\}$ . Then, every accumulation point of  $\{\mathbf{x}^k\}$ , if any, is a substationary (weakly Pareto efficient) point to (SMOP).

**Proof.** Let  $\bar{\mathbf{x}}$  be an accumulation point of the sequence  $\{\mathbf{x}^k\}$ . The proof goes along the lines of the one of Theorem 4.2, one only needs to use Lemma 5.2 instead of Lemma 4.1, and then all the steps (5) – (12) hold.

Summarizing, the occurrence of  $i_0 \in \{1, 2, \dots, m\}$  defines the subsequence  $\{k_{j_l}\}$ . Then, for all  $l = 1, 2, \dots$  we have

$$F_{i_0} \left( \mathbf{x}^{k_{j_l}} + \frac{1}{2^\gamma} \mathbf{v}^{k_{j_l}} \right) > F_{i_0} \left( \mathbf{x}^{k_{j_l}} \right) + \frac{\beta}{2^\gamma} \nabla F_{i_0} \left( \mathbf{x}^{k_{j_l}} \right)^\top \mathbf{v}^{k_{j_l}},$$

with the additional information that

$$\nabla F_{i_0} \left( \mathbf{x}^{k_{j_l}} \right)^\top \mathbf{v}^{k_{j_l}} = \left[ \nabla F_{i_0} \left( \mathbf{x}^{k_{j_l}} \right) \right]_{I_+}^\top \mathbf{u}^{k_{j_l}} + \left[ \nabla F_{i_0} \left( \mathbf{x}^{k_{j_l}} \right) \right]_{I_0}^\top \mathbf{z}^{k_{j_l}},$$

where

$$I = I_+(\mathbf{x}^{k_{j_l}}) \cup I_0(\mathbf{x}^{k_{j_l}}), \quad \mathbf{v}^{k_{j_l}}|_{I_+} = \mathbf{u}^{k_{j_l}} \quad \text{and} \quad \mathbf{v}^{k_{j_l}}|_{I_0} = \mathbf{z}^{k_{j_l}}, \quad \mathbf{z}^{k_{j_l}} \geq \mathbf{0}.$$

In the infinite sequence of points  $\{\mathbf{x}^{k_{j_l}}\}$  generated by Algorithm 5.1, there should be a partition  $(\hat{I}_+, \hat{I}_0)$  of  $I$  that occurs infinitely many times. This is due to the fact that there are only finitely many possible different partitions of the index set  $I$ . Let us define the infinite index sequence  $\iota(l)$  as a subsequence of  $k_{j_l}$  where the partition  $(\hat{I}_+, \hat{I}_0)$  occurs. For  $\mathbf{x}^{\iota(l)+1} = \mathbf{x}^{\iota(l)} + \frac{1}{2^\gamma} \mathbf{v}^{\iota(l)}$ , we have

$$F_{i_0} \left( \mathbf{x}^{\iota(l)+1} \right) > F_{i_0} \left( \mathbf{x}^{\iota(l)} \right) + \frac{\beta}{2^\gamma} \left( \left[ \nabla F_{i_0} \left( \mathbf{x}^{\iota(l)} \right) \right]_{\hat{I}_+}^\top \mathbf{u}^{\iota(l)} + \left[ \nabla F_{i_0} \left( \mathbf{x}^{\iota(l)} \right) \right]_{\hat{I}_0}^\top \mathbf{z}^{\iota(l)} \right),$$

and  $\mathbf{z}^{\iota(l)} \geq \mathbf{0}$  for all indices  $\iota(l)$ . We take the limit  $l \rightarrow \infty$  obtaining

$$\bar{\mathbf{x}} = \lim_{l \rightarrow \infty} \mathbf{x}^{\iota(l)}, \quad \bar{\mathbf{v}} = \lim_{l \rightarrow \infty} \mathbf{v}^{\iota(l)}, \quad \bar{\mathbf{u}} = \lim_{l \rightarrow \infty} \mathbf{u}^{\iota(l)}, \quad \bar{\mathbf{z}} = \lim_{l \rightarrow \infty} \mathbf{z}^{\iota(l)}, \quad \text{with}$$

$$F_{i_0} \left( \bar{\mathbf{x}} + \frac{1}{2^\gamma} \bar{\mathbf{v}} \right) \geq F_{i_0} (\bar{\mathbf{x}}) + \frac{\beta}{2^\gamma} \left( \left[ \nabla F_{i_0} (\bar{\mathbf{x}}) \right]_{\hat{I}_+}^\top \bar{\mathbf{u}} + \left[ \nabla F_{i_0} (\bar{\mathbf{x}}) \right]_{\hat{I}_0}^\top \bar{\mathbf{z}} \right), \quad (17)$$

and  $\bar{\mathbf{z}} \geq \mathbf{0}$ . Inequality (17) holds for the positive integer  $\gamma$  defined by (9), and for  $i_0$ , hence

$$JF(\bar{\mathbf{x}})\bar{\mathbf{v}} = [JF(\bar{\mathbf{x}})]_{\hat{I}_+} \bar{\mathbf{u}} + [JF(\bar{\mathbf{x}})]_{\hat{I}_0} \bar{\mathbf{z}} \not\leq \mathbf{0}, \quad \text{and} \quad \bar{\mathbf{z}} \geq \mathbf{0}.$$

Therefore, the system of linear inequalities in  $(\mathbf{u}, \mathbf{z}) \in \mathbb{R}^n$

$$[JF(\bar{\mathbf{x}})]_{\hat{I}_+} \mathbf{u} + [JF(\bar{\mathbf{x}})]_{\hat{I}_0} \mathbf{z} < \mathbf{0}, \quad \text{and} \quad \mathbf{z} \geq \mathbf{0} \quad (18)$$

has no solution for the given feasible solution  $\bar{\mathbf{x}} \geq \mathbf{0}$ . Applying Theorem 5.1, we get that  $\bar{\mathbf{x}}$  is a substationary (weakly Pareto efficient) point to (SMOP).  $\blacksquare$

## 6 Linearly constrained multiobjective optimization problem

In this section we consider linearly constrained multiobjective optimization problems of the form

$$\left. \begin{array}{l} \text{Min } F(\mathbf{x}) \\ A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{array} \right\} \quad (LMOP),$$

where  $A \in \mathbb{R}^{m \times n}$  is a matrix of rank  $m$  ( $m \leq n$ ) and  $\mathbf{b} \in \mathbb{R}^m$ . Denote the *feasible set* of  $(LMOP)$  by

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}_{\oplus}^n : A\mathbf{x} = \mathbf{b}\}$$

and assume it nonempty. Like in Section 5, given  $\mathbf{x} \geq \mathbf{0}$ , we define the following two sets of indices

$$I_+ = \{i : \mathbf{x}_i > 0\}, \quad \text{and} \quad I_0 = \{i : \mathbf{x}_i = 0\},$$

used for partitioning the columns of the matrix  $JF(\mathbf{x})$  into two parts denoted by  $JF(\mathbf{x})_{I_0}$  and  $JF(\mathbf{x})_{I_+}$ , respectively. Using them, we define a linear programming problem that will be employed for deriving feasible joint decreasing directions to a generalized linearly constrained multiobjective optimization problem at a point  $\mathbf{x} \in \mathcal{P}$ , namely

$$\left. \begin{array}{l} \max_{(\mathbf{q}, q_0) \in \mathbb{R}^n \times \mathbb{R}} q_0 \\ [JF(\mathbf{x})]_{I_+} \mathbf{u} + [JF(\mathbf{x})]_{I_0} \mathbf{z} + q_0 \mathbf{e} \leq \mathbf{0} \\ A_{I_+} \mathbf{u} + A_{I_0} \mathbf{z} = \mathbf{0} \\ \mathbf{z} \geq \mathbf{0} \\ 0 \leq q_0 \leq 1. \end{array} \right\} \quad (LPL(\mathbf{x}))$$

When  $\mathbf{x} \in \mathcal{P}$ , the first constraint of  $(LPL(\mathbf{x}))$  guarantees that a solution  $(\mathbf{u}, \mathbf{z}) \in \mathbb{R}^n$  defines in case  $q_0 = 1$  a joint decreasing direction at  $\mathbf{x}$  for the objective vector function  $F$ . The second and the third constraints of  $(LPL(\mathbf{x}))$  guarantee the feasibility of the joint decreasing direction at  $\mathbf{x}$  for problem  $(LMOP)$ . The following statement is the counterpart of Theorem 2.4 and Theorem 5.1 for multiobjective optimization problems with linear constraints, and, since it contains several modifications with respect to them, we provide its proof as well. The first occurrence of a similar result has been published by Illés and Lovics [20], however, the proof given below slightly differs from the one available there.

**Theorem 6.1** *Let  $(LMOP)$  with a feasible point  $\mathbf{x} \in \mathcal{P}$  and the associated  $(LPL(\mathbf{x}))$  be given. Let the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be continuously differentiable (and convex) on  $\mathcal{F} = \mathcal{P}$ . Assume that  $\mathcal{P}$  is bounded. Then,  $(LPL(\mathbf{x}))$  always has an optimal solution  $(\mathbf{u}^*, \mathbf{z}^*, q_0^*) \in \mathbb{R}^n \times \mathbb{R}$ . There are two possibilities for the optimal value of  $(LPL(\mathbf{x}))$ , either  $q_0^* = 0$  which means that  $\mathbf{x}$  is a substationary (weakly Pareto efficient) point to  $(LMOP)$ , or  $q_0^* = 1$  which means that  $\mathbf{q} = (\mathbf{u}^*, \mathbf{z}^*)^\top$  is a feasible joint decreasing direction of the vector function  $F$  on  $\mathcal{P}$ .*

**Proof.** The point  $(\mathbf{u}, \mathbf{z}, q_0) = (\mathbf{0}, \mathbf{0}, 0)$  is a feasible solution to  $(LPL(\mathbf{x}))$ . Therefore,  $(LPL(\mathbf{x}))$  has optimal solutions since it has feasible solutions and its objective function is bounded from above by 1.

Let us examine the case when the optimal objective function value of  $(LPL(\mathbf{x}))$  is positive, namely  $q_0 > 0$ , thus the linear system of inequalities

$$\left. \begin{aligned} [JF(\mathbf{x})]_{I_+} \mathbf{u} + [JF(\mathbf{x})]_{I_0} \mathbf{z} + q_0 \mathbf{e} &\leq \mathbf{0} \\ A_{I_+} \mathbf{u} + A_{I_0} \mathbf{z} &= \mathbf{0} \\ \mathbf{z} &\geq \mathbf{0}, \quad q_0 > 0 \end{aligned} \right\} \quad (\mathcal{E}'_1)$$

has solutions, that is equivalent to

$$\left. \begin{aligned} [JF(\mathbf{x})]_{I_+} \mathbf{u} + [JF(\mathbf{x})]_{I_0} \mathbf{z} &\leq -\mathbf{e} \\ A_{I_+} \mathbf{u} + A_{I_0} \mathbf{z} &= \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0} \end{aligned} \right\} \quad (\mathcal{E}_1).$$

Any solution of  $(\mathcal{E}_1)$  is a feasible joint decreasing direction to  $(LMOP)$ .

In case when the optimal objective function value of  $(LPL(\mathbf{x}))$  is zero, namely  $q_0 = 0$ , then  $(\mathcal{E}'_1)$  has no solution, likewise  $(\mathcal{E}_1)$  has no solution, either. According to the *Farkas Lemma*, the linear system of inequalities

$$\left. \begin{aligned} \mathbf{w}^\top [JF(\mathbf{x})]_{I_+} - \mathbf{v}^\top A_{I_+} &= \mathbf{0} \\ \mathbf{w}^\top [JF(\mathbf{x})]_{I_0} - \mathbf{v}^\top A_{I_0} &\geq \mathbf{0} \\ \mathbf{e}^\top \mathbf{w} = 1, \quad \mathbf{w} &\geq \mathbf{0} \end{aligned} \right\} \quad (\mathcal{E}_2)$$

has solutions. Let us analyze the weighted scalar optimization problem associated to  $(LMOP)$ , given as follows

$$\left. \begin{aligned} \min \mathbf{w}^\top F(\mathbf{x}) \\ A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{aligned} \right\} \quad (LMOP_{\mathbf{w}}),$$

with the weight  $\mathbf{w} \in \mathcal{S}_m$  that is part of a solution of  $(\mathcal{E}_2)$ . As  $\mathcal{P}$  is nonempty,  $(LMOP_{\mathbf{w}})$  has optimal solutions, that can be characterized by the KKT-conditions corresponding to  $(LMOP_{\mathbf{w}})$ , which are described in  $(\mathcal{E}_2)$ . Thus, any solution of  $(\mathcal{E}_2)$  is a certificate that  $\mathbf{x}$  is a substationary point to  $(LMOP)$ .  $\blacksquare$

The new iterative method we propose for solving  $(LMOP)$  is given in Algorithm 6.1.

**Remark 3** *The sequence  $\{\mathbf{x}^k\}$  generated in Algorithm 6.1 is feasible to  $(LMOP)$  because  $\mathbf{x}^0 \in \mathcal{P}$  and, given  $k \in \mathbb{N}$ , when  $\mathbf{x}^k \in \mathcal{P}$  one has, by construction,  $\mathbf{x}^{k+1} \in \mathbb{R}_{\oplus}^n$  and  $A\mathbf{x}^{k+1} = A\mathbf{x}^k + t_k A\mathbf{v}^k = \mathbf{b} + \mathbf{0} = \mathbf{b}$*

In the following statement we prove that the objective vector function values sequence  $\{F(\mathbf{x}^k)\}$  is  $\mathbb{R}_{\oplus}^m$ -decreasing.

---

**Algorithm 6.1** Algorithm for solving linearly constrained multiobjective optimization problems

---

[Step 1] take  $\mathbf{x}^0 \in \mathcal{F} = \mathcal{P}$ ,  $0 < \beta < 1$ ,  $k = 0$ ,  $J = \left\{ \frac{1}{2^n} : n \in \mathbb{N} \right\}$

[Step 2] determine the (feasible) joint decreasing direction  $\mathbf{q} = \begin{pmatrix} \mathbf{u}^* \\ \mathbf{z}^* \end{pmatrix} \in \mathbb{R}^n$  from

$$\begin{aligned} & \max_{(\mathbf{q}, q_0) \in \mathbb{R}^n \times \mathbb{R}} q_0 \\ & [JF(\mathbf{x}^k)]_{I_+} \mathbf{u} + [JF(\mathbf{x}^k)]_{I_0} \mathbf{z} + q_0 \mathbf{e} \leq \mathbf{0} \\ & A_{I_+} \mathbf{u} + A_{I_0} \mathbf{z} = \mathbf{0} \\ & 0 \leq q_0 \leq 1, \mathbf{z} \geq \mathbf{0} \end{aligned}$$

[Step 3] if  $q_0 = 0$  then STOP ( $\mathbf{x}^k$  is a substationary solution to (LMOP));

otherwise compute  $\sigma = \min \left\{ \frac{x_i^k}{-q_i} : q_i < 0 \text{ and } i \in I_+ \right\}$ ;

if  $\sigma \geq 1$  then  $\mathbf{v}^k := \mathbf{q}$ ;

otherwise  $\mathbf{v}^k := \frac{1}{\sigma} \mathbf{q}$ ;

[Step 4] choose  $t_k$  as the largest  $t \in J$  s.t.

$$F(\mathbf{x}^k + t \mathbf{v}^k) \leq F(\mathbf{x}^k) + \beta t JF(\mathbf{x}^k) \mathbf{v}^k$$

[Step 5] let  $\mathbf{x}^{k+1} = \mathbf{x}^k + t_k \mathbf{v}^k$  and  $k = k + 1$ ; go to Step 2

---

**Lemma 6.2** Let (LMOP) be given with  $\mathcal{F} = \mathcal{P} \neq \emptyset$  and  $\mathcal{P}$  is assumed to be bounded. Furthermore, let the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be continuously differentiable on  $\mathcal{F}$ . Assuming that Algorithm 6.1 starting at  $\mathbf{x}^0 \in \mathcal{F}$  produces the sequence of points  $\{\mathbf{x}^k\}$ , then

$$F(\mathbf{x}^{k+1}) < F(\mathbf{x}^k), \quad \forall k \geq 0. \quad (19)$$

**Proof.** Since  $\mathbf{v}^k$  is a feasible joint decreasing direction computed in Step 2 of Algorithm 6.1 using the linear programming problem ( $LPL(\mathbf{x}^k)$ ), the proof goes along the same lines as the one of Lemma 5.2. ■

**Theorem 6.3** Let (LMOP) be given with  $\mathcal{F} = \mathcal{P}$  such that  $\mathcal{P} \neq \emptyset$  and it is bounded. Furthermore, the vector function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is continuously differentiable (and convex) on  $\mathcal{F}$ . Assume that Algorithm 6.1 starting at  $\mathbf{x}^0 \in \mathcal{F}$  produces the sequence of points  $\{\mathbf{x}^k\}$ . Then, every accumulation point of  $\{\mathbf{x}^k\}$ , if any, is a substationary (weakly Pareto efficient) point to (LMOP).

**Proof.** Let  $\bar{\mathbf{x}}$  be an accumulation point of the sequence  $\{\mathbf{x}^k\}$ . The proof goes along the same lines as the one of Theorem 5.3, as the difference in the linear programming problems considered in Step 2 of the corresponding algorithms does not effect its flow, since the additional constraint only ensures the feasibility of the new iterates after applying the computed joint decreasing direction. Only at the very end of the proof we need to handle the question of the properties of  $\bar{\mathbf{x}}$ .

Like in the proof of Theorem 5.3, the system of linear inequalities in  $(\mathbf{u}, \mathbf{z}) \in \mathbb{R}^n$

$$[JF(\bar{\mathbf{x}})]_{I_+} \mathbf{u} + [JF(\bar{\mathbf{x}})]_{I_0} \mathbf{z} < \mathbf{0}, \quad \text{and} \quad \mathbf{z} \geq \mathbf{0} \quad (20)$$

has no solution for the given feasible solution  $\bar{\mathbf{x}} \in \mathcal{P}$ . Therefore, the linear system of inequalities  $(\mathcal{E}_1)$  has no solution, either. Exactly the same steps as in the proof of Theorem 6.1 lead to the same conclusion, that  $\bar{\mathbf{x}} \in \mathcal{P}$  is a substationary (weakly Pareto efficient) point to  $(LMOP)$ . ■

**Remark 4** *The fact that  $\mathcal{P}$  is assumed to be bounded was not used directly in the proof of Theorem 6.1 and in the one of Theorem 6.3. In both cases, we imposed this hypothesis in order to guarantee that  $(LMOP_{\mathbf{w}})$  has optimal solutions. This can be ensured with different assumptions as well, that would be discussed in a subsequent work. In particular, when  $F$  is convex on  $\mathcal{P}$ ,  $(LMOP_{\mathbf{w}})$  is a convex optimization problem and thus each solution of  $(\mathcal{E}_2)$  (their existence is guaranteed by the Farkas Lemma whenever Algorithm 6.1 stops) delivers an optimal solution to  $(LMOP_{\mathbf{w}})$ , making thus the assumption of boundedness on  $\mathcal{P}$  not necessary.*

## 7 Numerical results

We implemented Algorithm 4.1 and Algorithm 6.1 in MATLAB in order to showcase the practical usage of the iterative methods we propose in this work. From the test examples available in the literature (see, for instance, [1]) we chose a convex unconstrained multiobjective optimization problem to show how our algorithm performs. After that, we modified the selected test problem by adding linear constraints in such a way, that some of the weakly Pareto efficient points became infeasible for the constrained problem. Our numerical experiments using different starting points show the effect of having linear constraints, as well, see Figures 4, 5 and 6.

**Example 1** *Let  $F_1, F_2, F_3 : \mathbb{R}^2 \rightarrow \mathbb{R}$  be defined as follows*

$$F_1(\mathbf{x}) = \frac{1}{4} \left( (x_1 - 1)^4 + 2(x_2 - 2)^4 \right), \quad F_2(\mathbf{x}) = e^{\frac{x_1 + x_2}{2}} + x_1^2 + x_2^2, \quad F_3(\mathbf{x}) = \frac{1}{6} \left( e^{-x_1} + 2e^{-x_2} \right)$$

*and let  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ ,  $F(\mathbf{x}) = (F_1(\mathbf{x}), F_2(\mathbf{x}), F_3(\mathbf{x}))^\top$ . We want to solve the following convex unconstrained multiobjective optimization problem*

$$\left. \begin{array}{l} \text{Min } F(\mathbf{x}) \\ \mathbf{x} \in \mathbb{R}^2. \end{array} \right\}$$

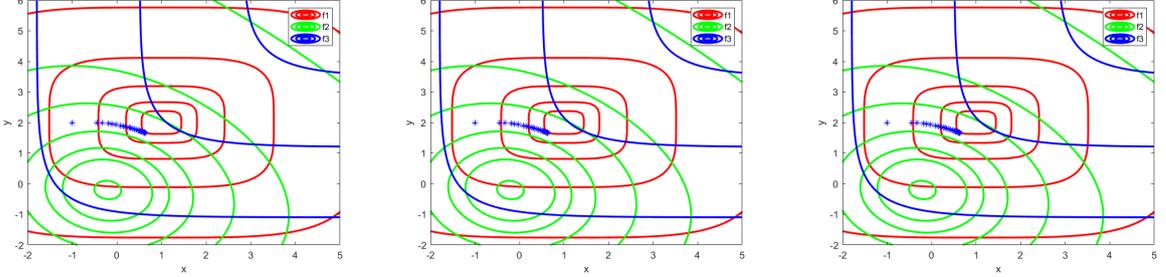
*We used two types of stopping criteria in our implementation, namely we ran the algorithm until*

*i.  $\|t_k \mathbf{q}\| \leq \epsilon$  or  $q_0 \leq \epsilon$ , or*

*ii.  $t_k \leq \epsilon$  or  $q_0 \leq \epsilon$ ,*

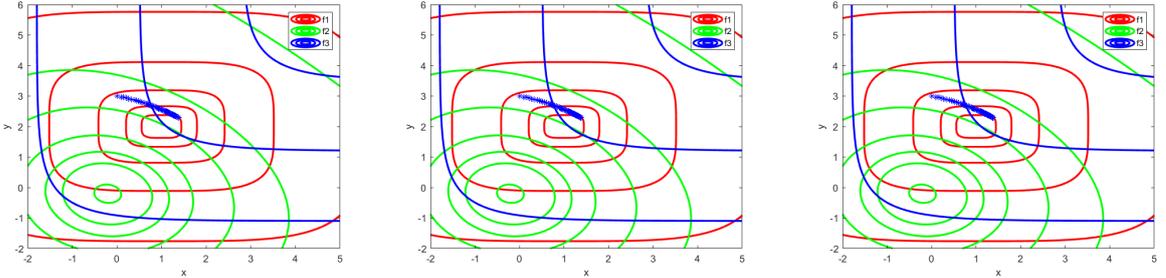
*where  $t_k$  is the stepsize and  $(\mathbf{q}, q_0)$  is the derived optimal solution to the linear programming problem from Step 2 of Algorithm 4.1. Furthermore, we also stopped the algorithm when the number of iterations reached 500. In Figures 1 and 2 we can see the obtained results with different values of  $\epsilon > 0$  by considering the starting point  $(-1, 2)^\top$  and  $(0, 3)^\top$ , respectively. In the first two cases we used the stopping criterion *i.*, while in the third*

one we applied the stopping criterion *ii*. The obtained weakly Pareto efficient points are the same within the limits of numerical errors. From the obtained numbers of iterations we can see that the algorithm is sensitive to the employed stopping criterion and to the admissible error  $\epsilon > 0$ . Note that in all cases we used  $\beta = 0.1$ .



$$\epsilon = 10^{-5}, \text{ stop } i, \text{ iter} = 26 \quad \epsilon = 10^{-10}, \text{ stop } i, \text{ iter} = 43 \quad \epsilon = 10^{-16}, \text{ stop } ii, \text{ iter} = 59$$

Figure 1: Results with starting point  $(-1, 2)^\top$ . The level sets of the used objective functions  $F_1$ ,  $F_2$  and  $F_3$  are plotted in red, green and blue, respectively. Here, stop *i* or *ii* represents the stopping criterion *i* or *ii*, while iter is the number of performed iterations before the activation of the stopping criterion.



$$\epsilon = 10^{-5}, \text{ stop } i, \text{ iter} = 54 \quad \epsilon = 10^{-10}, \text{ stop } i, \text{ iter} = 63 \quad \epsilon = 10^{-16}, \text{ stop } ii, \text{ iter} = 66$$

Figure 2: Results with starting point  $(0, 3)^\top$ . The level sets of the used objective functions  $F_1$ ,  $F_2$  and  $F_3$  are plotted in red, green and blue, respectively. Here, stop *i* or *ii* represents the stopping criterion *i* or *ii*, while iter is the number of performed iterations before the activation of the stopping criterion.

In order to get an idea about the whole set of weakly Pareto efficient points of the considered multiobjective optimization problem, we provide a so-called inner approximation of it, that comprises the outcomes of starting the proposed algorithm from as many as possible points (not to be confused with the outer approximation of a solution set of a multiobjective optimization problem, for which other methods are available, see Illés and Lovics [20]). To proceed, we divided the rectangle  $\mathcal{T} = [-2, 8] \times [-2, 8]$  into  $10 \times 10$  unit squares and we generated 5 points in each square. In this way, we obtained 500 starting points in  $\mathcal{T}$ . We ran Algorithm 4.1 with these starting points with  $\epsilon = 10^{-5}$  and by using the stopping criterion *i*. In Figure 3 we can see the obtained weakly Pareto efficient points after running the algorithm 500 times with these starting points. Hence, Figure 3 shows how the set of weakly Pareto efficient points looks like in case of Example 1. Note that the running time in this case was about 14 hours.

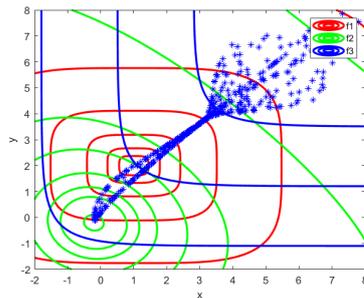


Figure 3: Set of weakly Pareto efficient points

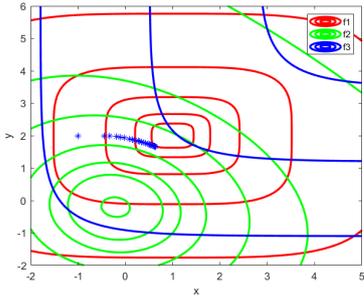
We also tested our new method on a linearly constrained multiobjective optimization problem.

**Example 2** Consider the following problem

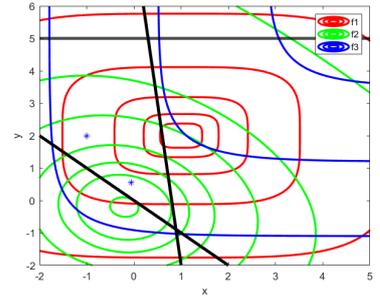
$$\left. \begin{array}{l} \text{Min } F(\mathbf{x}) \\ A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq \hat{\mathbf{x}}, \end{array} \right\}$$

where  $F$  is the function given in Example 1,  $A = \begin{pmatrix} 10 & 1 \\ 0 & 1 \\ -1 & -1 \end{pmatrix}$ ,  $\mathbf{b} = \begin{pmatrix} 8 \\ 5 \\ 0 \end{pmatrix}$  and  $\hat{\mathbf{x}} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$ .

Note that this problem is slightly different to (LMOP) in the sense that instead of  $A\mathbf{x} = \mathbf{b}$  we have  $A\mathbf{x} \leq \mathbf{b}$  and also a different lower bound imposed on the vector  $\mathbf{x}$ . To accommodate this situation, in our numerical computations we modified Step 2 of Algorithm 5.1. The obtained results can be seen in Figures 4 and 5, where we compared the results with the unconstrained case. In all these cases we used  $\beta = 0.1$ ,  $\epsilon = 10^{-5}$  and stopping criterion  $i$ . In the captions of the figures we can see the obtained weakly Pareto efficient point within the limits of the numerical calculation errors and the necessary numbers of iterations. From Figures 4 and 5 we can see the effect of the linear constraints on the solution. In Figure 4 when we have linear constraints the algorithm stops with  $q_0$  becoming less than  $\epsilon$ . We can see that the obtained weakly Pareto efficient point and the trajectory also differs from the unconstrained case, because the obtained point  $\bar{\mathbf{x}}_u$  is not feasible to the constrained problem. In Figure 5 when we have linear constraints the obtained weakly Pareto efficient point is on the boundary of the feasible polyhedron, and the followed trajectory is also different than in the unconstrained case. In both cases the numbers of iterations are also different, because the obtained point  $\bar{\mathbf{x}}_u$  is not feasible to the constrained problem. Figure 3 helps us to understand that from any feasible point we get a weakly Pareto efficient point that may differ from the unconstrained case solution, but even in this case it is a weakly Pareto efficient point corresponding to the feasible solution set of the problem.

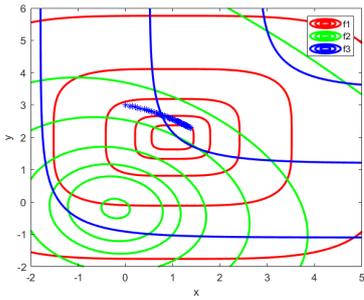


$$\bar{\mathbf{x}}_{\mathbf{u}} = (0.6407, 1.6583)^\top, \text{ iter} = 26$$

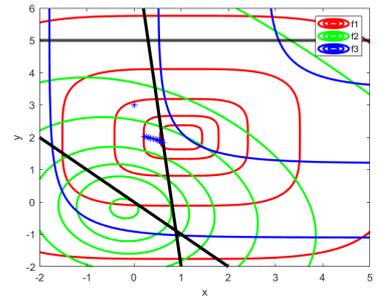


$$\bar{\mathbf{x}}_{\mathbf{c}} = (-0.0556, 0.5556)^\top, \text{ iter} = 2$$

Figure 4: Results with starting point  $(-1, 2)^\top$  without constraints and with linear constraints. The lines coming from the linear constraints are plotted in black. Here,  $\bar{\mathbf{x}}_{\mathbf{u}}$  and  $\bar{\mathbf{x}}_{\mathbf{c}}$  represent the obtained weakly Pareto efficient points within the numerical tolerance, and iter is the number of performed iterations before the activation of the stopping criterion.



$$\bar{\mathbf{x}}_{\mathbf{u}} = (1.3861, 2.2824)^\top, \text{ iter} = 54$$



$$\bar{\mathbf{x}}_{\mathbf{c}} = (0.6150, 1.8498)^\top, \text{ iter} = 11$$

Figure 5: Results with starting point  $(0, 3)^\top$  without constraints and with linear constraints. The lines coming from the linear constraints are plotted in black. Here,  $\bar{\mathbf{x}}_{\mathbf{u}}$  and  $\bar{\mathbf{x}}_{\mathbf{c}}$  represent the obtained weakly Pareto efficient points within the numerical tolerance, and iter is the number of performed iterations before the activation of the stopping criterion.

Let us define the rectangle  $\mathcal{T} = [-2, 1] \times [-1, 5]$ , such that it contains the feasible solution set of the problem. We divided  $\mathcal{T}$  into  $10 \times 10$  unit squares and we intersected the squares with the polyhedron obtained from the linear constraints coming from Example 2. We generated 10 points in each obtained form. We ran Algorithm 4.1 with these starting points with  $\beta = 0.1$ ,  $\epsilon = 10^{-5}$  and by using the stopping criterion  $i$ . In Figure 6 we can see the obtained weakly Pareto efficient points in each case. We can see that in many cases the algorithm stopped at the boundary of the polyhedron, which are weakly Pareto efficient points for the constrained case. Based on our observation, the running time in this case was 399 seconds, because we have less starting points and the linear constraints, as well.

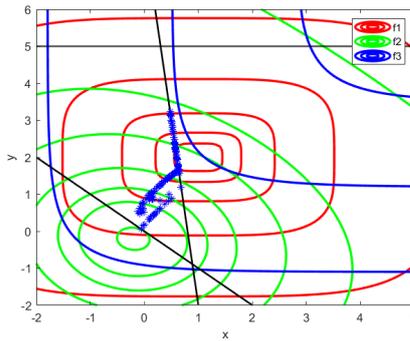


Figure 6: Set of weakly Pareto efficient points

## 8 Conclusions and further research plans

We proposed a generic scheme for solving multiobjective optimization problems. In particular, we derived new algorithms for solving unconstrained, sign constrained and linearly constrained multiobjective optimization problems, where we computed the feasible joint decreasing direction using suitably defined linear programming problems. In all the cases we showed that the objective vector function values sequence is decreasing with respect to the considered orthant. We proved that every accumulation point of the sequence generated by the algorithms, if any, is a substationary point to the considered multiobjective optimization problem. When the objective vector function of the considered multiobjective optimization problem is convex with respect to the corresponding non-negative orthant, each accumulation point of the mentioned sequence turns out to be weakly Pareto efficient. Different to their existing counterparts from the literature, a novelty of these algorithms is that the decreasing directions are easily computable via linear programming. Moreover, in the proposed algorithms both the computation of the joint decreasing direction using linear programming model and the line search can be exactly solved. The computational performance of our algorithms has been illustrated on convex unconstrained and convex linearly constrained multiobjective optimization problems. The numerical results in the constrained case show the effect of the linear constraints both on the running time and on the obtained weakly Pareto efficient set.

As further research we plan to consider other choices of stepsizes in the proposed iterative methods and to introduce algorithms where quadratic programming problems are used for determining the joint decreasing directions (following [26]). It would be also interesting to give a subdivision framework (like in Illés and Lovics [20]) for all these types of multiobjective optimization problems in order to efficiently approximate their entire weak Pareto frontiers.

## Acknowledgement

The research of Sorin-Mihai Grad was partially supported by a CIAS Senior Research Fellow Grant of the Corvinus Institute for Advanced Studies, and by a public grant from the Fondation Mathématique Jacques Hadamard as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH. Petra Renáta Rigó is thankful

for the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the PD\_22 funding scheme, project no. 142154. The research of Tibor Illés and Petra Renáta Rigó was partially supported by the Austrian-Hungarian Action Foundation, project number 116öu8.

## Declarations

All three authors have contributed equally to the work.

Declarations of interest: none.

## References

- [1] Md. A.T. Ansary, G. Panda. A modified Quasi-Newton method for vector optimization problem. *Optimization*, 64(11), 2289–2306, 2015.
- [2] J.Y. Bello Cruz, L.R. Lucambio Pérez, J.G. Melo. Convergence of the projected gradient method for quasiconvex multiobjective optimization. *Nonlinear Anal, Theory Methods Appl, Ser A, Theory Methods*, 74(16):5268–5273, 2011.
- [3] R.I. Boţ, S.-M. Grad. Inertial forward-backward methods for solving vector optimization problems. *Optimization*, 67(7):959–974, 2018.
- [4] R.I. Boţ, S.-M. Grad, G. Wanka. *Duality in Vector Optimization*. Springer-Verlag, Berlin-Heidelberg, 2009.
- [5] H. Bonnel, A.N. Iusem, B.F. Svaiter. Proximal methods in vector optimization. *SIAM J Optim*, 15(4):953–970, 2005.
- [6] E. Carrizosa, J.B.G. Frenk. Dominating sets for convex functions with some applications. *J Optim Theory Appl*, 96(2):281–295, 1998.
- [7] J. Chen, L. Tang, X. Yang. A Barzilai-Borwein descent method for multiobjective optimization problems. *Eur. J. Oper. Res.*, 311(1):196–209, 2023.
- [8] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer-Verlag, Berlin-Heidelberg, 2008.
- [9] H. Eschenauer, J. Koski, A. Osyczka. *Multicriteria Design Optimization: Procedures and Applications*. Springer-Verlag, Berlin, 1990.
- [10] G.W. Evans. An overview of techniques for solving multiobjective mathematical programs. *Manage Sci*, 30(11):1268–1282, 1984.
- [11] J. Fliege, B.F. Svaiter. Steepest descent methods for multicriteria optimization. *Math Methods Oper Res*, 51(3):479–494, 2000.

- [12] J. Fliege, L.M. Graña-Drummond, B.F. Svaiter. Newton’s method for multiobjective optimization. *SIAM J Optim*, 20(2):602–626, 2009.
- [13] E.H. Fukuda, L.M. Graña-Drummond. A survey on multiobjective descent methods. *Pesqui Oper*, 34(3):585–620, 2014.
- [14] E.H. Fukuda, L.M. Graña-Drummond. Inexact projected gradient method for vector optimization. *Comput Optim Appl*, 54(3):473–493, 2013.
- [15] E.H. Fukuda, L.M. Graña-Drummond. On the convergence of the projected gradient method for vector optimization. *Optimization*, 60(8–9):1009–1021, 2011.
- [16] S.-M. Grad. A survey on proximal point type algorithms for solving vector optimization problems. In: “Splitting Algorithms, Modern Operator Theory, and Applications”, Springer-Verlag, Cham, 269–308, 2019.
- [17] L.M. Graña-Drummond, 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, B.F. Svaiter. A quadratically convergent Newton method for vector optimization. *Optimization*, 63(5):661–677, 2014.
- [19] L.M. Graña-Drummond, B.F. Svaiter. A steepest descent method for vector optimization. *J Comput Appl Math*, 175(2):395–414, 2005.
- [20] T. Illés, G. Lovics. Approximation of the whole Pareto optimal set for the vector optimization problem. *Acta Polytech Hung*, 15(1):127–148, 2018.
- [21] A.N. Iusem, J.G. Melo, R.G. Serra. A strongly convergent proximal point method for vector optimization. *J Optim Theory Appl*, 190(1):183–200, 2021.
- [22] H.W. Kuhn, A.W. Tucker. Nonlinear programming. In: “Proceedings of the 2<sup>nd</sup> Berkeley Symposium on Mathematical and Statistical Probability”, University of California Press, 481–492, 1951.
- [23] D.T. Luc. *Theory of Vector Optimization*. Lect Notes Econ Math Syst, 319, Springer-Verlag, Berlin, 1989.
- [24] L.R. Lucambio Pérez, L.F. Prudente. Nonlinear conjugate gradient methods for vector optimization. *SIAM J Optim*, 28(3):2690–2720, 2018.
- [25] K.M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Norwell, 1999.
- [26] S. Schäffler, R. Schultz, K. Weinzierl. Stochastic method for the solution of unconstrained vector optimization problems. *J Optim Theory Appl*, 114(1):209–222, 2002.