Journal of the Operations Research Society of China (2024) 12:410–427
https://doi.org/10.1007/s40305-022-00410-y

A Nonmonotone Projected Gradient Method for Multiobjective Problems on Convex Sets

Gabrie Aníbal Carrizo[1] · **Nadia Soledad Fazzio**[2] · **María Laura Schuverdt**[2]

Received: 1 December 2020 / Revised: 23 February 2022 / Accepted: 25 February 2022 /
Published online: 16 April 2022

Abstract

In this work we consider an extension of the classical scalar-valued projected gradient method for multiobjective problems on convex sets. As in Fazzio et al. (Optim Lett 13:1365–1379, 2019) a parameter which controls the step length is considered and an updating rule based on the spectral gradient method from the scalar case is proposed. In the present paper, we consider an extension of the traditional nonmonotone approach of Grippo et al. (SIAM J Numer Anal 23:707–716, 1986) based on the maximum of some previous function values as suggested in Mita et al. (J Glob Optim 75:539–559, 2019) for unconstrained multiobjective optimization problems. We prove the accumulation points of sequences generated by the proposed algorithm, if they exist, are stationary points of the original problem. Numerical experiments are reported.

Keywords Multiobjective optimization · Projected gradient methods · Nonmonotone line search · Global convergence

Mathematics Subject Classification 90C29 · 49M37 · 65K05

This research was partially supported by ANPCyT (Nos. PICT 2016-0921 and PICT 2019-02172), Argentina.

✉ María Laura Schuverdt
schuverd@mate.unlp.edu.ar

Gabrie Aníbal Carrizo
gabriel.carrizo@uns.edu.ar

Nadia Soledad Fazzio
nfazzio@mate.unlp.edu.ar

1 Department of Mathematics, National University of the South , Bahía Blanca, Argentina

2 Department of Mathematics, University of La Plata, La Plata, Argentina

1 Introduction

We consider the multiobjective optimization problem (MOP) on convex sets of the form:

$$\text{Minimize } F(x) \text{ s.t. } x \in C, \tag{1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^r$, $F(x) = (F_1(x), \cdots, F_r(x))$ is a continuously differentiable multiobjective function in $\mathbb{R}^n$ and $C \subset \mathbb{R}^n$ is a closed and convex set.

As a concept of optimality, we consider the notion proposed by Pareto in [1]. A feasible point is called Pareto optimum or efficient solution [1] if there is no $x \in C$ such that $F(x) \leqslant F(x^*)$ and $F(x) \neq F(x^*)$. A point $x^* \in C$ is said to be a weak Pareto optimum point or a weakly efficient solution if there is no $x \in C$ such that $F(x) < F(x^*)$.

The general procedure we study in this work is the well-known projected gradient method (PGM) for vector optimization studied previously in [2–5]. These iterative algorithms compute the search direction by solving a convex subproblem that depends on a fixed parameter $\beta > 0$ and then they choose the stepsize on the search direction by using the Armijo line search. Thus, those methods share two essential features:

- they are all descent methods: the objective values decrease with each iteration (in the component-wise sense);
- the projected gradient search direction is computed by using a fixed parameter $\beta > 0$ instead of an exogenous sequence $\{\beta_k\}$, $\beta_k > 0$, as suggested in [6].

Monotone line search methods generate a sequence of feasible iterative points $\{x^k\}$ such that $F(x^{k+1}) < F(x^k)$. Nonmonotone line search methods permit some growth in the values of the objective function, with the purpose of improving the speed of convergence.

In the present paper, we consider an extension of the earliest nonmonotone line search framework developed by Grippo, Lampariello and Lucidi (GLL) for Newton's method, [7], in the scalar case. The nonmonotone line search developed in [7] is based on the maximum functional values of the scalar objective function $f(x)$ on some previous iterates: for $r = 1$, given $\sigma \in (0, 1)$, $M \geqslant 1$ and $d^k \in \mathbb{R}^n$ the step length α_k must satisfy

$$f(x^k + \alpha_k d^k) \leqslant M_k + \sigma \alpha_k \nabla f(x^k)^\top d^k, \tag{2}$$

where $M_k = \max\limits_{0 \leqslant j \leqslant m(k)} f(x^{k-j})$ and $m(k) = \min\{m(k-1) + 1, M\}$.

Nonmonotone line searches were previously used in the context of multiobjective descent methods in [6,8,9]. In [8] the authors use nonmonotone line searches for unconstrained MOP. In [6] the authors present the global analysis of a nonmonotone PGM based on the nonmonotone scheme given in [10]. In [9] the authors propose two nonmonotone gradient algorithms for MOP with a convex objective function based on the nonmonotone line search given by [7].

In the present work, we consider the PGM with the max-type nonmonotone line search and we prove that, without convexity assumptions on the multiobjective function, the accumulation points of the sequences generated by the method are stationary for problem (1). Our convergence result expands the global convergence theorems proved in [2–5]. The major difference between the proposed algorithm and the algorithms in [2–5] is that the PGM defined in those works is implemented with an Armijo-like rule which means that the sequence of functional values decreases monotonously. Even though in [6] a nonmonotone PGM is considered, the main difference with this paper is that the nonmonotone strategy in [6] is given by the average value of a previous function, whereas in this work the nonmonotone search is based on the maximum of some previous functional values. We emphasize that we are not attempting to find, or somewhat characterize, the set of all Pareto optimal solutions, weak or otherwise.

It is well established in the literature of scalar problems that nonmonotone strategies combined with spectral gradient choices of the step length β_k may accelerate the convergence process [11,12]. Taking this into account, we propose to use a variable step length sequence $\{\beta_k\}$ to analyze the efficiency of the technique with some numerical experiments.

This paper is organized as follows. In Sect. 2 we define the implementable nonmonotone projected gradient method. In Sect. 3 we present the convergence analysis of the method. In Sect. 4 we propose a sequence $\{\beta_k\}$ for MOP based on the spectral gradient method. In Sect. 5 we present some numerical experiments. Conclusions and final remarks are given in Sect. 6.

Notation We denote: $\mathbb{N} = \{0, 1, 2, \cdots\}$, $\mathbb{R}_+ = \{t \in \mathbb{R} \mid t \geqslant 0\}$, $\mathbb{R}_{++} = \{t \in \mathbb{R} \mid t > 0\}$, $\mathbb{R}^r_{++} = \mathbb{R}_{++} \times \cdots \times \mathbb{R}_{++}$, $\|\cdot\|$ an arbitrary vector norm. If x and y are two vectors of $\mathbb{R}^n$, we write $x \leqslant y$ if $x_i \leqslant y_i$, $i = 1, \cdots, n$, and $x < y$ if $x_i < y_i$, $i = 1, \cdots, n$, v_i is the $i-$th component of the vector v. If $F : \mathbb{R}^n \to \mathbb{R}^m$, $F = (f_1, \cdots, f_m)$, $JF(x)$ stands for the Jacobian matrix of F at x: $JF(x)$ is a matrix in $\mathbb{R}^{m\times n}$ with entries $(JF(x))_{ij} = \frac{\partial f_i(x)}{\partial x_j}$. If $B \in \mathbb{R}^{n\times n}$ is a positive definite matrix, $\|B\|$ denotes the 2-norm of B. If $K = \{k_0, k_1, k_2, \cdots\}$ is an infinite subset of $\mathbb{N}$ ($k_{j+1} > k_j, \forall j$), we denote $\lim\limits_{k\in K} x^k = \lim\limits_{j\to\infty} x^{k_j}$.

2 The Nonmonotone Projected Gradient Method for MOP

For the multiobjective problem on the convex set C (1) we say that a point $x^* \in C$ is stationary if and only if

$$JF(x^*)(C - \{x^*\}) \cap [-\mathbb{R}^r_{++}] = \varnothing,$$

where $JF(x^*)(C - \{x^*\}) := \{JF(x^*)(x - x^*) : x \in C\}$ and $C - \{x^*\} = \{u - x^* : u \in C\}$.

So $x^* \in C$ is stationary for F if, and only if, $\forall x \in C$, $\max\limits_{i=1,\cdots,r} \{\nabla F_i(x^*)^\top (x-x^*)\} \geqslant 0$.

Stationarity is a necessary, but generally not a sufficient condition, for weak Pareto optimality. See [13].

In order to explain how the search direction is computed we define, for a given point $x \in \mathbb{R}^n$, the function $\varphi_x : \mathbb{R}^n \to \mathbb{R}$ by

$$\varphi_x(v) = \max_{i=1,\cdots,r} \{\nabla F_i(x)^\top v\}. \tag{3}$$

Then, we consider the following convex scalar-valued minimization problem

$$\min_{v \in C-\{x\}} \beta\varphi_x(v) + \frac{\|v\|^2}{2}, \tag{4}$$

where $\beta > 0$ is a parameter. This problem is well-defined and it has a unique optimal solution $v_\beta(x)$, which is called the projected gradient direction, see [4]. Therefore, $v_\beta(x)$ is given by

$$v_\beta(x) = \arg\min_{v \in C-\{x\}} \beta\varphi_x(v) + \frac{\|v\|^2}{2}. \tag{5}$$

Let us call $\theta_\beta(x)$ the optimal value of (4), that is to say,

$$\theta_\beta(x) = \beta\varphi_x(v_\beta(x)) + \frac{\|v_\beta(x)\|^2}{2}. \tag{6}$$

The following proposition characterizes stationary points of the multiobjective problem (1) in terms of $\theta_\beta(\cdot)$ and $v_\beta(\cdot)$. The proof follows from Proposition 3 in [5].

Proposition 1 *Let $\beta > 0$, $v_\beta : C \to \mathbb{R}^n$ and $\theta_\beta : C \to \mathbb{R}$ be given by* (5) *and* (6). *Then, the following statements hold.*

1. $\theta_\beta(x) \leqslant 0$ *for all* $x \in C$.
2. *The function* $\theta_\beta(\cdot)$ *is continuous.*
3. *The following conditions are equivalent*:
   (a) *The point* $x \in C$ *is nonstationary.*
   (b) $\theta_\beta(x) < 0$.
   (c) $v_\beta(x) \neq 0$.

In particular, x is stationary if and only if $\theta_\beta(x) = 0$.

The global convergence result given in Sect. 3 is based on the following properties of v_β and θ_β.

Proposition 2 *Let $\beta > 0$, $v_\beta : C \to \mathbb{R}^n$ and $\theta_\beta : C \to \mathbb{R}$ be given by* (5) *and* (6). *Then,*

1. $\|v_\beta(x)\|^2 \leqslant 2|\theta_\beta(x)|$.
2. $\beta\varphi_x(v_\beta(x)) \leqslant -|\theta_\beta(x)|$.

Proof From the theory of convex analysis and optimization we know that, from (5), there exists $w(x) \in \mathbb{R}^r$ such that

$$\sum_{i=1}^{r} w_i(x) = 1,\ w_i(x) \geqslant 0 \text{ for all } i = 1, \cdots, r, \tag{7}$$

$$w_i(x)(\nabla F_i(x)^\top v_\beta(x) - \varphi_x(v_\beta(x))) = 0 \text{ for all } i = 1, \cdots, r, \tag{8}$$

$$\left(\beta \sum_{i=1}^{r} w_i(x)\nabla F_i(x) + v_\beta(x)\right)^\top (v - v_\beta(x)) \geqslant 0 \text{ for all } v \in C - \{x\}. \tag{9}$$

Then, since (9) holds for all $v \in C - \{x\}$, by considering $v = 0$ we conclude that

$$\left(\beta \sum_{i=1}^{r} w_i(x)\nabla F_i(x) + v_\beta(x)\right)^\top (-v_\beta(x)) \geqslant 0.$$

Thus,

$$\|v_\beta(x)\|^2 \leqslant -\beta \sum_{i=1}^{r} w_i(x)\nabla F_i(x)^\top v_\beta(x). \tag{10}$$

Therefore, it follows from (7) and (8) that $\varphi_x(v_\beta(x)) = \sum_{i=1}^{r} w_i(x)\nabla F_i(x)^\top v_\beta(x)$ and, by (10), we have that

$$\theta_\beta(x) = \beta \sum_{i=1}^{r} w_i(x)\nabla F_i(x)^\top v_\beta(x) + \frac{\|v_\beta(x)\|^2}{2} \leqslant -\|v_\beta(x)\|^2 + \frac{\|v_\beta(x)\|^2}{2}$$
$$= -\frac{\|v_\beta(x)\|^2}{2}.$$

Thus,

$$\|v_\beta(x)\|^2 \leqslant 2|\theta_\beta(x)|,$$

which proves 1.

Since $\theta_\beta(x) \leqslant 0$ we have that $|\theta_\beta(x)| = -\theta_\beta(x)$ and

$$\beta\varphi_x(v_\beta(x)) + |\theta_\beta(x)| = \beta\varphi_x(v_\beta(x)) - \theta_\beta(x) = -\frac{\|v_\beta(x)\|^2}{2}.$$

Therefore,

$$\beta\varphi_x(v_\beta(x)) \leqslant -|\theta_\beta(x)|$$

and the conclusion 2 holds

Observe that, for a nonstationary point $x \in C$ we get $\beta\varphi_x(v_\beta(x)) \leqslant -\frac{\|v_\beta(x)\|^2}{2} < 0$, which implies that $JF(x)v_\beta(x) < 0$ and $v_\beta(x)$ is a descent direction for F at x. In a monotone line search method, the stepsize $\alpha_k > 0$ is chosen so that $F(x^{k+1}) < F(x^k)$, as suggested in [2–5]. In a nonmonotone scheme an increase in the objective values can be allowed as done for PGM in [6] by using the nonmonotone line search based on the average of previous function values as proposed in [10].

In the present work we consider the nonmonotone line search framework developed by Grippo et al. [7] that is based on the maximum functional value on some previous iterates. Given $\sigma \in (0, 1)$, $M \geqslant 1$ and the search direction $v_{\beta_k}(x^k)$, for each $i = 1, \cdots, r$ we set

$$A_i^k = \max\{F_i(x^{k-j}) : 0 \leqslant j \leqslant m(k)\}, \tag{11}$$

where $m(k) = \min\{k, M\}$. Then, the step α_k must satisfy the inequality (in the component-wise sense)

$$F(x^k + \alpha_k v_{\beta_k}(x^k)) \leqslant A^k + \sigma\alpha_k JF(x^k)v_{\beta_k}(x^k), \tag{12}$$

where $A^k = (A_1^k, \cdots, A_r^k)^\top \in \mathbb{R}^r$.

We can now define an extension of the classical PGM using the nonmonotone line search technique (12) for the constrained MOP (1).

Algorithm 1 (PGM-GLL) *Choose* $\sigma \in (0, 1)$, $M > 0$, $0 < \beta_{\min} < \beta_{\max} < \infty$ *and* $\beta_0 \in [\beta_{\min}, \beta_{\max}]$. *Let* $x^0 \in C$ *be an arbitrary starting point. Set* $k = 0$.

Step 1 *Compute the search direction.*

$$v_{\beta_k}(x^k) := \underset{v \in C-\{x^k\}}{\arg\min}\, \beta_k\varphi_{x^k}(v) + \frac{\|v\|^2}{2}. \tag{13}$$

Step 2 *Stopping criterion. Compute* $\theta_{\beta_k}(x^k) = \beta_k\varphi_{x^k}(v_{\beta_k}(x^k)) + \frac{1}{2}\left\|v_{\beta_k}(x^k)\right\|^2$. *If* $\theta_{\beta_k}(x^k) = 0$, *then stop.*

Step 3 *Compute the step length. Let* $m(k) = \min\{k, M\}$ *and choose* α_k *as the largest* $\alpha \in \left\{\frac{1}{2^j} : j = 0, 1, 2, \cdots, \right\}$ *such that* (12) *holds where* $A^k = (A_1^k, \cdots, A_r^k)^\top \in \mathbb{R}^r$ *and* A_i^k *is given by* (11).

Step 4 *Set* $x^{k+1} = x^k + \alpha_k v_{\beta_k}(x^k)$. *Define* β_{k+1} *such that* $\beta_{k+1} \in [\beta_{\min}, \beta_{\max}]$. *Set* $k = k + 1$, *and return to Step* 1.

Observe that Algorithm 1 stops at a feasible stationary point of (1) or it generates an infinite sequence $\{x^k\}$ of nonstationary points.

3 Convergence Analysis

In this section we suppose that Algorithm 1 does not have a finite termination and therefore it generates infinite sequences $\{x^k\}$, $\{v_{\beta_k}(x^k)\}$ and $\{\alpha_k\}$.

Lemma 1 *For each k, $F(x^k) \leqslant A^k$.*

Proof It follows easily from the definition of A^k, see Lemma 5 in [8].

Lemma 2 *Algorithm 1 is well defined: If x^k is not a stationary point, then there exists a stepsize $\alpha_k > 0$ satisfying condition (12).*

Proof Since x^k is nonstationary we have that $\theta_{\beta_k}(x^k) < 0$ and $JF(x^k)v_{\beta_k}(x^k) < 0$. Then, by Proposition 1 in [5] there exists a stepsize $\alpha_k > 0$ satisfying the Armijo condition:

$$F(x^k + \alpha_k v_{\beta_k}(x^k)) \leqslant F(x^k) + \sigma\alpha_k JF(x^k)v_{\beta_k}(x^k).$$

Then, by Lemma 1 the inequality (12) holds and the proof is complete.

Lemma 3 *The scalar sequence $\{\|v_{\beta_k}(x^k)\|\}$ is bounded.*

Proof See Lemma 3 in [6].

Proposition 3 *Let $\{x^k\} \subset \mathbb{R}^n$ be a sequence generated by Algorithm 1 and K an infinite subset of $\mathbb{N}$, $\beta^* > 0$, $x^* \in C$ such that $\lim_{k\in K} x^k = x^*$, $\lim_{k\in K} \beta_k = \beta^*$. Then, there exists K^* an infinite subset of K such that*

$$\lim_{k\in K^*} \theta_{\beta_k}(x^k) = \theta_{\beta^*}(x^*). \tag{14}$$

Proof See Proposition 1 in [6].

Proposition 4 *Let $\{x^k\} \subset \mathbb{R}^n$ be a sequence generated by Algorithm 1. Then*

$$\lim_{k\to\infty} \alpha_k|\theta_{\beta_k}(x^k)| = 0.$$

Proof Follows from the proof of Lemma 8 in [8] and the fact that inequalities 1. and 2. of Proposition 2 hold for the search direction sequence $\{v_{\beta_k}(x^k)\}$.

The following theorem is the main result of this section: the global convergence of Algorithm 1.

Theorem 1 *Let $\{x^k\} \subset \mathbb{R}^n$ be a sequence generated by Algorithm 1. Then, every accumulation point of $\{x^k\}$ is a feasible stationary point of (1).*

Proof Let $x^* \in \mathbb{R}^n$ be an accumulation point of $\{x^k\}$. Then, there exists an infinite subset K of $\mathbb{N}$ such that $\lim_{k\in K} x^k = x^*$. The feasibility of x^* follows combining the fact that C is closed with $x^k \in C$ for all k. The sequence $\{\beta_k\}$ is bounded, then there exists $K_0 \subset K$ and $\beta^* > 0$ such that $\lim_{k\in K_0} \beta_k = \beta^*$.

By Proposition 4 we have the following two cases: (a) $\lim_{k\in K_0} |\theta_{\beta_k}(x^k)| = 0$, or (b) $\lim_{k\in K_0} \alpha_k = 0$.

In the first case, by Proposition 3 we obtain that $\theta_{\beta^*}(x^*) = 0$ which proves that x^* is a stationary point.

Now we assume that (b) holds. Due to Lemma 3, there exist an infinite subset K_1, $K_1 \subset K_0$ and $v^* \in \mathbb{R}^n$ such that $\lim_{k\in K_1} v_{\beta_k}(x^k) = v^*$ and $\lim_{k\in K_1} \alpha_k = 0$. Note that we have

$$\max_{i=1,\cdots,r} \{\beta_k \nabla F_i(x^k)^\top v_{\beta_k}(x^k)\} \leqslant \theta_{\beta_k}(x^k) < 0.$$

So, letting $k \in K_1$ go to infinity we get

$$\max_{i=1,\cdots,r} \{\beta^* \nabla F_i(x^*)^\top v^*\} \leqslant 0. \tag{15}$$

Take now a fixed but arbitrary positive integer q. Since $\lim_{k\in K_1} \alpha_k = 0$, for $k \in K_1$ is large enough, we have $\alpha_k < \frac{1}{2^q}$, which means that the nonmonotone condition in Step 3 of Algorithm 1 is not satisfied for $\alpha = \frac{1}{2^q}$ at x^k:

$$F(x^k + \frac{1}{2^q} v_{\beta_k}(x^k)) \nleqslant A^k + \sigma \frac{1}{2^q} JF(x^k)^\top v_{\beta_k}(x^k),$$

so for each $k \in K_1$ large enough there exists $i = i(k) \in \{1, \cdots, r\}$ such that

$$F_i(x^k + \frac{1}{2^q} v_{\beta_k}(x^k)) > A_i^k + \sigma \frac{1}{2^q} \nabla F_i(x^k)^\top v_{\beta_k}(x^k).$$

Since $\{i(k)\}_{k\in K_1} \subset \{1, \cdots, r\}$, there exist an infinite subset K_2, $K_2 \subset K_1$ and an index i_0 such that $i_0 = i(k)$ for all $k \in K_2$,

$$F_{i_0}(x^k + \frac{1}{2^q} v_{\beta_k}(x^k)) > A_{i_0}^k + \sigma \frac{1}{2^q} \nabla F_{i_0}(x^k)^\top v_{\beta_k}(x^k)$$

and, by Lemma 1 we obtain that

$$F_{i_0}(x^k + \frac{1}{2^q} v_{\beta_k}(x^k)) \geqslant F_{i_0}(x^k) + \sigma \frac{1}{2^q} \nabla F_{i_0}(x^k)^\top v_{\beta_k}(x^k).$$

Taking the limit when $k \in K_2$ goes to infinity in the above inequality, we obtain $F_{i_0}(x^* + 1/2^q v^*) \geqslant F_{i_0}(x^*) + \sigma 1/2^q \nabla F_{i_0}(x^*)^\top v^*$. Since this inequality holds for any positive integer q and for i_0 (depending on q), by Proposition 1 in [5] it follows that $JF(x^*)v^* \nless 0$, then $\max_{i=1,\cdots,r} \{\nabla F_i(x^*)^\top v^*\} \geqslant 0$, which, together with (15) implies $\theta_{\beta^*}(x^*) = 0$. Therefore, we conclude that x^* is a stationary point of (1).

Observe that, if we assume that the mapping $F : \mathbb{R}^n \to \mathbb{R}^r$ is $\mathbb{R}^r$-convex:

$$F(\lambda x + (1-\lambda)z) \leqslant \lambda F(x) + (1-\lambda)F(z)$$

for all $x, z \in \mathbb{R}^n$ and all $\lambda \in [0, 1]$, by using Theorem 3.1 in [14] we can guarantee that every accumulation point is a weak Pareto point:

Corollary 1 *Assume that $F : \mathbb{R}^n \to \mathbb{R}^r$ is $\mathbb{R}^r$-convex and let $\{x^k\} \subset \mathbb{R}^n$ be a sequence generated by Algorithm 1. Then, every accumulation point of $\{x^k\}$ is a weak Pareto optimal solution of* (1).

4 The Sequence β_k

The global convergence analysis of the previous section is independent of the choice of β_k, always between $\beta_{\min}$ and $\beta_{\max}$. In this section we propose a way to choose β_k inspired in the spectral PGM for the scalar case.

We observe that problem (4) is similar to the quadratic subproblem used in [12] to compute the search direction. In [12], in order to solve the constrained scalar problem Minimize $f(x)$ subject to $x \in C$ where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable scalar function in $\mathbb{R}^n$ and $C \subseteq \mathbb{R}^n$ is a closed and convex set, the authors suggest computing the search direction by solving the quadratic problem

$$\text{Minimize } Q_k(d) \text{ s.t. } x^k + d \in C, \tag{16}$$

where $Q_k(d) = \frac{1}{2} d^\top B_k d + \nabla f\left(x^k\right)^\top d$ and $B_k \in \mathbb{R}^{n\times n}$ is a positive definite matrix such that $\|B_k\| \leqslant L$ and $\left\|B_k^{-1}\right\| \leqslant L$ for a fixed scalar $L > 0$. In fact, they consider the particular case: $B_k = \frac{1}{\lambda_k^{spg}} I$, where λ_k^{spg} is the spectral gradient choice defined by

$$\lambda_k^{spg} = \begin{cases} \min\left\{\lambda_{\max}, \max\left\{\lambda_{\min}, \frac{s_k^\top s_k}{y_k^\top s_k}\right\}\right\}, & \text{if } y_k^\top s_k > 0, \\ \lambda_{\max}, & \text{otherwise,} \end{cases}$$

where $s_k = x^k - x^{k-1}$ and $y_k = \nabla f\left(x^k\right) - \nabla f\left(x^{k-1}\right)$.

Firstly, we observe that, by comparing problems (4) and (16) we have that the stationarity condition $\theta_\beta(x) = 0$ for (1) still holds if we consider

$$v_\beta(x) := \underset{v\in C-\{x\}}{\arg\min} \varphi_x(v) + \frac{\|v\|^2}{2\beta}$$

and $\theta_\beta(x) := \varphi_x\left(v_\beta(x)\right) + \frac{\|v_\beta(x)\|^2}{2\beta}$, for any $\beta > 0$.

Secondly, we observe that, if $r = 1$ then $s_k^\top y_k = s_k^\top\left(\nabla f\left(x^k\right) - \nabla f\left(x^{k-1}\right)\right) = \varphi_{x^k}(s_k) - \varphi_{x^{k-1}}(s_k)$. Finally, the previous analysis allows us to consider, in the multiobjective case, a sequence of parameters $\{\beta_k\}$ defined as follows

$$\beta_k = \begin{cases} \min\left\{\beta_{\max}, \max\left\{\beta_{\min}, \frac{s_k^\top s_k}{\varphi_{x^k}(s_k) - \varphi_{x^{k-1}}(s_k)}\right\}\right\}, & \text{if } \varphi_{x^k}(s_k) - \varphi_{x^{k-1}}(s_k) > 0; \\ \beta_{\max}, & \text{otherwise.} \end{cases}$$

Therefore, we have an interesting sequence of parameters $\{\beta_k\}$ that deserves to be analysed from the practical point of view.

5 Numerical Experiments

In order to exhibit the behavior of the algorithm, we have examined a set of problems taken from the literature [14–21]. All of these problems are box-constrained and the details are presented in Table 1. For each problem we have considered 10 starting points. These points were obtained using random points which belong to the feasible region of each problem. The algorithm has been coded in Python 3.0 and the subproblem (4) was solved with the SLSQP solver [22] included as part of SciPy [23]. The code has been executed on a personal computer with an INTEL core i5-7400 3.00 GHz Processor, 8 GB of RAM and Linux Mint 18.3 operating system.

To carry out an in-depth analysis of our method, we have considered two related algorithms: the one proposed in [6] (PGM-ZH) and the monotone line search version of Algorithm 1 (called PGM-monotone), i.e. Algorithm 1 with the Armijo line search in Step 3.

For each of these algorithms, we have presented a version A with $\beta_k = 1$ for all k and a version B where β_k has been updated according to the analysis made in Sect. 4.

We have established the maximum number of iterations as $maxiter = 1\,000$ and the stopping criterion was $\left|\theta_{\beta_k}(x^k)\right| \leqslant \epsilon = 10^{-6}$. A numerical experiment has previously been carried out in order to choose algorithm-specific parameters such as η and M and the line search parameter σ. In this numerical test, all problems have been tested taking into account 10 initial points to combine all the values of $\sigma = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$, $M = 5, 10, 15, 20$ and $\eta = 0.8, 0.85, 0.9$. Furthermore, this test has shown that the monotone line search method was the worst-performing method in the sense that for many points it did not converge to several initial points. Then, we have fixed the parameter σ in order to obtain the greatest number of starting points convergent to $\sigma = 10^{-2}$. Using the same numerical test, we have considered the number of iterations needed to establish convergence of each of the values of η and we have observed that $\eta = 0.8$ worked better for the method when β_k is fixed and updated according to Sect. 4. Finally, for the M parameter of the GLL line search, we have obtained that $M = 15$ was a better option for β_k when fixed and $M = 10$ for β_k when updated following Sect. 4. For these reasons, these values have been used for each case and the bounds of the sequence $\{\beta_k\}$ have been set in $\beta_{\min} = \beta_{\max}^{-1} = 10^{-6}$.

For the purpose of evaluating the performance of these algorithms, in Fig. 1 we have compared the algorithms by using a performance profile for the number of iterations for achieving convergence, according to the method proposed in [24]. We have denoted the set of solvers as $\mathcal{S}$ and the set of problems as $\mathcal{P}$. Furthermore, $it_{p,s}$ denotes the number of iterations used by the solver s to solve problem p, the best $it_{p,s}$ for each p as $it_p^\star = \min\left\{it_{p,s} : s \in \mathcal{S}\right\}$. The distribution function $F_{\mathcal{S}}(it)$ for a method s is defined by

Table 1 List of test problems

	n	m	convex	L	U	Source
Brown and Dennis	4	5	No	$[-25,\ -5,\ -5,\ -1]$	$[25,\ 5,\ 5,\ 1]$	[18][1]
DD1	5	2	No	$[-20,\cdots,-20,]$	$[20,\cdots,20]$	[15][2,3]
FGS	10	3	Yes	$[-2,\cdots,-2]$	$[2,\cdots,2]$	[14][2]
Helical valley	3	3	No	$[-2,\cdots,-2]$	$[2,\cdots,2]$	[18][1]
JOS1	5	2	Yes	$[-2,\cdots,-2]$	$[2,\cdots,2]$	[16][2]
KW2	2	2	No	$[-3,\ -3]$	$[3,\ 3]$	[17]
Linear function rank1	10	4	Yes	$[-1,\cdots,-1]$	$[1,\cdots,1]$	[18][1,2]
Rosenbrock	4	3	No	$[-2,\cdots,-2]$	$[2,\cdots,2]$	[20][1,2]
SD	4	2	Yes	$[-3,\cdots,-3]$	$\left[1,\ \sqrt{2},\ \sqrt{2},\ 1\right]$	[19]
Shifted TRIDIA	4	4	No	$[-1,\cdots,-1]$	$[1,\cdots,1]$	[20][1,2]
TOI4	4	2	Yes	$[-5,\cdots,-5]$	$[2,\cdots,2]$	[20][1]
TRIDIA	3	3	Yes	$[-1,\cdots,-1]$	$[1,\cdots,1]$	[20][2]
Trigonometric	4	4	No	$[-1,\cdots,-1]$	$[1,\cdots,1]$	[18][1,2]
ZDT1	30	2	Yes	$[0,\cdots,0]$	$[1,\cdots,1]$	[21][2]
ZDT4	10	2	No	$[0,\ -5,\cdots,-5]$	$[1,\ 5,\cdots,5]$	[21][2]

[1] It is an adaptation of a single-objective optimization problem to the multiobjective setting. Since the problem is originally unconstrained, bound constraints were added. We have used the adaptation proposed in [8].
[2] In the original version, either the bounds L and U or the variable n can be modified. We have used the values proposed in [8].
[3] This is actually a modified version of [15] which can be found in [8,14]

$$F_{\mathcal{S}}(it) = \frac{\left|\left\{p \in \mathcal{P} :\ it_{p,s} \leqslant it_p^\star\right\}\right|}{|\mathcal{P}|}.$$

To present the results more precisely, we have used a semilogarithmic scale in Fig. 1.

Due that line searches increase the number of function evaluations, which could be a drawback for these algorithms, we also consider a performance profiles for function evaluations in Fig. 2. This second figure presents performance profiles for function evaluations using a semi logarithmic scale too and shows that the performances measured in number of function evaluations and iterations are similar.

We can observe that nonmonotone line searches are faster than monotone ones and use less function evaluations and iterations. The algorithm considering the average line search outperforms the other line searches, at least in our test problems. Also, it is clear the relevance of the use of the sequence $\{\beta_k\}$ defined in Sect. 4 as follows: The performance of all algorithms has been improved using the sequence or parameters $\{\beta_k\}$. In particular, the PGM-ZH Algorithm was the worst-performing algorithm when β_k is fixed, whereas it becomes the best-performing one when β_k is updated according to Sect. 4.

As we have already mentioned in the introduction, we are not attempting to find the set of all Pareto or weak Pareto points. Nevertheless, for the sake of completeness,

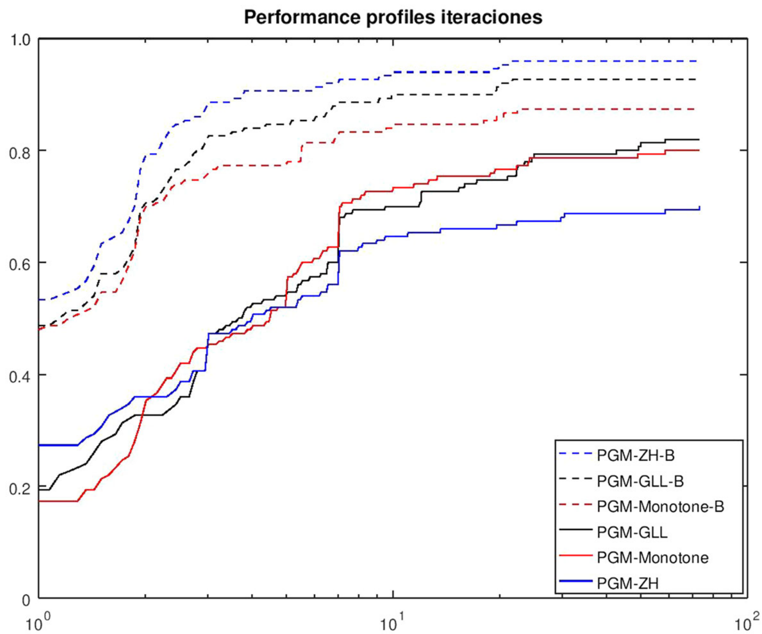

Fig. 1 Performance profiles for projected gradient algorithms with different line search by using the number of iterations as a comparison measurement

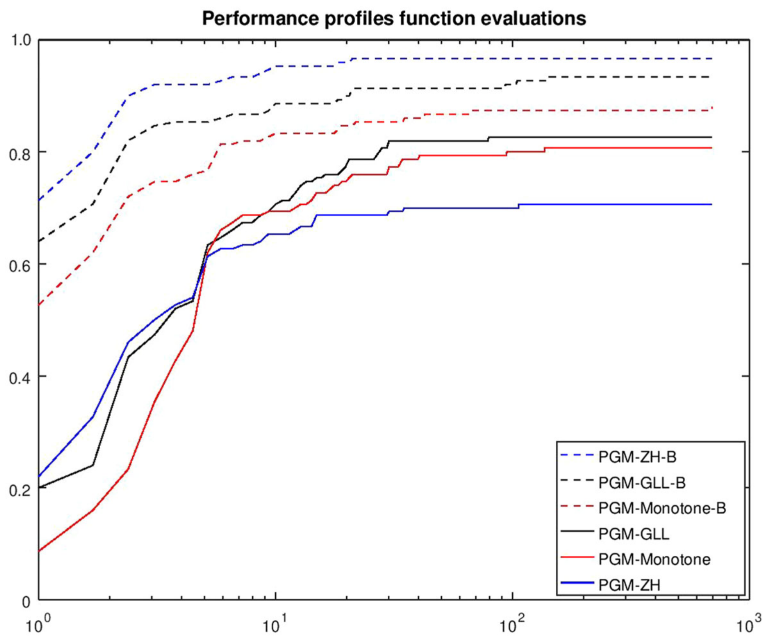

Fig. 2 Performance profiles for projected gradient algorithms with different line search by using the number of function evaluations as a comparison measurement

we have also taken into account some metrics used for evaluating the accuracy in approximation of Pareto fronts. If x is an efficient solution (weakly efficient), then $F(x)$ is a nondominated (weakly non-dominated) point. Let $F_{p,s}$ denote the approximated nondominated front determined by the solver $s \in \mathcal{S}$ for problem $p \in \mathcal{P}$. Let also F_p denote an approximation to the exact nondominated front of problem p, calculated by first forming $\cup_{s \in \mathcal{S}} F_{p,s}$ and then removing from this set any dominated points. Thus, we have considered the following metrics:

- Purity metric [25,26] is used to compare the nondominated fronts obtained by different solvers. It consists then of computing, for solver $s \in \mathcal{S}$ and problem $p \in \mathcal{P}$, the ratio $c_{p,s}^{F_p}/c_{p,s}$, where $c_{p,s}^{F_p} = \left|F_{p,s} \cap F_p\right|$ and $c_{p,s} = \left|F_{p,s}\right|$. This metric is then represented by a number between zero and one. Higher values for this ratio indicate a better nondominated front in terms of the percentage of nondominated points.
- Spread metric [25,26] attempts to measure the maximum size of the holes of an approximated nondominated front. Let us assume that solver $s \in \mathcal{S}$ has computed, for problem $p \in \mathcal{P}$, an approximated nondominated front with N points, indexed by $1, \cdots, N$. We have also considered the extreme points indexed by 0 and $N+1$, these points are computed considering the minimum and maximum value of each objective function among all the points in F_p. The metric $\Gamma_{p,s} > 0$ consists of setting:

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

  where $\delta_{i,j} = f_{i+1,j} - f_{i,j}$ (and we assume that the objective function values have been sorted out by increasing order for each objective j).
- Additive epsilon indicator [25,26]: The additive epsilon indicator $I_{\epsilon+}$ is based on additive ϵ-dominance:

$$z_1 \preceq_{\epsilon+} z_2 \Leftrightarrow z_i^1 \leqslant \epsilon + z_i^2, \forall i \in \{1, \cdots, m\},$$

  and defined, for each solver $s \in \mathcal{S}$ and problem $p \in \mathcal{P}$, with respect to a nondominated reference set F_p as:

$$I_{\epsilon+}\left(F_{p,s}\right) = I_{\epsilon+}\left(F_{p,s}, F_p\right) = \inf\left\{\epsilon \mid \forall y \in F_p \ \exists\, z \in F_{p,s} : z \preceq_{\epsilon+} y\right\}.$$

These metrics are presented in Tables 2, 3 and 4 where the values indicated with— correspond to the combinations of solvers and problems such that none of the starting points produce convergent sequences.

Table 5 presents a summary of the average number of iterations while Table 6 shows the average number of function evaluations. Finally, Table 7 illustrates the number of instances the problem was solved by each solver. In these tables the best average number of function evaluations and of iterations were written in boldface. Taking these numbers into account, we can observe that for the three line search strategies presented the sequence $\{\beta_k\}$ defined in Sect. 4 reduces, most of the time, the number of function evaluations used, at least in our test problems.

Table 2 The value of purity metric in solving test problems

	PGM-Monotone	PGM-Monotone-B	PGM-GLL	PGM-GLL-B	ZH	PGM-ZH-B
Brown and Dennis	0.667	0.75	0.667	0.571	0.333	1.0
DD1	1.0	1.0	1.0	1.0	1.0	1.0
FGS	1.0	1.0	1.0	1.0	1.0	1.0
Helical valley	1.0	0.6	0.125	0.8	0.0	0.75
JOS1	0.0	1.0	0.0	1.0	0.0	1.0
KW2	1.0	1.0	1.0	1.0	1.0	1.0
Linear function rank1	1.0	1.0	1.0	1.0	1.0	1.0
Rosenbrock	0.667	0.286	–[1]	1.0	1.0	0.667
SD	1.0	1.0	1.0	1.0	1.0	1.0
Shifted TRIDIA	1.0	1.0	0.75	1.0	1.0	0.667
TOI4	0.2	0.0	0.0	0.0	1.0	0.0
TRIDIA	0.375	0.375	0.556	1.0	0.4	1.0
Trigonometric	0.0	1.0	0.0	0.714	0.0	0.571
ZDT1	0.2	0.5	1.0	0.667	1.0	0.667
ZDT4	0.5	1.0	0.25	1.0	0.667	1.0

[1] For this problem and algorithm none of the starting points achieves convergence

Table 3 The value of spread metric in solving test problems

	PGM-Monotone	PGM-Monotone-B	PGM-GLL	PGM-GLL-B	ZH	PGM-ZH-B
Brown and Dennis	23.35	23.35	23.339	23.35	23.35	23.35
DD1	541.737	541.172	489.321	541.172	489.321	541.172
FGS	475.535	466.555	460.919	466.183	460.919	466.183
Helical valley	2 974.474	2 974.475	2 974.362	2 974.345	2 974.355	2 974.475
JOS1	2.4	2.399	2.4	2.399	2.4	2.399
KW2	2.967	4.31	2.433	4.31	2.494	4.31
Linear function rank1	8.614	8.986	8.614	8.986	8.614	8.986
Rosenbrock	1 169.153	1 240.275	–[1]	1 169.273	1 168.338	1 210.844
SD	6.147	6.132	6.147	6.132	6.147	6.132
Shifted TRIDIA	5.146	5.146	4.824	5.201	5.0	4.577
TOI4	1.273	1.155	2.139	4.612	0.441	4.612
TRIDIA	6.841	5.455	5.411	5.302	5.454	5.302
Trigonometric	1.154	1.137	1.129	1.146	1.129	1.155
ZDT1	1.245	1.274	1.633	1.274	1.633	1.274
ZDT4	53.61	56.747	42.591	56.747	56.69	56.747

[1] For this problem and algorithm none of the starting points achieves convergence

Table 4 The value of epsilon indicator in solving test problems

	PGM-Monotone	PGM-Monotone-B	PGM-GLL	PGM-GLL-B	ZH	PGM-ZH-B
Brown and Dennis	1.866	0.218	23.339	15.252	23.058	23.045
DD1	99.537	127.181	93.165	127.181	93.165	127.181
FGS	0.693	8.713	28.59	8.713	28.59	8.713
Helical valley	2.08	0.0	2 924.495	149.859	2 924.495	149.859
JOS1	0.0	0.0	0.0	0.0	0.0	0.0
KW2	0.934	1.169	1.877	0.686	1.108	0.686
Linear function rank1	0.771	0.165	4.496	0.165	2.49	0.165
Rosenbrock	1 169.118	2.134	–[1]	441.618	1 168.338	436.59
SD	0.022	0.06	0.022	0.06	0.022	0.06
Shifted TRIDIA	5.146	5.146	4.068	0.322	5.0	1.514
TOI4	0.0	0.0	0.441	0.0	0.357	0.0
TRIDIA	1.107	0.006	2.148	0.128	1.107	0.128
Trigonometric	0.679	1.13	1.129	0.31	1.129	0.001
ZDT1	0.514	1.161	1.633	0.0	1.633	0.0
ZDT4	3.121	56.747	32.448	56.747	0.0	56.747

[1] For this problem and algorithm none of the starting points achieves convergence

Table 5 Number of iterations for each algorithm

Problem	PGM-Monotone	PGM-Monotone-B	PGM-GLL	PGM-GLL-B	PGM-ZH	PGM-ZH-B
Brown and Dennis	69.4	**12.7**	260.0	15.1	81.6	19.0
DD1	73.4	75.556	**37.7**	70.5	**37.7**	69.9
FGS	192.2	199.1	**23.8**	194.0	**23.8**	194.0
Helical valley	**3.167**	36.875	58.6	66.2	18.667	52.125
JOS1	13.9	**2.0**	13.9	**2.0**	13.9	**2.0**
KW2	43.0	**6.25**	152.5	7.667	75.4	7.222
Linear function rank1	2.0	2.1	**1.0**	2.2	8.7	2.2
Rosenbrock	365.667	**161.5**	–	275.4	235.5	281.7
SD	18.1	**6.1**	18.1	**6.1**	18.1	**6.1**
Shifted TRIDIA	38.0	42.0	67.75	25.167	**8.5**	19.667
TOI4	41.4	**3.1**	59.0	3.2	64.7	3.2
TRIDIA	11.5	**3.9**	13.6	7.9	23.4	7.9
Trigonometric	10.0	34.875	14.2	**7.9**	10.75	30.3
ZDT1	9.8	7.3	6.0	**2.0**	6.0	**2.0**
ZDT4	35.9	45.9	**21.4**	45.9	28.2	45.9

Table 6 Number of function evaluations for each algorithm

Problem	PGM-Monotone	PGM-Monotone-B	PGM-GLL	PGM-GLL-B	PGM-ZH	PGM-ZH-B
Brown and Dennis	94.7	22.7	265.333	**17.5**	88.5	22.5
DD1	144.0	77.667	**39.2**	72.0	**39.2**	71.5
FGS	923.4	202.0	**24.9**	195.1	**24.9**	195.1
Helical valley	**17.5**	602.125	310.5	70.0	86.333	941.375
JOS1	14.9	**3.0**	14.9	**3.0**	14.9	**3.0**
KW2	62.8	**9.125**	254.0	11.0	151.5	10.444
Linear function rank1	11.0	6.3	6.25	6.2	41.9	**6.2**
Rosenbrock	3 207.667	**256.25**	–	283.0	1 694.0	331.0
SD	19.1	**7.1**	19.1	**7.1**	19.1	**7.1**
Shifted TRIDIA	114.0	44.0	104.75	170.167	**13.0**	21.667
TOI4	44.0	4.8	61.0	**4.5**	67.0	**4.5**
TRIDIA	32.9	**8.5**	20.3	10.6	71.5	10.6
Trigonometric	**11.0**	1595.5	18.4	44.4	11.75	682.6
ZDT1	14.9	10.7	7.0	**3.0**	7.0	**3.0**
ZDT4	131.3	**49.0**	64.4	**49.0**	95.0	**49.0**

Table 7 Number of times each problem was solved for each algorithm

Problem	PGM-Monotone	PGM-Monotone-B	PGM-GLL	PGM-GLL-B	PGM-ZH	PGM-ZH-B
Brown and Dennis	10	10	3	10	10	10
DD1	10	9	10	10	10	10
FGS	10	10	10	10	10	10
Helical valley	6	8	10	10	6	8
JOS1	10	10	10	10	10	10
KW2	10	8	8	9	10	9
Linear function rank1	10	10	4	10	10	10
Rosenbrock	3	8	0	10	2	10
SD	10	10	10	10	10	10
Shifted TRIDIA	1	1	4	6	2	3
TOI4	10	10	2	10	10	10
TRIDIA	10	10	10	10	10	10
Trigonometric	1	8	5	10	4	10
ZDT1	10	10	10	10	10	10
ZDT4	10	10	10	10	10	10

6 Conclusions

We have presented a modification of the well-known projected gradient method for multiobjective optimization on convex sets. At each iteration the search direction was computed by considering a variable step length instead of a fixed parameter. The novelty element has been the use of the specific max-type nonmonotone line search instead of the classical Armijo strategy or the average-type update rule.

Stationarity of the accumulation points of the sequences generated by the proposed algorithm has been established in the general case under standard assumptions.

The method was implemented and tested. For a better analysis we also consider the monotone version of the presented algorithm and another algorithm with the non-monotone line search defined in [6], these three algorithms were considered with the standard step length and the spectral step length. In our experimentation we observe a better performance for the nonmonotone line searches and even better using the spectral step length. As in the scalar case, the nonmonotone line search shows a good performance in combination with spectral step length.

Acknowledgements We are deeply indebted to the anonymous Referees for their insightful comments and invaluable suggestions. They have significantly improved the quality of the paper.

References

[1] Pareto, V.: Manual D'economie Politique. F. Rouge, Lausanne (1896)
[2] Fukuda, E.H., Graña Drummond, L.M.: On the convergence of the projected gradient method for vector optimization. Optimization **60**, 1009–1021 (2011)
[3] Fukuda, E.H., Graña Drummond, L.M.: Inexact projected gradient method for vector optimization. Comput. Optim. Appl. **54**, 473–493 (2013)
[4] Fukuda, E.H., Graña Drummond, L.M.: A survey on multiobjective descent methods. Pesq. Oper. **34**, 585–620 (2014)
[5] Graña Drummond, L.M., Iusem, A.N.: A projected gradient method for vector optimization problems. Comput. Optim. Appl. **28**, 5–29 (2004)
[6] Fazzio, N.S., Schuverdt, M.L.: Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems. Optim. Lett. **13**, 1365–1379 (2019)
[7] Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
[8] Mita, K., Fukuda, E.H., Yamashita, N.: Nonmonotone line searches for unconstrained multiobjective optimization problems. J. Glob. Optim. **75**, 63–90 (2019)
[9] Qu, S., Ji, Y., Jiang, J., Zhang, Q.: Nonmonotone gradient methods for vector optimization with a portfolio optimization application. Eur. J. Oper. Res. **263**, 356–366 (2017)
[10] Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Opt. **14**, 1043–1056 (2004)
[11] Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**, 1196–1211 (1999)
[12] Birgin, E.G., Martínez, J.M., Raydan, M.: Inexact spectral projected gradient methods on convex sets. IMA J. Numer. Anal. **23**, 539–559 (2003)
[13] Luc, D.T.: Theory of Vector Optimization. Lecture Notes in Economics and Mathematical Systems, vol. 319. Springer (1989)
[14] Fliege, J., Graña Drummond, L.M., Svaiter, B.F.: Newton's method for multiobjective optimization. SIAM J. Optim. **20**, 602–626 (2009)
[15] Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J. Optim. **8**, 631–657 (1998)
[16] Jin, Y., Olhofer, M., Sendhoff, B.: Dynamic weighted aggregation for evolutionary multi-objective optimization: why does it work and how? In: GECCO'01 Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, pp. 1042–1049 (2001)
[17] Kim, I.Y., de Weck, O.L.: Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. Struct. Multidiscipl. Optim. 29149–158 (2006)
[18] Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**, 17–41 (1981)

[19] Stadler, W., Dauer, J.: Multicriteria optimization in engineering: a tutorial and survey. In: Kamat, M.P. (ed.). Progress in Aeronautics and Astronautics: Structural Optimization: Status and Promise, vol. 150, pp. 209–249. American Institute of Aeronautics and Astronautics (1992)
[20] Toint, P.L.: Test problems for partially separable optimization and results for the routine PSPMIN. Tech. Rep. 83/4, Department of Mathematics, University of Namur, Brussels, Belgium (1983)
[21] Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evolut. Comput. **8**, 173–195 (2000)
[22] Kraft, D.: A software package for sequential quadratic programming. Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center, Institute for Flight Mechanics, Koln, Germany (1988)
[23] Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001). http://www.scipy.org/
[24] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
[25] Custódio, A.L., Madeira, J.F A., Vaz, A.I F., Vicente, L.N.: Direct multisearch for multiobjective optimization. SIAM J. Optim. **21**(3), 1109–1140 (2011)
[26] Morovati, V., Pourkarimi, L.: Basirzadeh: Barzilai and Borwein's method for multiobjective optimization problems. Numer. Algorithm **72**, 539–604 (2016)