



A Projected Subgradient Method for Nondifferentiable Quasiconvex Multiobjective Optimization Problems

Xiaopeng Zhao¹ · Markus A. Köbis² · Yonghong Yao¹ · Jen-Chih Yao^{3,4}

Received: 19 September 2020 / Accepted: 10 May 2021 / Published online: 31 May 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In this paper, we propose a projected subgradient method for solving constrained non-differentiable quasiconvex multiobjective optimization problems. The algorithm is based on the Plastria subdifferential to overcome potential shortcomings known from algorithms based on the classical gradient. Under suitable, yet rather general assumptions, we establish the convergence of the full sequence generated by the algorithm to a Pareto efficient solution of the problem. Numerical results are presented to illustrate our findings.

Keywords Multiobjective optimization · Pareto optimality · Quasiconvex functions · Projected subgradient method · Quasi-Fejér convergence

Mathematics Subject Classification 49M37 · 65K05 · 90C26 · 90C29

Communicated by Fabián Flores-Bazán.

✉ Jen-Chih Yao
yaojc@mail.cmu.edu.tw

Xiaopeng Zhao
xiaopengzhao@tiangong.edu.cn

Markus A. Köbis
markus.kobis@ntnu.no

Yonghong Yao
yyhtgu@hotmail.com

¹ School of Mathematical Sciences, Tiangong University, Tianjin 300387, China

² Norwegian University of Science and Technology, 7030 Trondheim, Norway

³ Center for General Education, China Medical University, Taichung 40402, Taiwan

⁴ Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 80424, Taiwan

1 Introduction

Multiobjective optimization, also known as multicriteria optimization, refers to the process of optimizing several objective functions simultaneously over a feasible region. Usually, for nontrivial problems, it is nearly impossible to find a single point that individually optimizes all given objective functions at the same time. Therefore, the concept of optimality is replaced by the concept of *Pareto optimality* or *efficiency*.

Multiobjective optimization problems have been widely applied in many areas, such as engineering [12], management science [31,35], environmental analysis [13] and design [16] to name just a few. For more details on the applications of multiobjective optimization in economy, industry, agriculture and many more, one can see [24,26].

Due to its various applications, the research on multiobjective optimization problems has captured a great amount of attention and great development has been made in many aspects including the existence of solutions, the relationships between solutions under different definitions, Farkas-type results, optimality conditions and numerical optimization algorithms, see, for instance, [8,11,14,15,18,20,21,23,33,34] and the references therein.

In the present paper, we focus on optimization algorithms for solving multiobjective optimization problems with not necessarily differentiable or convex objectives over a convex feasible region. Up to now, many classical (i.e., scalar-valued) optimization methods have been successfully extended to the multiobjective optimization setting, such as the projected gradient method [6,18], steepest descent method [4], Newton method [34], proximal point method [8,27,28], trust-region method [33] and several more.

Among these strategies, this paper is concerned with the projected gradient method. As a classical and important basic scheme, the projected gradient method and its modified versions have been used frequently to solve nonsmooth optimization problems [1,25,37–40]. For the multiobjective optimization case, Graña Drummond and Iusem [18] considered two versions of projected gradient methods for constrained vector optimization problems with continuously differentiable objective function. For one version of the algorithm (with exogenously given square summable step lengths), by using an Armijo-like rule, the authors established the convergence of the generated sequence to a weakly efficient optimum when the objective function is convex. For another version of the algorithm (without exogenous step lengths and convexity assumption), they just showed the stationarity of the cluster points (if any), whose convergence to a weakly efficient optimum was established in [17] with the convexity assumption on the objective function. For nondifferentiable convex vector optimization problems, by exploring the structure of the vector problem, Bello Cruz [2] proposed a projected subgradient method and showed that all cluster points of the generated sequence are weakly efficient optimum points of the problem. Recently, Brito et al. [6] and Bento et al. [5] have proposed projected subgradient methods for solving constrained convex multiobjective optimization problems, where the multiobjective function is componentwise convex. By employing a scalarization process, the authors obtained the convergence of the generated sequence toward a Pareto optimal point. Their method serves as the starting point for the algorithm presented in this work. It

is, however, extended by means of a more general gradient concept on the one hand and additional algorithmic degrees of freedom on the other hand.

It is worth pointing out that the above-cited work centers on convex optimization models. In many real-life problems, one often encounters the situation that the involved objective function is not necessarily convex. Thus, the study of nonconvex multiobjective optimization problems is of high importance from both the theoretical and the application point of view.

In this paper, we consider constrained quasiconvex multiobjective optimization problems, meaning that each objective function is a quasiconvex function. Our motivation stems from the extensive applications of this type of models in fields like microeconomy or location theory, see [19,27] and Example 5.4.

For the projected gradient method for quasiconvex multiobjective optimization, let us mention the work in [3], where the objective function is assumed to be continuously differentiable. In [3], the authors showed the convergence of the sequence generated by the method introduced therein to a stationary point and furthermore to a weak Pareto efficient solution when the components of the multiobjective function are pseudoconvex. As to unconstrained nondifferentiable quasiconvex multiobjective optimization problems, by using the lower subdifferentials of the quasiconvex function [30,36], Cruz Neto et al. [9] proposed a subgradient-type method to obtain the Pareto optimal points. There are, however, no numerical examples to illustrate the implementation of the method therein.

Inspired by the above-mentioned work, we aim to propose a projected subgradient method for solving quasiconvex multiobjective optimization problems on nonempty closed convex subsets of the Euclidean space. In this case, each objective function is not necessarily differentiable. First order descent information regarding the objective function is hereby based on the concept of the *Plastia subdifferential*, see [30] and Definition 2.3. This subdifferentiability concept was chosen because (a) its actual values can be obtained easily from the classical gradient and the Lipschitz constant of the objective in the smooth case, (b) its definition relies on the lower level sets of the function (making it particularly interesting for (nonsmooth) minimization problems) and (c) albeit from being infinite (therefore well defined in a very general setting), the norm of some elements of the *Plastia subdifferential* can be bounded from above, which is important from the numerical viewpoint.

Under suitable hypotheses, we obtain total convergence for a Pareto optimal solution of the problem. Furthermore, the performance of our method is illustrated by numerical experiments.

The outline of this article is as follows. In Sect. 2, we recall some basic definitions and preliminary materials that are used in the sequel. We also provide some examples for the proposed subdifferential calculus to prepare the numerical problems and make the paper self-contained. In Sect. 3, we present the algorithm and prove its well-definedness. Sect. 4 contains the convergence analysis of this algorithm. Specifically, the convergence of the sequence produced by the algorithm to a Pareto optimal point of the problem is established under some reasonable assumptions. Numerical examples are presented in Sect. 5 to demonstrate the performance of this method. Finally, we provide some concluding remarks in Sect. 6.

2 Preliminaries and Auxiliary Results

In this section, we recall some notations and results that are used throughout our presentation. We denote by $\langle \cdot, \cdot \rangle$ the usual inner product of \mathbb{R}^n and by $\| \cdot \|$ its corresponding norm. If A is countable, we denote by $\#A$ the number of its elements. Let \mathbb{R}_+^m and \mathbb{R}_{++}^m denote the nonnegative orthant and positive orthant of \mathbb{R}^m , respectively. We consider the partial orders \leq ($<$) in \mathbb{R}^m , defined by

$$x \leq y \text{ (resp. } x < y) \iff y - x \in \mathbb{R}_+^m \text{ (resp. } y - x \in \mathbb{R}_{++}^m).$$

Consider the function $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$, given by $F(x) = (f_1(x), f_2(x), \dots, f_m(x))$. We are interested in the following multiobjective optimization problem

$$\min_{x \in C} F(x), \tag{1}$$

where $C \subseteq \mathbb{R}^n$ is a nonempty closed convex set.

Definition 2.1 A point $x^* \in C$ is called a weak Pareto optimal point or weak Pareto efficient solution for the problem (1), if there does not exist $x \in C$ such that $F(x) < F(x^*)$. If there does not exist $x \in C$ satisfying $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$, then x^* is said to be a Pareto optimal point or Pareto efficient solution for (1).

We will denote the set of Pareto (resp. weak Pareto) efficient solutions of the problem by E (resp. E_w). For a given $x \in \mathbb{R}^n$ and a closed convex set $D \subseteq \mathbb{R}^n$, recall that the orthogonal projection of x onto D , denoted by $P_D(x)$, is the unique point in D such that $\|x - P_D(x)\| = \inf_{y \in D} \|x - y\|$; moreover, one has

$$\langle x - P_D(x), y - P_D(x) \rangle \leq 0, \quad y \in D. \tag{2}$$

A function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be quasiconvex if for every $x, y \in \mathbb{R}^n$ and $\alpha \in [0, 1]$,

$$f(\alpha x + (1 - \alpha)y) \leq \max\{f(x), f(y)\}.$$

It is clear that a convex function is also a quasiconvex function. Following [24, Corollary 6.6], the above definition can be adapted to multiobjective functions as follows.

Definition 2.2 Let the multiobjective function $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ be given by $F = (f_1, f_2, \dots, f_m)$. Then F is quasiconvex if each component function f_i is quasiconvex.

For a given $z \in \mathbb{R}$ and a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$, denote by

$$S_f^{\leq}(z) = \{x \in \mathbb{R}^n : f(x) \leq f(z)\} \quad \text{and} \quad S_f^{<}(z) = \{x \in \mathbb{R}^n : f(x) < f(z)\}$$

the level set and the strict level set of f at z . The following notion of lower subdifferentiability and strict lower subdifferentiability were introduced by Plastria [30], which extend the notion of the Fenchel–Moreau subdifferentials

$$\partial^{F-M} f(z) = \{s \in \mathbb{R}^n : \langle s, x - z \rangle \leq f(x) - f(z) \quad \forall x \in \mathbb{R}^n\},$$

of the convex function (see [32]).

Definition 2.3 Given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and a point $z \in \mathbb{R}^n$, the lower subdifferential, respectively, the strict lower subdifferential (in the sense of Plastria [30]) of f at z are defined by

$$\partial^< f(z) = \{s \in \mathbb{R}^n : \langle s, x - z \rangle \leq f(x) - f(z) \quad \forall x \in S_f^<(z)\}$$

and

$$\partial^{\leq} f(z) = \{s \in \mathbb{R}^n : \langle s, x - z \rangle \leq f(x) - f(z) \quad \forall x \in S_f^{\leq}(z)\}.$$

The function f is said to be lower subdifferentiable (resp. strictly lower subdifferentiable) at z if $\partial^< f(z) \neq \emptyset$ (resp. $\partial^{\leq} f(z) \neq \emptyset$).

It is clear that $\partial^{\leq} f(z) \subset \partial^< f(z)$. Moreover, note that (also one can see [30, Theorem 3.1] or [9]) for any $s \in \partial^{\leq} f(z)$ and any $\lambda \geq 1$, one has $\lambda s \in \partial^{\leq} f(z)$, which shows that the set $\partial^{\leq} f(z)$ is unbounded. Thus, the lower subdifferential (or the strict lower subdifferential) is generally different from the Fenchel–Moreau subdifferential, even for convex functions. However, from the definition it is already clear that the inclusion $\partial^{F-M} f(z) \subseteq \partial^{\leq} f(z)$ holds for any convex function.

The following property of the strict lower subdifferential can be obtained easily from the definition, and one can also refer to [36, Proposition 2.2(a)].

Proposition 2.1 For a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $z \in \mathbb{R}^n$, $0 \in \partial^< f(z)$ if and only if z is a global minimizer of f .

Some other useful properties of the (strict) lower subdifferential for quasiconvex and Lipschitz continuous functions are listed in the following proposition, for (i), one can see [30, Theorem 2.3], while for (ii), one can see [36, Propositions 2.4 and 2.6]. But before stating it, let us first recall that a quasiconvex function f is said to be essentially quasiconvex if each local minimizer of f is global.

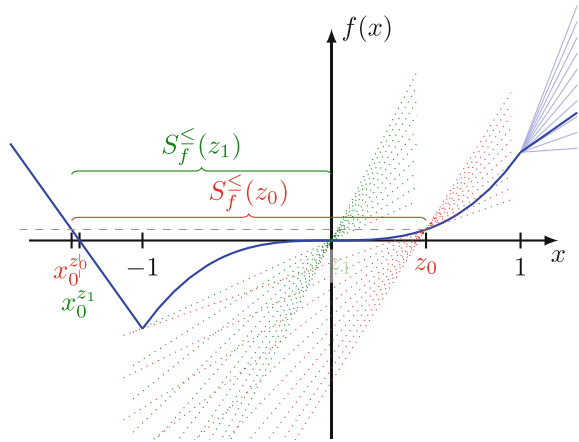
Proposition 2.2 Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a quasiconvex and Lipschitz continuous function with Lipschitz constant L . Then the following assertions hold:

- (i) For each $z \in \mathbb{R}^n$, there exists $s \in \partial^< f(z)$ such that $\|s\| \leq L$.
- (ii) Suppose further that f is essentially quasiconvex. Then $\partial^{\leq} f(z) \neq \emptyset$ for every $z \in \mathbb{R}^n$, and $\partial^< f(z) = \partial^{\leq} f(z)$ for every nonminimizer z of f .

Example 2.1 Consider the function (see [9, Example 1] for a similar example and Fig. 1) $f : \mathbb{R} \rightarrow \mathbb{R}$ given by

$$f(x) = \begin{cases} -\frac{1}{3} - (x - (-1)) & \text{for } x \leq -1 \\ \frac{x^3}{3} & \text{for } x \in (-1, 1), \\ \frac{1}{3} + c \cdot (x - 1) & \text{for } x \geq 1. \end{cases} \quad (c \geq 0)$$

Fig. 1 Sketch of the function f in Example 2.1, values of $c > 0$ range between 0.05 and 2.0; the thick solid line indicating $c = \frac{1}{2}$



The function f is (for $c \neq 1$) not differentiable at $x = \pm 1$ and quasiconvex. As the function is monotone increasing on $[-1, \infty)$, the level sets $S_f^z(z)$ for $z \in [-1, +\infty)$ are given by

$$S_f^{\leq}(z) = \{x \in \mathbb{R} | x_0^z \leq x \leq z\} \quad \text{and} \quad S_f^\leq(z) = \{x \in \mathbb{R} | x_0^z \leq x < z\},$$

where x_0^z denotes the x -coordinate of the left intersection point of $f \equiv f(z)$, which is always in $(-\infty, -1]$.

We calculate the Plastia subdifferential of f for two points:

1. At $z_0 = \frac{1}{2}$, the Plastia subdifferential is given by

$$\partial^{\leq} f(z_0) = \left[\frac{1}{4}, \infty \right).$$

2. At $z_1 = 0$ we can calculate the Plastia subdifferential as

$$\partial^{\leq} f(z_1) = \left[\frac{1}{3}, \infty \right).$$

One disadvantage of working with subdifferentials instead of the classical gradient or approximations thereof is the numerical tractability of calculating ∂f .

One can, however, show, cf. [9, Cor. 2.22], that for differentiable function with known Lipschitz constant, an element of the Plastia subdifferential can always be found by appropriately upscaling the gradient ∇f , immediately widening the potential applications of this subgradient concept.

Example 2.2 A classical example of a quasiconvex function is the negative Gaussian bell curve. In the one-dimensional case, this function is (scaled) given by

$$f(x) = -\exp(-x^2/2).$$

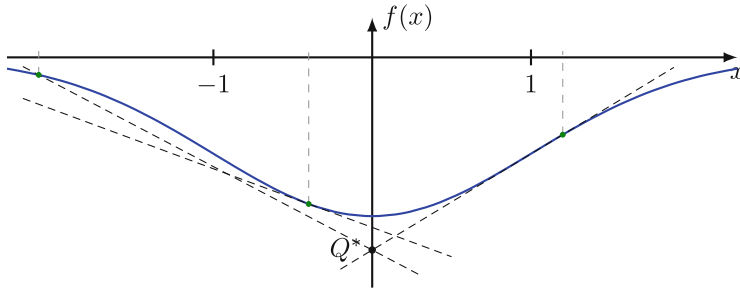


Fig. 2 Sketch of the function f in Example 2.2

Instead of giving exact formulas for $\partial^{\leq} f(x)$ or just strongly over-approximating it, we will use a ‘mix’ of the classical gradient and a heuristic to approximate $\partial^{\leq} f(x)$ for the nonconvex areas.

Between the two inclination points $x \in [-1, 1]$, the function is strictly convex, and for the calculation of $\partial^{\leq} f(x)$, one can simply rely on the classical gradient of f . If x is exactly at one of these points $x_0 = \pm 1$, the intersection point of the linear approximation $f(x_0) + \nabla f(x_0) \cdot (\bullet - x_0)$ with the y -axis is at $Q^* := (0, -2 \cdot \exp(-1/2)) \approx (0, -1.2131)$. To obtain elements of the Plastria subdifferential for $|x| > 1$, we can use the slope of the linear interpolant between Q^* and $(x, f(x))$. This leads to the formula

$$\tilde{s}(x) \subset \partial^{\leq} f(x), \quad \text{with} \quad \tilde{s}(x) := \begin{cases} x \cdot \exp(-x^2/2) & \text{for } |x| \leq 1 \\ \left(\frac{2}{\sqrt{\exp(1)}} - \exp(-x^2/2) \right) / x & \text{else.} \end{cases} \quad (3)$$

In Fig. 2, the result of this approximation is illustrated.

Finally, we conclude this section by presenting the well-known quasi-Fejér convergence theorem, which has been widely used to analyze gradient and subgradient methods, see, among others, [3,6,9,17,18].

Definition 2.4 A sequence $\{u_k\} \subseteq \mathbb{R}^n$ is said to be quasi-Fejér convergent to a nonempty set $U \subseteq \mathbb{R}^n$ if and only if for each $u \in U$, there exists a summable sequence $\{\epsilon_k\} \subseteq \mathbb{R}_+$ such that

$$\|u_{k+1} - u\|^2 \leq \|u_k - u\|^2 + \epsilon_k.$$

Proposition 2.3 [7, Theorem 1] *If $\{u_k\} \subseteq \mathbb{R}^n$ is quasi-Fejér convergent to a nonempty set $U \subseteq \mathbb{R}^n$, then $\{u_k\}$ is bounded. Furthermore, if a cluster point \bar{u} of $\{u_k\}$ belongs to U , then the whole sequence $\{u_k\}$ converges to \bar{u} .*

3 The Algorithm

In this section, we describe our proposed algorithm to solve (1). In order to guarantee its well-definedness, we assume throughout the whole paper that the following condition holds.

Assumption 3.1 Every component function $f_i: \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, 2, \dots, m$, of the objective function $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is essentially quasiconvex and Lipschitz continuous with Lipschitz constant L .

Consider the sequences $\{\gamma_k\} \subseteq (0, 1]$ and $\{\alpha_k\} \subseteq \mathbb{R}_+$ with

$$\sum_{k=0}^{+\infty} \gamma_k \alpha_k = +\infty, \quad \sum_{k=0}^{+\infty} \alpha_k^2 < +\infty. \tag{4}$$

The projected subgradient algorithm is defined as follows.

Algorithm 3.1 Input: (i) objective mapping $F = (f_1, f_2, \dots, f_m): \mathbb{R}^n \rightarrow \mathbb{R}^m$ with Lipschitz constant $L > 0$, (ii) nonempty, closed, convex set $C \subseteq \mathbb{R}^n$, (iii) parameter sequences $\{\alpha_k\} \subseteq \mathbb{R}_+, \{\gamma_k\} \subseteq (0, 1]$ fulfilling (4), (iv) for arbitrary $k = 0, 1, \dots$: nonempty index sets $\{I_k\} \subseteq \{1, 2, \dots, m\}$, (v) weighting parameters $\{(\lambda_1^k, \dots, \lambda_{\#I_k}^k)\}$ such that for all $k = 0, 1, \dots$ it holds $(\lambda_1^k, \dots, \lambda_{\#I_k}^k) \subset [0, 1]^{\#I_k}$ and $\sum_{i \in I_k} \lambda_i^k = 1$, (vi) required optimization goal ('weak' or 'strong' Pareto points).

Step 0: Find an initial iteration point $x_0 \in C$. Set $k = 0$.

Step 1: Take $s_i^k \in \partial^{\leq} f_i(x_k)$ such that $\|s_i^k\| \leq L$ for each $i = 1, 2, \dots, m$. If $\min_{i=1,2,\dots,m} \|s_i^k\| = 0$ (for optimization goal 'weak') or

$\max_{i=1,2,\dots,m} \|s_i^k\| = 0$ (for optimization goal 'strong'), then **STOP**.

Otherwise, let $\eta_k = \max_{i \in I_k} \{\|s_i^k\|, 1\}$ and compute

$$v_k := \operatorname{argmin}_{v \in C - x_k} h_{x_k}(v), \tag{5}$$

where $h_{x_k}(v) := \frac{\alpha_k}{\eta_k} \langle \sum_{i \in I_k} \lambda_i^k s_i^k, v \rangle + \frac{\|v\|^2}{2}$.

Step 2: Define

$$x_{k+1} = x_k + \gamma_k v_k. \tag{6}$$

Set $k := k + 1$ and go back to Step 1.

Output: (Weak-) Pareto optimal point $x^* = x_k$.

Remark 3.1 (Practical Considerations)

1. The main computational costs of the algorithm lie in the solution of the internal minimization problems $\min_v h_{x_k}(v)$ (and potentially finding elements of the Plastia subdifferential). These constitute quadratic optimization problems over a convex domain. Techniques from quadratic programming (interior point methods, active set methods, augmented Lagrangian methods, extended simplex schemes and others) or projection-based specially tailored solvers (cf. Proposition 4.2) are the method of choice here.

2. Finding a point x_0 in C is simply stated as the initial step here. If the constraint set can be represented or sufficiently well approximated by linear inequalities, routines from linear programming can be used in Step 0. Otherwise, projection-based algorithms can be used to solve the according feasibility problem.
3. The strict stopping criteria of the algorithm can hardly be reached in a practical implementation of the algorithm. So, instead of the ‘hard’ zeros, it is useful to introduce tolerances for these tests. Even in this case, it rarely happens as one can show that if the algorithm stops at iteration k with optimization goal ‘weak,’ then x_k is optimal for at least one component of F and in the quasi-interior of C , (in particular a weak Pareto optimal point). If it pre-terminates with optimization goal ‘strong,’ the iterate is optimal for all components of F . (So, there are no conflicting goals in F , yet x_k is Pareto optimal.)

In the examples below, the stopping criterion is based on norm of the steps in the iteration.

4. The main purpose of relating the subgradient values s_i^k to the Lipschitz constant of f_i is to bound the subdifferential vectors. Practically, the exact Lipschitz constants are often either unknown or of no practical value as the slope of f_i varies heavily over C . In such cases, it might be useful to change the condition to one involving ‘local’ Lipschitz constants instead.
5. Note that evaluations of the objective function F are not explicitly part of the algorithm. It is much more solely based on subgradient evaluations. So, in a computer implementation, there is potential for strong improvements if the actual objective values are taken into account in, e.g., a local line search that would adapt the value of γ_k in each step instead of fixing it a priori. The general conditions on γ_k, α_k , the index sets I_k and the weight vectors λ_i^k practically include many step size selection algorithms already in the convergence analysis below.

Note also that the method might arrive in a Pareto optimal point but not immediately ‘recognize’ this as the used subdifferentials are sets not just single points and the underlying optimality conditions are based on $0 \in \partial^\leq$ (so a specific element of ∂^\leq). The conditions on step size parameters, however, ensure that the iterates cannot deviate far away from an already reached optimal point in the iteration.

Remark 3.2 Under the Assumption 3.1, it follows from Propositions 2.1 and 2.2 that for each $i = 1, 2, \dots, m$, there exists $s_i^k \in \partial^\leq f_i(x_k)$ such that $\|s_i^k\| \leq L$. Indeed, for any fixed k and $i \in \{1, 2, \dots, m\}$, if x_k is a global minimizer of f_i , then by Proposition 2.1, one can take $s_i^k = 0$; otherwise, Proposition 2.2 ensures the existence of such an s_i^k . As a result, in view of the strong convexity (ensured by the Tikhonov-type quadratic term) of h_{x_k} , the direction v_k can be obtained.

Remark 3.2 guarantees the well-definedness of our algorithm.

From this point on, we assume without loss of generality that the sequence $\{x_k\}$ generated by the algorithm is infinite.

Remark 3.3

- (i) The algorithm proposed here uses exogenous weights λ_i^k for the subgradients s_i^k in a manner of a weighting method. Following [5,29], the weighting method allows the

resulting scalar objective function (a positive linear combination of the objectives) to be solved for a single Pareto optimal solution that represents the preference of the human decision maker, which is well acceptable in applications and favorable for mathematical analysis.

Note, however, that the algorithm in its general form does not (implicitly or explicitly) fix a weighting and can therefore not be easily recast to a single-objective algorithm applied to the weighted form.

- (ii) The algorithm proposed in the present paper covers several existing algorithms as special cases. For example, in the case when each $I_k = \{1, 2, \dots, m\}$ and the Fenchel–Moreau subdifferential is used, the algorithm presented here reduces to the weighting subgradient algorithm that was introduced in [5], which has been applied to solve the multiobjective optimization problem (1) but under the assumption that each component function f_i is convex.

Assuming for a moment that each f_i is a continuously differentiable quasiconvex function, the sequences $\{\alpha_k\}$ and $\{\eta_k\}$ are constant, and each $I_k = \{i_k\}$ is a singleton with $i_k := \operatorname{argmax}_{i \in \{1, 2, \dots, m\}} \langle \nabla f_i(x_k), v \rangle$, then the algorithm introduced here reduces to the one in [3], which uses an Armijo-like rule to obtain the step lengths.

4 Convergence Analysis

In this section, we show that the sequence $\{x_k\}$ generated by our algorithm converges to a Pareto optimal point for (1). We begin the analysis by showing the feasibility of $\{x_k\}$.

Proposition 4.1 *The sequence $\{x_k\}$ belongs to C .*

Proof The proof is constructed by induction over k . Note that by the initialization of the algorithm, the initial iterate x_0 belongs to C . Assume that $x_k \in C$. Since $\gamma_k \in (0, 1]$ and $v_k \in C - x_k$ (which means that $v_k + x_k \in C$), rewritten $x_{k+1} = x_k + \gamma_k v_k$ as

$$x_{k+1} = (1 - \gamma_k)x_k + \gamma_k(x_k + v_k),$$

we conclude by the convexity of C that x_{k+1} belongs to C . □

We continue the convergence analysis with the following auxiliary results.

Lemma 4.1 *For all $k \in \mathbb{N}$, $\|v_k\| \leq 2\alpha_k$ and $\|x_{k+1} - x_k\| \leq 2\alpha_k$.*

Proof Fix $k \in \mathbb{N}$. In the case when $v_k = 0$, both assertions are trivial. So we assume that $v_k \neq 0$. By Proposition 4.1, we obtain that $0 \in C - x_k$. Thus, one has $h_{x_k}(v_k) \leq h_{x_k}(0) \leq 0$, that is

$$\frac{\alpha_k}{\eta_k} \left\langle \sum_{i \in I_k} \lambda_i^k s_i^k, v_k \right\rangle + \frac{\|v_k\|^2}{2} \leq 0.$$

Then, by applying the Cauchy–Schwarz inequality and recalling that $\sum_{i \in I_k} \lambda_i^k = 1$ and $\|s_i^k\| \leq \eta_k$, one can get

$$\|v_k\|^2 \leq -2 \frac{\alpha_k}{\eta_k} \left\langle \sum_{i \in I_k} \lambda_i^k s_i^k, v_k \right\rangle \leq 2 \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \|s_i^k\| \|v_k\| \leq 2\alpha_k \|v_k\|.$$

Consequently,

$$\|v_k\| \leq 2\alpha_k. \tag{7}$$

Finally, it follows from (6) and (7) that $\|x_{k+1} - x_k\| = \gamma_k \|v_k\| \leq 2\alpha_k$. □

The next proposition establishes a fundamental inequality satisfied by the iterates of the method.

Proposition 4.2 *For all $z \in C$, it holds that*

$$\|x_{k+1} - z\|^2 \leq 8\alpha_k^2 + \|x_k - z\|^2 + 2 \frac{\gamma_k \alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \langle s_i^k, z - x_k \rangle, \quad \forall k \in \mathbb{N}.$$

Proof Let $z \in C$ and $k \in \mathbb{N}$. By using (6) and Lemma 4.1, one has

$$\begin{aligned} \|x_{k+1} - z\|^2 &= \|x_{k+1} - x_k\|^2 + \|x_k - z\|^2 + 2 \langle x_{k+1} - x_k, x_k - z \rangle \\ &\leq 4\alpha_k^2 + \|x_k - z\|^2 + 2\gamma_k \langle v_k, x_k - z \rangle. \end{aligned} \tag{8}$$

Note that from (5) and taking into account the definition of h_{x_k} , we get

$$\begin{aligned} v_k &= \operatorname{argmin}_{v \in C - x_k} \left\{ \frac{\alpha_k}{\eta_k} \left\langle \sum_{i \in I_k} \lambda_i^k s_i^k, v \right\rangle + \frac{\|v\|^2}{2} \right\} \\ &= \operatorname{argmin}_{v \in C - x_k} \left\{ \left\langle \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k s_i^k, v \right\rangle + \frac{1}{2} \left(\|v\|^2 + \left\| \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k s_i^k \right\|^2 \right) \right\} \\ &= \operatorname{argmin}_{v \in C - x_k} \left\{ \frac{1}{2} \left\| v - \left(-\frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k s_i^k \right) \right\|^2 \right\} \\ &= P_{C - x_k} \left(-\frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k s_i^k \right). \end{aligned} \tag{9}$$

Then, by applying (2) with $D = C - x_k$, $x = -\frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k s_i^k$ and $y = z - x_k$, it follows that

$$\left\langle -\frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k s_i^k - v_k, z - x_k - v_k \right\rangle \leq 0,$$

which yields that

$$\begin{aligned}
 \langle v_k, x_k - z \rangle &\leq -\|v_k\|^2 + \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \langle s_i^k, z - x_k \rangle - \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \langle s_i^k, v_k \rangle \\
 &\leq \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \langle s_i^k, z - x_k \rangle + \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \|s_i^k\| \|v_k\| \\
 &\leq \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \langle s_i^k, z - x_k \rangle + \alpha_k \|v_k\| \\
 &\leq \frac{\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k \langle s_i^k, z - x_k \rangle + 2\alpha_k^2,
 \end{aligned}
 \tag{10}$$

where the third inequality holds by the fact that $\sum_{i \in I_k} \lambda_i^k \|s_i^k\| \leq \eta_k$ and the last inequality holds by Lemma 4.1. Consequently, by combining (8) and (10), also note that $\gamma_k \in (0, 1]$, we obtain the desired result. \square

In order to prove the convergence of the sequence $\{x_k\}$ produced by our algorithm, we will assume the following condition.

A1. The set $T := \{z \in C : F(z) \preceq F(x_k), \forall k \in \mathbb{N}\}$ is nonempty.

This assumption is related to the completeness of the image of F , namely, all nonincreasing sequences with respect to \preceq in the image of F have a lower bound. It is important to point out that completeness is a standard assumption for ensuring the existence of Pareto optimal points for vector optimization problems [24]. Furthermore, this condition is standard in the analysis of extensions of classical methods to vector optimization, such as the projected gradient method [2,3,5,6,9,17,18], steepest descent method [4] and proximal point method [8,27,28].

Proposition 4.3 *Assume that A1 holds. Then the sequence $\{x_k\}$ is quasi-Fejér convergent to T and bounded.*

Proof Let $z \in T$. Then, we have that for each $k \in \mathbb{N}$,

$$f_i(z) - f_i(x_k) \leq 0, \quad \forall i \in I_k. \tag{11}$$

Since $s_i^k \in \partial^{\leq} f_i(x_k)$, it holds by definition that

$$\langle s_i^k, z - x_k \rangle \leq f_i(z) - f_i(x_k), \quad \forall i \in I_k. \tag{12}$$

From (12) and Proposition 4.2, we obtain

$$\|x_{k+1} - z\|^2 \leq \|x_k - z\|^2 + 8\alpha_k^2 + 2\frac{\gamma_k \alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k (f_i(z) - f_i(x_k)). \tag{13}$$

Combining this with (11) yields that

$$\|x_{k+1} - z\|^2 \leq \|x_k - z\|^2 + 8\alpha_k^2.$$

Then, by (4) and Definition 2.4 with $\epsilon_k = 8\alpha_k^2$, we conclude that the sequence $\{x_k\}$ is quasi-Fejér convergent to T . The boundedness of $\{x_k\}$ follows from Proposition 2.3. \square

To conclude that the sequence converges to a Pareto optimal point for (1), we need the following requirement.

A2. Assume, there exists a number $\underline{\lambda} > 0$ such that for each $z \in T$, it holds

$$i_k := \operatorname{argmax}_{i=1,2,\dots,m} \{f_i(x_k) - f_i(z)\} \in I_k \quad \text{and} \quad \lambda_{i_k}^k \geq \underline{\lambda}, \quad \forall k \in \mathbb{N}. \tag{14}$$

From the algorithm’s construction, it is clear that such a constant $\underline{\lambda}$ always lies in $(0, 1)$. Now, we present the main convergence result.

Theorem 4.1 *Assume that A1 and A2 hold. Then the sequence $\{x_k\}$ converges to a Pareto efficient solution of the problem (1).*

Proof First, we show that $\{x_k\}$ converges to some $x^* \in T$. Take $z \in T$. Note that $\eta_k \leq L$ for all $k \in \mathbb{N}$. Then, by using the inequality (13), one has

$$\begin{aligned} \frac{2\gamma_k\alpha_k}{L} \sum_{i \in I_k} \lambda_i^k (f_i(x_k) - f_i(z)) &\leq \frac{2\gamma_k\alpha_k}{\eta_k} \sum_{i \in I_k} \lambda_i^k (f_i(x_k) - f_i(z)) \\ &\leq \|x_k - z\|^2 - \|x_{k+1} - z\|^2 + 8\alpha_k^2. \end{aligned} \tag{15}$$

Summing (15) with k between 0 and N , we get

$$\begin{aligned} \frac{2}{L} \sum_{k=0}^N \gamma_k\alpha_k \left[\sum_{i \in I_k} \lambda_i^k (f_i(x_k) - f_i(z)) \right] &\leq \|x_0 - z\|^2 - \|x_{N+1} - z\|^2 + 8 \sum_{k=0}^N \alpha_k^2 \\ &\leq \|x_0 - z\|^2 + 8 \sum_{k=0}^{\infty} \alpha_k^2. \end{aligned}$$

Letting N go to $+\infty$ and noting (4), it follows that

$$\liminf_{k \rightarrow +\infty} \sum_{i \in I_k} \lambda_i^k (f_i(x_k) - f_i(z)) \leq 0.$$

Furthermore, owing to the fact that $z \in T$ and the definition of i_k in (14), we have

$$\begin{aligned} 0 &\leq \liminf_{k \rightarrow +\infty} \lambda_{i_k}^k \max_{i=1,2,\dots,m} \{f_i(x_k) - f_i(z)\} = \liminf_{k \rightarrow +\infty} \lambda_{i_k}^k (f_{i_k}(x_k) - f_{i_k}(z)) \\ &\leq \liminf_{k \rightarrow +\infty} \sum_{i \in I_k} \lambda_i^k (f_i(x_k) - f_i(z)) \leq 0, \end{aligned}$$

and thus

$$\liminf_{k \rightarrow +\infty} \lambda_{i_k}^k \max_{i=1,2,\dots,m} \{f_i(x_k) - f_i(z)\} = 0. \tag{16}$$

As $\lambda_{i_k}^k \geq \underline{\lambda}$ for each $k \in \mathbb{N}$, (16) implies that

$$\liminf_{k \rightarrow +\infty} \max_{i=1,2,\dots,m} \{f_i(x_k) - f_i(z)\} = 0.$$

Then, one can take a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ such that

$$\liminf_{k \rightarrow +\infty} \max_{i=1,2,\dots,m} \{f_i(x_{k_j}) - f_i(z)\} = 0. \tag{17}$$

Since $\{x_k\}$ belongs to C and is bounded by Proposition 4.1 and Proposition 4.3, we may assume that there exists $x^* \in C$ such that $\lim_{j \rightarrow +\infty} x_{k_j} = x^*$. Then, by the continuity of the maximal function, we have from (17) that

$$\max_{i=1,2,\dots,m} \{f_i(x^*) - f_i(z)\} = 0.$$

Consequently, $F(x^*) = F(z)$ because $z \in T$, and so $x^* \in T$. By applying Proposition 4.3 and Proposition 2.3, one gets that $\{x_k\}$ converges to $x^* \in T$. Moreover, from z being arbitrary, we get

$$F(x^*) = F(y), \quad \text{for all } y \in T. \tag{18}$$

Now, we show that x^* is a Pareto efficient solution of (1). In fact, if there exists $y \in C$ such that

$$f_i(y) \leq f_i(x^*), \quad i = 1, 2, \dots, m, \quad i \neq j, \quad f_j(y) < f_j(x^*),$$

then $y \in T$, thus yielding a contradiction to (18). The desired result is proved. \square

Note that in the special case when $I_k = \{1, 2, \dots, m\}$ for all $k \in \mathbb{N}$, then for each $z \in T$, the condition $i_k \in I_k$ in A2 holds automatically. With that in mind, we can think of another condition that is weaker than A2.

A2⁻. Let $I_k = \{1, 2, \dots, m\}$ for any $k \in \mathbb{N}$ and assume, there exists a number $\underline{\lambda} > 0$ such that for each $z \in T$, it holds

$$\lambda_{i_k}^k \geq \underline{\lambda} \quad \text{with} \quad i_k := \operatorname{argmax}_{i=1,2,\dots,m} \{f_i(x_k) - f_i(z)\}, \quad \forall k \in \mathbb{N}.$$

As illustrated before, such a constant $\underline{\lambda}$ is in the interval $(0, 1)$. Then, one can obtain directly from Theorem 4.1 the following result, which improves the ones in [5, Theorem 2] and [6, Theorem 3.2] in twofold: (i) Corollary 4.1 extends the results in [5,6] from the convex case to the quasicontvex case; (ii) the condition $\lambda_i^k \geq \underline{\lambda}$, for all $i = 1, 2, \dots, m$, that is required in [5,6] is weakened to $\lambda_{i_k}^k \geq \underline{\lambda}$ as stated in A2⁻.

Corollary 4.1 *Assume that A1 and A2⁻ hold. Then the sequence $\{x_k\}$ converges to a Pareto efficient solution of the problem (1).*

5 Numerical Results

In this section, we illustrate the functionality of the proposed method by means of several small-scale examples. Note that Condition A2 is a rather weak one introduced in the theoretical proof to ensure that the sequence generated by the algorithm converges to an efficient solution. It may, however, not be easy to ensure the validity of A2 beforehand. So, in the actual implementation of the algorithm, one can still adopt the relatively strong but easy to operate parameter setting that the weights are always bounded from below by a positive constant, specifically, $I_k = \{1, 2, \dots, m\}$ for any $k \in \mathbb{N}$ and $\lambda_i^k \geq \underline{\lambda} > 0$ for each $i \in I_k$. It is seen that, if additionally the weight values λ_i^k are kept constant, the multiobjective optimization problem is then transformed into a scalar one.

We use the first example¹ to illustrate Algorithm 3.1’s parameters and the assumptions in the main convergence result Theorem 4.1.

Example 5.1 Consider the objective function $F: \mathbb{R} \rightarrow \mathbb{R}^2$ given by

$$F = (f_1, f_2) = \begin{pmatrix} \exp(-x) \\ x^2 \end{pmatrix} \quad \text{and} \quad C = \mathbb{R}.$$

This problem is strictly convex and continuously differentiable. The set of Pareto optimal points is given by $\{x \in \mathbb{R} : 0 \leq x\}$, and the Plastia subdifferentials of f_1 and f_2 can be derived from the classical derivatives

$$\begin{aligned} \partial^{\leq} f_1(x) &= \{s \in \mathbb{R} : s \leq \frac{df_1(z)}{dz} \Big|_{z=x}\} = (-\infty, -\exp(-x)], \\ \partial^{\leq} f_2(x) &= \{s \in \mathbb{R} : |s| \geq \frac{df_2(z)}{dz} \Big|_{z=x}, \text{sgn } s = \text{sgn } x\} = \{M \cdot x, M \geq 2\}. \end{aligned}$$

Both components of F are not Lipschitz continuous. However, on any subinterval of the real line that is bounded from below, a Lipschitz constant L_1 for f_1 can be determined and f_2 is Lipschitz continuous on any bounded domain. For the sake of simplicity, we will use $L_1 = 1$ in the following, meaning that we use $-1 \in \partial^{\leq} f_1(x_k)$ when the subdifferential (at $x_k \geq 0$) is needed.

Theorem 4.1 states that the iterates of Algorithm 3.1 converge toward a Pareto efficient solution provided that Assumptions A1 and A2 are fulfilled. To make this implication more clear, we will consider algorithmic parameters that violate them.

We consider the choice

$$\forall k : I_k := \{1\},$$

which is obviously not in accordance with A2 as it completely disregards the second component of F . The problem is not constrained, so the minimization of $h_{x_k}(v)$ can be performed analytically. One obtains

¹ Motivating the algorithm and its parameters using this example was prompted to us by one of the anonymous referees.

$$\min_v h_{x_k}(v) = \min_v \frac{\alpha_k}{\eta_k} \left\langle \lambda_1^k \cdot s_1^k, v \right\rangle + \frac{\|v\|^2}{2} \implies v_k = \alpha_k,$$

where we used $s_1^k = -1$, $\lambda_1^k = 1$ and $\eta_k = 1$ which follow from the above consideration of $\partial^{\leq} f_1$ and the choice for I_k (tacitly assuming that x_k is sufficiently large. For this example, this is without loss of generality.) The iterates of Algorithm 3.1 therefore are explicitly given by

$$x_k = x_0 + \sum_{i=0}^{k-1} \gamma_k \cdot \alpha_k.$$

Taking (4) into account, one sees the divergence of $\{x_k\}$ as $k \rightarrow \infty$.

To avoid that, next we choose the index set $I_k = \{1, 2\}$ and modify A2 to

$$\exists \underline{\lambda} > 0 : \forall k : \lambda_1^k, \lambda_2^k \geq \underline{\lambda},$$

which is yet slightly stronger than A2⁻. For a nonnegative starting value x_0 , the termination of the search direction v_k can be performed analytically. One obtains

$$v_k = \frac{\alpha_k}{\eta_k} \cdot \left(\underbrace{(\lambda_2^k - 1)}_{<0} \cdot s_1^k - \underbrace{\lambda_2^k}_{\geq \underline{\lambda}} \cdot s_2^k \right)$$

with $s_1^k \geq -L_1$, $s_2^k \geq 2 \cdot x_k$, $\frac{\alpha_k}{\eta_k} > 0$. One can see that, however, small $\underline{\lambda}$ is, v_k becomes negative when x_k is sufficiently large. In this case, the sequence $\{x_k\}$ no longer diverges but converges to a Pareto efficient point.

In an actual implementation, these shortcomings might not be relevant after all: As discussed above, a more practical stopping criterion is to exit the main loop whenever the performed steps of the algorithm are sufficiently small. In Fig. 3, we depict results for a numerical study of this example. The (constant) weight λ_1 (x-axis) is varied from zero (corresponding to $I_k = \{2\}$) to one (as in the divergent example above), and $\alpha_k = \frac{1}{k+1}$, $\gamma_k = 1$, have been chosen. The initial value has been fixed to $x_0 = 1$ and we depict the final iterates x^* of Algorithm 3.1 (point markers) and the total number of steps (bars). The iteration was stopped as soon as the step size reduced to below 10^{-4} . Even though the iterates diverge in exact arithmetic, this is not observed for the numerical approximation (which was computed in double precision) even though the number of steps (bar plot in Fig. 3) increases significantly. Note that the overall high iteration numbers for this experiment result from the poor parameter choice and rough representation $\partial^{\leq} f_1(x) \ni -1$ of the Plastia subdifferential.

The next example was given by Bello Cruz et al. [3] as an counterexample to show that their method may fail when the pseudoconvexity for the quasiconvex objective functions required therein was dropped.

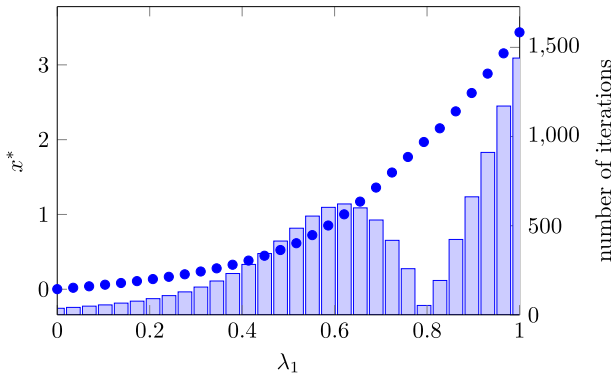


Fig. 3 Numerical results for Example 5.1: x^* (left y-axis, bullets in the plot) indicate the final iterates of Algorithm 3.1, the bars (right y-axis) show the number of iterations before the main loop terminates due to sufficiently small step lengths

Example 5.2 Consider the quasiconvex multiobjective problem defined by

$$F = (f_1, f_2): \mathbb{R} \rightarrow \mathbb{R}^2 \quad \text{with} \quad f_1(x) = \frac{x^5}{5}, \quad f_2(x) = \frac{x^3}{3} \quad \text{for all } x \in \mathbb{R},$$

and the constraint set $C = [-1, 1]$.

It can be verified that the set of stationary points of the problem is $\{-1, 0\}$, and the set of Pareto efficient solutions (as well as the weak Pareto efficient solutions) is the singleton $\{-1\}$. (In this academic example there are no conflicting goals involved.) Recall that a point $x \in C$ is said to be a stationary point if

$$-\mathbb{R}_{++}^m \cap J_F(x)(C - x) = \emptyset,$$

where $J_F(x)$ is the Jacobian matrix of F at x . Note that in the general setting of the algorithm, the notion of a Jacobian matrix does not have to make sense since it requires some amount of smoothness of the objectives. Since the functions f_1 and f_2 in this example are not pseudoconvex, the authors showed in [3] that there are some cases in which the sequence $\{x_k\}$ generated by the method proposed therein converges to 0 when starting at $x_0 \in [0, 1]$. Next, we will underline that our method (given some special algorithmic parameters) is feasible with any initial point in C .

1. In this example, we take $\alpha_k = \frac{1}{k+1}$, $I_k = \{1, 2\}$, $\lambda_1^k = \lambda_2^k = \frac{1}{2}$, $s_1^k = s_2^k = 1$ (It can be computed by definition that $1 \in \partial^{\leq} f_i(x)$ for each $i = 1, 2$ and $x \in C$, so we take these values in the procedure.) and $\gamma_k = \frac{1}{2}$ for all $k \in \mathbb{N}$.

First, we consider the case when the initial point $x_0 \in [-1, 0]$. Then, one can obtain by calculation that

$$x_k = \frac{1}{2^k}x_0 - \frac{2^k - 1}{2^k}, \quad \text{for all } k \in \mathbb{N},$$

and it is seen that x_k converges to -1 . If we consider the stopping criterion $\|x_{k+1} - x_k\| < 0.0001$, an easy calculation shows that at most $k = 14$ iterations are necessary

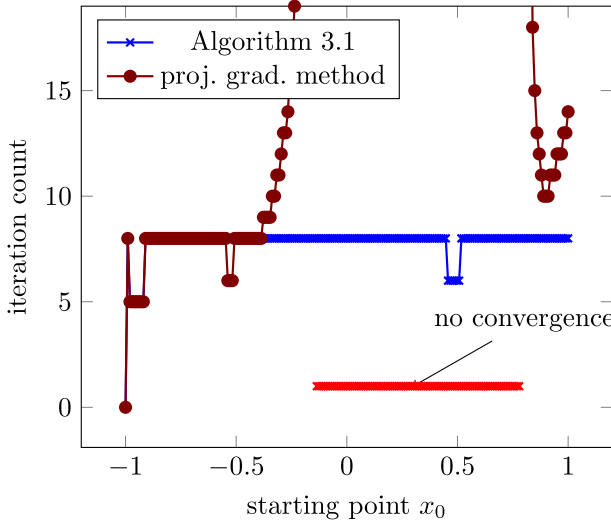


Fig. 4 Numerical results for Example 5.2

for the algorithm to solve the problem. (In an actual implementation, the convergence is much faster.)

Now, consider the case when the initial point $x_0 \in (0, 1]$. Then, it can be computed that after $k = 3$ or $k = 4$ iterations, $x_k \in [-1, 0)$. Consequently, x_k converges to -1 as shown in the above situation.

2. To further illustrate, we used a MATLAB implementation of the algorithm with $\gamma_k = 0.95$, $I_k = \{1, 2\}$, $\lambda_1^k = \lambda_2^k = \frac{1}{2}$, $\alpha_k = \frac{1}{k+1}$ and solved the quadratic programming problems numerically using the routine ‘quadprog.m’ from the Optimization toolbox (interior point method). In Fig. 4, we compare Algorithm 3.1 with the projected gradient algorithm of [3] with comparable solver settings. The iteration limit was set to 200, the step tolerance to 10^{-8} in both cases and the initial guesses were varied along the interval $[-1, 1]$. The crosses marked as ‘no convergence’ indicate the cases when the projected gradient algorithm failed to converge to the actual solution within the iteration limit (92 out of 200 cases). As can be easily seen, Algorithm 3.1 clearly outperforms similar methods based on the classical gradient with no more than $k = 8$ iterations to reach a reasonably good approximation and an average iteration count of 7.79 as opposed to the average of 16.92 iterations for the successful runs of the gradient-based method.

This behavior can even be amplified if instead of using a function with one stationary point, we change to one with multiple such ‘traps.’

Example 5.3 Consider the (for simplicity one-dimensional) objective function

$$f(x) = \frac{1}{\sqrt{2}} \cdot (x + \sin(x)), \quad C = [0, \infty).$$

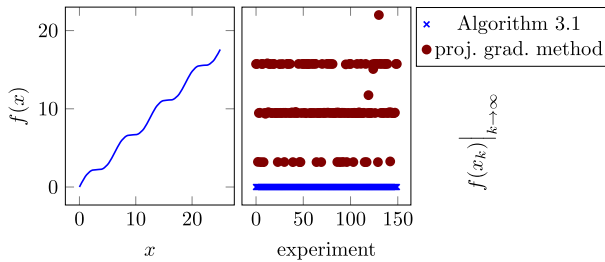


Fig. 5 Numerical results for the function f in Example 5.3, left: objective; right: final solutions

If we use a similar setting as before with $\gamma_k = 0.99$ and $\alpha_k = 5/(k + 1)^{9/10}$, step size tolerance 10^{-7} and iteration limit 500, we obtain the results displayed in Fig. 5. To solve the involved quadratic programs, we used IBM ILOG CPLEX’s quadratic programming capabilities (i.e., the MATLAB routine `cploxqp.m`; note, however, that for this simple one-dimensional example, the projection can easily be performed without additional libraries) [22]. The initial guesses were randomly sampled as $x_0 \in [25, 27]$. Of the 150 experiments, all were solved correctly by Algorithm 3.1 while none(!) of the runs of the gradient-based solvers reached the neighborhood of the Pareto optimal point $E = \{0\}$. The average number of iterations for Algorithm 3.1 was 9.447.

The examples provided so far are for illustrative purposes only and were chosen to underline *the generality* of Algorithm 3.1 and its *convergence* as stated in Theorem 4.1 (in particular when comparing to alternatives from the literature). They are *not* intended to illustrate Algorithm 3.1’s performance or to give a practical application case. In all three examples, a full characterization of the (weak) Pareto optimal points is analytically possible. This characterization can be achieved using methods as outlined, for example, in [14,15] and details are provided in “Appendix” of this article. Actual numerical test cases will follow in the next two examples.

Now, we present Algorithm 3.1 when applied to a problem from facility location theory.

Example 5.4 Consider the problem of finding an optimal location of a building in the two-dimensional Euclidean plane. Optimality means that the weighted sum of (appropriately measured) distances to already existing facilities should be as small as possible. We use the simple mathematical model, [19]

$$f_1(x) = \sum_{i=1}^N \mu_i \cdot \|x - x_i^E\|_{p_i}$$

to implement this requirement, where N is the number of existing facilities, located at $(x_i^E)_{i=1}^N \subseteq \mathbb{R}^2$, the weights $(\mu_i)_{i=1}^N$, $\mu_i \geq 0 \forall i$, measure the importance of that facility and $p_i \in [1, \infty]$ gives a certain degree of freedom in modeling the distances. In the numerical example, we used $N = 7$ existing locations with coordinate values in $[-5, 4]$ and varying weights between 1 and 2. The norms were partially Euclidean

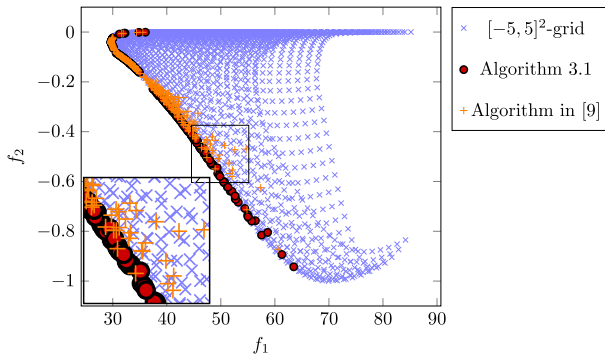


Fig. 6 Objective space view for the facility location problem

($p_i = 2$), partially the Manhattan norm ($p_i = 1$) was inserted, which is very common in facility location problems.

When the weights are all positive (as in our case), the objective function f_1 is convex. Therefore, we introduce a second objective that can measure, for example, the ‘merit distance’ to a hospital. Once a certain distance from that hospital is reached, it does not make a large difference anymore, whether the facility is placed yet further away or not. A quasiconvex function, that can model this is the Gaussian

$$f_2(x) = -\exp\left(-\left(\sum_{i=1}^2 (x_i - x_i^H)^2 / a_i\right)\right),$$

where $x = (x_1, x_2)$, $x^H = (x_1^H, x_2^H)$ is the position of the hospital and the additional parameters a_1, a_2 give the modeler yet more freedom to fine-tune the model.

To derive the Plastia subdifferential to this function (that is f_2), the same heuristic as for the one-dimensional case (see Example 2.2) was used. For the first objective f_1 , the Plastia subdifferential is identical with the Moreau–Fenchel subdifferential and there are established ways to find elements thereof. In Fig. 6, the objective values of the facility location problem along a grid ($x, y \in [-5, 5]$) are displayed as small markers. The larger markers indicate the final iterations of Algorithm 3.1 with $\gamma_k = 0.9$, $\alpha_k = \frac{1}{k+1}$ and 300 initial values drawn randomly from a normal distribution. We chose $I_k = \{1, 2\}$ and $\lambda_1^k = \lambda_2^k = \frac{1}{2}$ for this experiment. The internal optimization problems were not solved by a specialized solver here as constraints were explicitly disregarded for this example. It is apparent how almost all solutions lie along the nonconvex Pareto front of the problem.

To further judge the performance, the optimization process (using the same initial points and comparable numerical settings) was repeated using the algorithm of Cruz Neto et al. [9] (which is similar to Algorithm 3.1 but disregards constraints and the regularization term in $h_{x_k}(v)$). The results are also displayed in Fig. 6. As can be seen, Algorithm 3.1 performs more robust for the given tolerances at this problem. However, the average number of iterations was approximately twelve percent higher (219 instead of 195) and for finer tolerances, both algorithms succeed to approximate

the Pareto front reliably. Given the simple nature of the problem, it is obvious that both algorithms can easily be tuned to better performance by the use of, for example, improved line search strategies in a next step.

Finally, to illustrate the scalability of Algorithm 3.1, we give an example with varying parameters for the dimension n of the decisions variables, the dimension m of the objective space and the constraints imposed for the definition of the constraint set C .

Example 5.5 We consider again the negative Gaussian as in Examples 2.2 and 5.4. For this experiment (cf. Fig. 7), we used the full index set $I_k = \{1, \dots, m\}$, $k = 0, 1, \dots$, a value of $\gamma_k = 0.99$, $k = 0, 1, \dots$, and $\alpha_k := \frac{5 \cdot n}{k+1}$. The minimization of h_{x_k} was again carried out using CPLEX (standard parameters) and the tolerances (in the projected subgradient method) were set to 10^{-4} . The maximum number of iterations was set to 9999 (but in all the optimization runs, this bound was only reached once). The weights λ_i^k were set such that $\forall k : \lambda_1^k = \frac{1}{2}$ and the weights on all other objectives were equally distributed.

For the construction of F at given parameters $n \in \{2, 3, \dots, 10\}$, $m \in \{2, 3, 4\}$ and an integer parameter $p \in \{0, 2\}$, we proceeded 100 times as follows:

1. Choose a $p \times n$ matrix $A_{=}$ and a p -dimensional vector $b_{=}$ at random and let

$$C := \{x \in \mathbb{R}^n : (\forall i : -5 \leq x_i \leq 5) \wedge (A_{=} \cdot x = b_{=})\}.$$

2. Choose an initial point $x_0 \in C$ at random. (in Step 0 of Algorithm 3.1)
3. Define random offset values $o^{(l)} \in [-5, 5]^n$ and (nonlinear) weights $a^{(l)} \in [5, 15]^n$, $l = 1, \dots, m$, and set

$$F(x) := \begin{pmatrix} -\exp\left(-\sum_{i=1}^n \frac{x_i - o_i^1}{a_i^1}\right) \\ \vdots \\ -\exp\left(-\sum_{i=1}^n \frac{x_i - o_i^m}{a_i^m}\right) \end{pmatrix}.$$

In Fig. 7, we display the mean number of iterations (dashed lines) and their medians (solid lines) over the varied dimension n of the decision variables x . The experiments were only carried out if $p < n$ and $m \leq n - p$ was satisfied to ensure that non-Pareto elements of C exist.

A Pareto optimal element could always be found with a single exception for $m = 3$, $p = 0$ and the average number of steps is typically kept below 50, thereby underlining the competitive capacity of Algorithm 3.1.

6 Conclusion and Outlook

In this work, we introduce a projected subgradient method to solve constrained multiobjective optimization problems, where the objective function is assumed to be

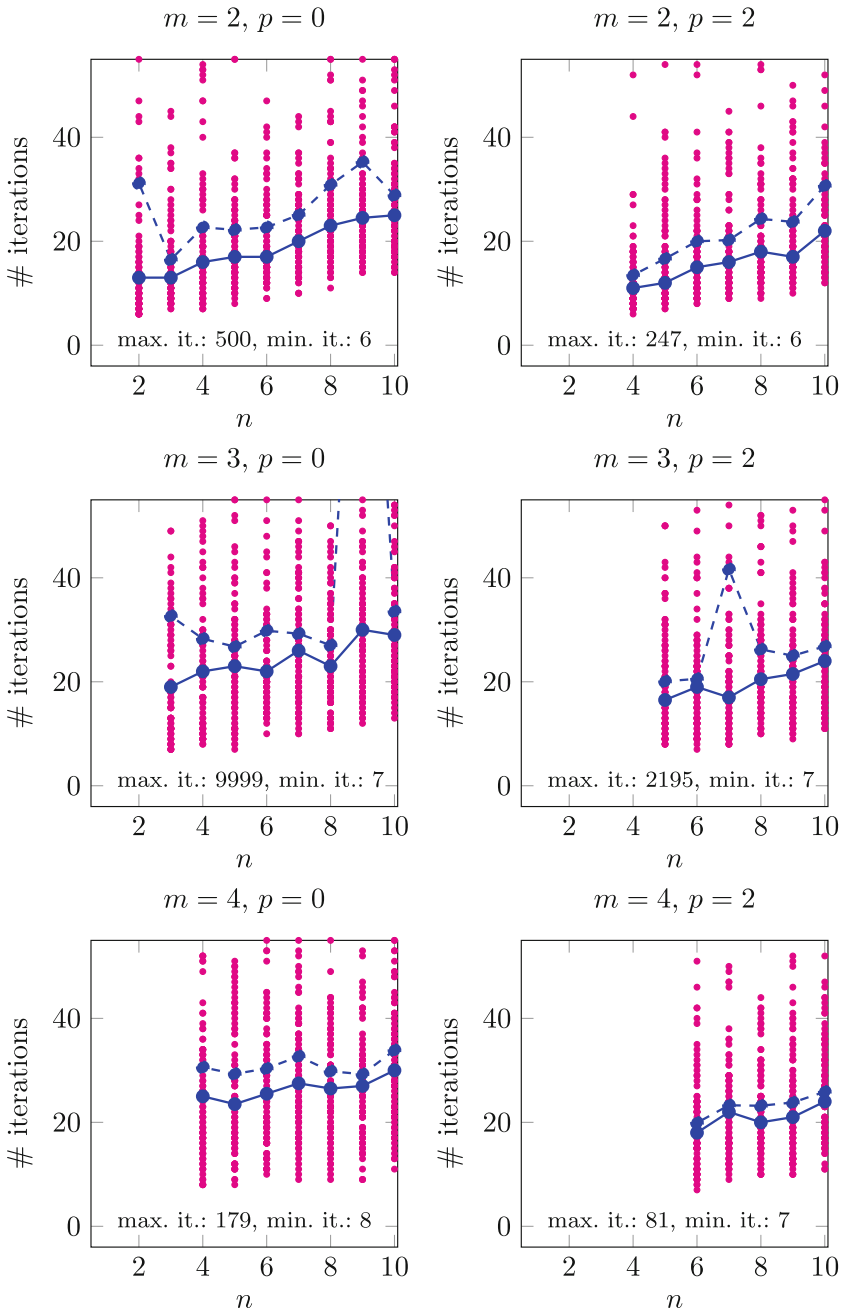


Fig. 7 Numerical results for Example 5.5, dashed lines indicate the average value of iterations, solid lines indicate their median, the disconnected dots (magenta in the online version) display the iteration numbers of every single experiment, the numbers ‘max. it.’ and ‘min. it.’ denote the maximal/minimal number of iterations over all displayed optimization runs in each subplot

componentwise essentially quasiconvex and Lipschitz continuous but not necessarily differentiable. Under reasonable assumptions, we show that for any initial point in the constraint set, the sequence produced by our method converges to a Pareto efficient solution of the problem. Analytical tools included the subdifferential calculus in the sense of Plastria and quasi-Fejér convergence. Numerical examples are presented to indicate the practical validity of our method.

As already indicated earlier, there are several possible extensions on the numerical side like the implementation of a penalty-based version to improve the algorithm's behavior at the boundary of C or improving the stopping criteria of the method. So far, the Plastria subdifferential used was implemented as a deterministic function (always the same subdifferential for the same argument). As, by nature, the subdifferential is a set-valued mapping, extensions in this direction might also open new research directions as do possible extensions of the algorithm that include reusing previous values of the subdifferential (conjugated subdifferential methods or BFGS-type subgradient methods).

The extension of this type of algorithm to more general space is also of great significance, which needs to consider the subdifferential of quasiconvex function in such generalized space, for example the family of quasiconvex subdifferentials presented in [10]. Meanwhile, note that the boundedness property of the subdifferential plays a key role in ensuring the convergence of the algorithm. Therefore, in order to get the expected results in generalized spaces, one needs to explore the related properties of the involving subdifferential.

From the analytical viewpoint, we intend to analyze the rate of convergence of this type and other extended methods for multiobjective optimization problems as a next step.

Acknowledgements The authors are grateful to the handling editor and the anonymous referees for their valuable remarks, comments and new references, which helped to improve the original presentation. The research of Xiaopeng Zhao was supported in part by the National Natural Science Foundation of China (Grant Number 11801411) and the Natural Science Foundation of Tianjin (Grant Number 18JCQNJC01100). The work of M. Köbis was carried out during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Programme of the author at the Norwegian University of Science and Technology. The research of Yonghong Yao was partially supported in part by University Innovation Team of Tianjin [Grant Number TD13-5033]. The research of Jen-Chih Yao was supported in part by the Grant MOST 108-2115-M-039-005-MY3.

Appendix: Analytically Finding Pareto Optimal Solutions in the Small Test Cases

The (weak) Pareto optimal solutions in Examples 5.1–5.3 can be obtained by applying the techniques from [14,15]. We partially adapt the notation of these two references to outline the analysis below:

Example 5.1 Here, we have $f_1 = x^2$, $f_2 = \exp(-x)$ and no constraints (i.e., $C = \mathbb{R}$). In this example, we obtain $\operatorname{argmin}_{x \in C} f_1 = \{0\} = [0, 0] =: [\alpha, \beta]$ (bounded) and $\operatorname{argmin}_{x \in C} f_2 = \emptyset$ (empty), corresponding to 'Case 2' in [14, p. 95]. Then, the inclusion $[0, \infty) \subseteq E_w$ follows from [14, Theorem 4.3(b)]. On

the other hand, note that

$$B_- := \{x \in C : x \geq \beta, f_2(x) = f_2(\beta)\} = [0, 0] =: [\beta, \beta^-]$$

and that $\{x \in C : x < 0, \exp(-x) = \exp(0) = 1\} = \emptyset$. So, we obtain from the second case in [14, Theorem 4.3(b)] that $E_w \cap \{x \in C : x < 0\} = \emptyset$. Consequently, $E_w = [0, \infty)$.

Furthermore, note that f_2 is strictly decreasing. From [15, Corollary 5.5(b)], it can be obtained immediately that $E = [0, \infty)$.

Example 5.2 For $f_1 = \frac{x^5}{5}$, $f_2 = \frac{x^3}{3}$, $C = [-1, 1]$, we get

$$\operatorname{argmin}_{x \in C} f_1 \cap \operatorname{argmin}_{x \in C} f_2 = \{-1\} \neq \emptyset.$$

From [14, Proposition 3.1] (resp. [15, Proposition 2.1]), we therefore conclude directly that $E_w = \{-1\}$ (resp. $E = \{-1\}$).

Example 5.3 For a single objective, we always have the set of (classically) optimal solutions to this one dimension coinciding with E (if existent). From that (or formally also from [14, Proposition 3.1] and [15, Proposition 2.1]), we get $E = E_w = \{0\}$.

References

1. Bauschke, H.H., Borwein, J.M.: On projection algorithms for solving convex feasibility problems. *SIAM Rev.* **38**(3), 367–426 (1996). <https://doi.org/10.1137/s0036144593251710>
2. Bello Cruz, J.Y.: A subgradient method for vector optimization problems. *SIAM J. Optim.* **23**(4), 2169–2182 (2013). <https://doi.org/10.1137/120866415>
3. Bello Cruz, J.Y., Lucambio Pérez, L.R., Melo, J.G.: Convergence of the projected gradient method for quasiconvex multiobjective optimization. *Nonlinear Anal. Theory Methods Appl.* **74**(16), 5268–5273 (2011). <https://doi.org/10.1016/j.na.2011.04.067>
4. Bento, G.C., Cruz Neto, J.X., Oliveira, P.R., Soubeyran, A.: The self regulation problem as an inexact steepest descent method for multicriteria optimization. *Eur. J. Oper. Res.* **235**(3), 494–502 (2014). <https://doi.org/10.1016/j.ejor.2014.01.002>
5. Bento, G.C., Cruz Neto, J.X., Santos, P.S.M., Souza, S.S.: A weighting subgradient algorithm for multiobjective optimization. *Optim. Lett.* **12**(2), 399–410 (2018). <https://doi.org/10.1007/s11590-017-1133-x>
6. Brito, A.S., Cruz Neto, J.X., Santos, P.S.M., Souza, S.S.: A relaxed projection method for solving multiobjective optimization problems. *Eur. J. Oper. Res.* **256**(1), 17–23 (2017). <https://doi.org/10.1016/j.ejor.2016.05.026>
7. Burachik, R., Graña Drummond, L.M., Iusem, A.N., Svaiter, B.F.: Full convergence of the steepest descent method with inexact line searches. *Optimization* **32**(2), 137–146 (1995). <https://doi.org/10.1080/02331939508844042>
8. Ceng, L.C., Yao, J.C.: Approximate proximal methods in vector optimization. *Eur. J. Oper. Res.* **183**(1), 1–19 (2007). <https://doi.org/10.1016/j.ejor.2006.09.070>
9. Da Cruz Neto, J.X., Da Silva, G.J.P., Ferreira, O.P., Lopes, J.O.: A subgradient method for multiobjective optimization. *Comput. Optim. Appl.* **54**(3), 461–472 (2013). <https://doi.org/10.1007/s10589-012-9494-7>
10. Daniilidis, A., Hadjisavvas, N., Martínez-Legaz, J.E.: An appropriate subdifferential for quasiconvex functions. *SIAM J. Optim.* **12**(2), 407–420 (2001). <https://doi.org/10.1137/S1052623400371399>

11. Dinh, N., Goberna, M.A., Long, D.H., López-Cerdá, M.A.: New Farkas-type results for vector-valued functions: a non-abstract approach. *J. Optim. Theory Appl.* **182**(1), 4–29 (2019). <https://doi.org/10.1007/s10957-018-1352-z>
12. Eschenauer, H., Koski, J., Osyczka, A. (eds.): *Multicriteria Design Optimization*. Springer, Berlin Heidelberg (1990). <https://doi.org/10.1007/978-3-642-48697-5>
13. Fliege, J.: OLAF—A general modeling system to evaluate and optimize the location of an air polluting facility. *OR Spektrum* **23**(1), 117–136 (2001). <https://doi.org/10.1007/pl00013342>
14. Flores-Bazán, F., Vera, C.: Weak efficiency in multiobjective quasiconvex optimization on the real-line without derivatives. *Optimization* **58**(1), 77–99 (2009). <https://doi.org/10.1080/02331930701761524>
15. Flores-Bazán, F., Vera, C.: Efficiency in quasiconvex multiobjective nondifferentiable optimization on the real line. *Optimization* pp. 1–23 (2021). <https://doi.org/10.1080/02331934.2021.1892103>
16. Fu, Y., Diwekar, U.M.: An efficient sampling approach to multiobjective optimization. *Ann. Oper. Res.* **132**(1–4), 109–134 (2004). <https://doi.org/10.1023/b:anor.0000045279.46948.dd>
17. Fukuda, E.H., Graña Drummond, L.M.: On the convergence of the projected gradient method for vector optimization. *Optimization* **60**(8–9), 1009–1021 (2011). <https://doi.org/10.1080/02331934.2010.522710>
18. Graña Drummond, L.M., Iusem, A.N.: A projected gradient method for vector optimization problems. *Comput. Optim. Appl.* **28**(1), 5–29 (2004). <https://doi.org/10.1023/b:coap.0000018877.86161.8b>
19. Gromicho, JAd.S.: *Quasiconvex Optimization and Location Theory*. Springer, US (1998). <https://doi.org/10.1007/978-1-4613-3326-5>
20. Gutiérrez, C., Jiménez, B., Novo, V.: On second-order Fritz John type optimality conditions in nonsmooth multiobjective programming. *Math. Program.* **123**(1), 199–223 (2010). <https://doi.org/10.1007/s10107-009-0318-1>
21. Huong, N.T.T., Luan, N.N., Yen, N.D., Zhao, X.: The Borwein proper efficiency in linear fractional vector optimization. *J. Nonlinear Convex Anal.* **20**(12), 2579–2595 (2019)
22. IBM Corp.: *IBM ILOG CPLEX Optimization Studio CPLEX User’s Manual* (2009 et seqq.). Version 12.1
23. Kim, D.S., Phạm, T.S., Tuyen, N.V.: On the existence of Pareto solutions for polynomial vector optimization problems. *Math. Program.* **177**(1–2), 321–341 (2019). <https://doi.org/10.1007/s10107-018-1271-7>
24. Luc, D.T.: *Theory of Vector Optimization*. Springer, Berlin Heidelberg (1989). <https://doi.org/10.1007/978-3-642-50280-4>
25. Maingé, P.E.: Strong convergence of projected subgradient methods for nonsmooth and nonstrictly convex minimization. *Set-Valued Anal.* **16**(7–8), 899–912 (2008). <https://doi.org/10.1007/s11228-008-0102-z>
26. Miettinen, K.M.: *Nonlinear Multiobjective Optimization*. Springer, US (1998). <https://doi.org/10.1007/978-1-4615-5563-6>
27. Papa Quiroz, E.A., Apolinário, H.C.F., Villacorta, K.D., Oliveira, P.R.: A linear scalarization proximal point method for quasiconvex multiobjective minimization. *J. Optim. Theory Appl.* **183**(3), 1028–1052 (2019). <https://doi.org/10.1007/s10957-019-01582-z>
28. Papa Quiroz, E.A., Cruzado, S.: An inexact scalarization proximal point method for multiobjective quasiconvex minimization. *Ann. Oper. Res.* (2020). <https://doi.org/10.1007/s10479-020-03622-8>
29. Pardalos, P.M., Steponavičė, I., Žilinskas, A.: Pareto set approximation by the method of adjustable weights and successive lexicographic goal programming. *Optim. Lett.* **6**(4), 665–678 (2012). <https://doi.org/10.1007/s11590-011-0291-5>
30. Plastria, F.: Lower subdifferentiable functions and their minimization by cutting planes. *J. Optim. Theory Appl.* **46**(1), 37–53 (1985). <https://doi.org/10.1007/bf00938758>
31. Prabuddha, D., Ghosh, J.B., Wells, C.E.: On the minimization of completion time variance with a bicriteria extension. *Oper. Res.* **40**(6), 1148–1155 (1992). <https://doi.org/10.1287/opre.40.6.1148>
32. Rockafellar, R.T.: *Convex Analysis*. Princeton University Press (1970 (reprint 2015))
33. Thomann, J., Eichfelder, G.: A trust-region algorithm for heterogeneous multiobjective optimization. *SIAM J. Optim.* **29**(2), 1017–1047 (2019). <https://doi.org/10.1137/18m1173277>
34. Wang, J., Hu, Y., Yu, C.K.W., Li, C., Yang, X.: Extended Newton methods for multiobjective optimization: majorizing function technique and convergence analysis. *SIAM J. Optim.* **29**(3), 2388–2421 (2019). <https://doi.org/10.1137/18m1191737>
35. White, D.J.: Epsilon-dominating solutions in mean-variance portfolio analysis. *Eur. J. Oper. Res.* **105**(3), 457–466 (1998). [https://doi.org/10.1016/s0377-2217\(97\)00056-8](https://doi.org/10.1016/s0377-2217(97)00056-8)

36. Xu, H., Rubinov, A.M., Glover, B.M.: Strict lower subdifferentiability and applications. *J. Aust. Math. Soc. Ser. B Appl. Math.* **40**(3), 379–391 (1999). <https://doi.org/10.1017/s033427000010961>
37. Alber, Y.I., Iusem, A.N., Solodov, M.V.: On the projected subgradient method for nonsmooth convex optimization in a Hilbert space. *Math. Program.* **81**(1), 23–35 (1998). <https://doi.org/10.1007/bf01584842>
38. Zhao, X., Köbis, M.A.: On the convergence of general projection methods for solving convex feasibility problems with applications to the inverse problem of image recovery. *Optimization* **67**(9), 1409–1427 (2018). <https://doi.org/10.1080/02331934.2018.1474355>
39. Zhao, X., Ng, K.F., Li, C., Yao, J.C.: Linear regularity and linear convergence of projection-based methods for solving convex feasibility problems. *Appl. Math. Optim.* **78**(3), 613–641 (2018). <https://doi.org/10.1007/s00245-017-9417-1>
40. Zhao, X., Yao, Y.: Modified extragradient algorithms for solving monotone variational inequalities and fixed point problems. *Optimization* **69**(9), 1987–2002 (2020). <https://doi.org/10.1080/02331934.2019.1711087>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.