A subgradient method with non-monotone line search

O P Ferreira, G N Grapiglia, E M Santos, J C O Souza

A subgradient method with non-monotone line search

O. P. Ferreira[1] · G. N. Grapiglia[2] · E. M. Santos[3] · J. C. O. Souza[4,5]

Abstract
In this paper we present a subgradient method with non-monotone line search for the minimization of convex functions with simple convex constraints. Different from the standard subgradient method with prefixed step sizes, the new method selects the step sizes in an adaptive way. Under mild conditions asymptotic convergence results and iteration-complexity bounds are obtained. Preliminary numerical results illustrate the relative efficiency of the proposed method.

Keywords Subgradient method · Non-monotone line search · Convex function

1 Introduction

The subgradient method for solving non-differentiable convex optimization prob-lems has its origin in the 60's, see [7, 26]. Over the years it has been the subject of much interest, attracting the attention of the scientific community working on

✉ O. P. Ferreira
orizon@ufg.br

G. N. Grapiglia
geovani.grapiglia@uclouvain.be

E. M. Santos
elianderson.santos@ifma.edu.br

J. C. O. Souza
joaocos.mat@ufpi.edu.br

[1] Instituto de Matemática e Estatística, Universidade Federal de Goiás, Goiânia, GO CEP 74001-970, Brazil

[2] ICTEAM/INMA, Université Catholique de Louvain, Avenue Georges Lemaître, 4-6/ L4.05.01, 1348 Louvain-la-Neuve, Belgium

[3] Instituto Federal de Educação, Ciência e Tecnologia do Maranhão, Açailândia, MA CEP 65930-000, Brazil

[4] AMSE, CNRS, Aix-Marseille University, Marseille, France

[5] Department of Mathematics, Federal University of Piauí, Teresina, PI, Brazil

convex optimization. One of the factors that explains the interest in the subgradient method lies in its simplicity and ease of implementation for a wide range of problems, where the sub-differential of the objective function can be easily computed. In addition, this method has low storage cost and ready exploitation of separability and sparsity, which makes it attractive in solving large-scale problems. For all these reasons, several variants of this method have emerged and properties of it have been discovered throughout the years, resulting in a wide literature on the subject; including for example [2, 8, 15, 16, 20–23] and the references therein.

The classical subgradient method employs a predefined sequence of step sizes. Standard choices include a constant step size and also sequences that converge to zero sublinearly. In this paper, we propose a subgradient method with adaptive step sizes for the minimization of convex functions with simple convex constraints in which a projection on it is easily computed. At each iteration, the selection of the step size is done by a line search in the direction opposite to the subgradient. Since, in general, this direction is not a descent direction, we endow the method with a non-monotone line search. The possible increase in the objective function values at consecutive iterations is limited by a sequence of positive parameters that implicitly controls the step sizes. Remarkably, it is shown that the proposed method enjoys convergence and complexity properties similar to the ones of the classical subgradient method when the sequence that controls the non-monotonicity satisfies suitable conditions. Illustrative numerical results are also presented. They show that the proposed non-monotone method compares favorably with the classical subgradient method endowed with usual prefixed step sizes. It is worth mentioning that other modified versions of the subgradient method with non-monotonic linear search proposed in [12] were considered in [17–19], see also [14].

The organization of the paper is as follows. In Sect. 2, we present some notation and basic results used in our presentation. In Sect. 3 we describe the subgradient method with non-monotone line search and the main results of the present paper, including the converge theorems and iteration-complexity bounds. Some numerical experiments are provided in Sect. 4. We conclude the paper with some remarks in Sect. 5.

2 Preliminaries

In this section we present some notations, definitions, and results that will be used throughout the paper, which can be found in [1, 13].

Denotes $\mathbb{N} := \{1, 2, 3, \ldots\}$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be σ-strongly convex with modulus $\sigma \geq 0$ if $f(\tau x + (1-\tau)y) \leq \tau f(x) + (1-\tau)f(y) - \frac{\sigma}{2}\tau(1-\tau)\|x-y\|^2$, for all $x, y \in \mathbb{R}^n$ and $\tau \in [0, 1]$. For $\sigma = 0$ we say that f is a convex function.

Proposition 1 *The function $f : \mathbb{R}^n \to \mathbb{R}$ is σ-strongly convex with modulus $\sigma \geq 0$ if and only if $f(y) \geq f(x) + \langle v, y - x\rangle + (\sigma/2)\|y - x\|^2$, for all $x, y \in \mathbb{R}^n$ and all $v \in \partial f(x)$.*

A function $f : \mathbb{R}^n \to \mathbb{R}$ is $L_{f,\mathcal{C}}$-*Lipschitz continuous on $\mathcal{C} \subset \mathbb{R}^n$ if there exist a constant $L_{f,\mathcal{C}} > 0$* such that $|f(x) - f(y)| \leq L_{f,\mathcal{C}}\|x - y\|$, for all $x, y \in \mathcal{C}$. Whenever $\mathcal{C} = \mathbb{R}^n$ we set $L_f \equiv L_{f,\mathbb{R}^n}$.

Proposition 2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex. Then, for all $x \in \mathbb{R}^n$ the set $\partial f(x)$ is a non-empty, convex, compact subset of $\mathbb{R}^n$. In addition, f is $L_{f,\mathcal{C}}$-Lipschitz function on $\mathcal{C} \subset \mathbb{R}^n$ if and only if $\|v\| \leq L_{f,\mathcal{C}}$ for all $v \in \partial f(x)$ and $x \in \mathcal{C}$.*

Remark 1 In view of Proposition 2, if $\mathcal{C} \subset \mathbb{R}^n$ is a compact set then f is a $L_{f,\mathcal{C}}$-Lipschitz function on $\mathcal{C} \subset \mathbb{R}^n$ for some $L_{f,\mathcal{C}} > 0$.

Definition 3 Let $\mathcal{C} \subset \mathbb{R}^n$ be a closed convex set. The *projection map*, denoted by $\mathcal{P}_{\mathcal{C}} : \mathbb{R}^n \rightrightarrows \mathcal{C}$, is defined as follows $\mathcal{P}_{\mathcal{C}}(y) := \arg\min\{\|y - z\| : z \in \mathcal{C}\}$.

The next lemma presents an important property of the projection.

Proposition 4 *Let $y \in \mathbb{R}^n$ and $z \in \mathcal{C}$. Then, we have $\left\|\mathcal{P}_{\mathcal{C}}(y) - z\right\|^2 \leq \|y - z\|^2$.*

Definition 5 Let S be a nonempty subset of $\mathbb{R}^n$. A sequence $(v^k)_{k\in\mathbb{N}} \subset \mathbb{R}^n$ is said to be quasi-Fejér convergent to S, if and only if, for all $v \in S$ there exists $\bar{k} \geq 0$ and a summable sequence $(\epsilon_k)_{k\in\mathbb{N}}$, such that $\|v^{k+1} - v\|^2 \leq \|v^k - v\|^2 + \epsilon_k$ for all $k \geq \bar{k}$.

In the following lemma, we state the main properties of quasi-Fejér sequences that we will need; a comprehensive study on this topic can be found in [4].

Lemma 6 *Let $(v^k)_{k\in\mathbb{N}}$ be quasi-Fejér convergent to S. Then, the following conditions hold*:

(i) *the sequence $(v^k)_{k\in\mathbb{N}}$ is bounded*;
(ii) *if a cluster point $\bar{v}$ of $(v^k)_{k\in\mathbb{N}}$ belongs to S, then $(v^k)_{k\in\mathbb{N}}$ converges to $\bar{v}$.*

3 Subgradient method with non-monotone line search

We are interested in the following constrained optimization problem

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } x \in \mathcal{C}, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function and $\mathcal{C} \subset \mathbb{R}^n$ is a closed and convex set. *Denote by* Ω^* *the optimal set of the problem* (1) *and by* f^* *the optimal value*. Throughout the paper we will consider problem (1) under the following two assumptions:

(H1) $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function and $L_{f,\mathcal{C}}$-Lipschitz continuous;

(H2) $f^* := \inf_{x\in\mathcal{C}} f(x) > -\infty$.

We propose the following conceptual algorithm to find a solution of problem (1).

Algorithm 1. SubGrad projection method with non-monotone line search

Step 0. *Fix* $c > 0$, $(\gamma_k)_{k\in\mathbb{N}} \subset \mathbb{R}_{++}$ *a non-increasing sequence,* $\rho > 1/2$, $\beta \in (0,1)$ *and* $\alpha > 0$. *Choose an initial point* $x_1 \in C$. *Set* $\alpha_1 = \alpha$ *and* $k = 1$;

Step 1. *Choose* $s_k \in \partial f(x_k)$. *If* $s_k = 0$, *then STOP and returns* x_k;

Step 2. *Compute*

$$\ell_k := \min\left\{\ell \in \mathbb{N} : \ \beta^\ell \alpha_k \le c\beta\gamma_k, \ f(\mathcal{P}_\mathcal{C}(x_k - \beta^\ell \alpha_k s_k)) \le f(x_k) - \rho(\beta^\ell \alpha_k)\|s_k\|^2 + \gamma_k\right\}; \tag{2}$$

Step 3. *Set* $x_{k+1} := \mathcal{P}_\mathcal{C}(x_k - \beta^{\ell_k}\alpha_k s_k)$, $\alpha_{k+1} := \beta^{\ell_k - 1}\alpha_k$. *Update* $k \leftarrow k+1$ *and go to Step 1.*

Remark 2 It follows from [13, Theorem 4.2.3] that the set where convex functions fail to be differentiable is of zero measure. Consequently, almost every opposite direction of a subgradient is a descent direction. Therefore, we expect Algorithm 1 to be able to skip non-differentiability points that are not minimum points and then behave similarly to the gradient method with non-monotonic line search at differentiability points. It is worth to noting that the idea of using general non-monotone line searches in differentiable optimization, generalizing the non-monotone searches proposed in [12, 27], have appeared in [11, 24]. A modified version of the subgradient method with the non-monotone line search proposed in [12] was considered in [17–19], see also [14].

In the following lemmas we establish general inequalities that are important in our analysis. We begin presenting the well definition of ℓ_k defined in Step 2 of Algorithm 1 and two inequalities that follows as a consequence.

Lemma 7 *There exists* ℓ_k *satisfying* (2). *As a consequence, the following inequalities hold*:

$$\alpha_{k+1} \le c\gamma_k, \qquad f(x_{k+1}) \le f(x_k) - \rho\beta\alpha_{k+1}\|s_k\|^2 + \gamma_k, \quad \forall k \in \mathbb{N}, \tag{3}$$

and $x_{k+1} \in \mathcal{C}$, *for all* $k \in \mathbb{N}$.

Proof Since f and the projection $\mathcal{P}_\mathcal{C}$ are continuous functions and the point $x_k \in \mathcal{C}$, we have $\lim_{\alpha\to 0^+}(f(\mathcal{P}_\mathcal{C}(x_k - \alpha s_k)) - f(x_k) + \rho\alpha\|s_k\|^2) = 0$. Hence, due to $\gamma_k > 0$,

there exists $\eta_k > 0$ such that $f(\mathcal{P}_{\mathcal{C}}(x_k - \alpha s_k)) - f(x_k) + \rho\alpha\|s_k\|^2 < \gamma_k$, for all $\alpha \in (0, \eta_k]$, or equivalently,

$$f(\mathcal{P}_{\mathcal{C}}(x_k - \alpha s_k)) \leq f(x_k) - \rho\alpha\|s_k\|^2 + \gamma_k, \qquad \alpha \in (0, \eta_k]. \tag{4}$$

Hence, due to $\beta \in (0, 1)$ we have $\lim_{\ell \to 0^+} \beta^\ell \alpha_k = 0$, and since $\eta_k > 0$, we obtain that there exists $\ell_* \in \mathbb{N}$ such that $\ell \geq \ell_*$ implies $\beta^\ell \alpha_k \in (0, \eta_k]$. Therefore, due to (4) be hold for all $\alpha \in (0, \eta_k]$, there exists ℓ_k satisfying (2), which proves the first statement. The inequalities in (3) and inclusion $x_{k+1} \in \mathcal{C}$ follow from the definitions of x_{k+1} and α_{k+1} in Step 3. □

From now on $(x_k)_{k\in\mathbb{N}}$ *denotes the sequence generated by Algorithm* 1. In the next lemma we recall a classical inequality used in the study of subgradient methods, see for example [1, Lemma 8.11] and se also [5]. We give the proof here for the sake of completeness.

Lemma 8 *For any* $x \in \mathbb{R}^n$ *there holds*

$$2\beta\alpha_{k+1}(f(x_k) - f(x)) \leq \|x_k - x\|^2 - \|x_{k+1} - x\|^2 + \beta^2\alpha_{k+1}^2\|s_k\|^2, \quad \forall k \in \mathbb{N}. \tag{5}$$

In addition, if f *is a* σ*-strongly convex function then there holds*

$$2\beta\alpha_{k+1}(f(x_k) - f(x)) \leq (1 - \sigma\beta\alpha_{k+1})\|x_k - x\|^2 - \|x_{k+1} - x\|^2 + \beta^2\alpha_{k+1}^2\|s_k\|^2, \quad \forall k \in \mathbb{N}. \tag{6}$$

Proof Since the inequality (6) becomes (5) for $\sigma = 0$, it is sufficient to prove (6). It follows from the definition of x_{k+1} in Step 3 of Algorithm 1, Proposition 4 and also definition of α_{k+1} that

$$\begin{aligned} \|x_{k+1} - x\|^2 &= \|\mathcal{P}_{\mathcal{C}}(x_k - \beta^{\ell_k}\alpha_k s_k) - x\|^2 \\ &\leq \|x_k - \beta^{\ell_k}\alpha_k s_k - x\|^2. \\ &= \|x_k - x\|^2 + 2\beta\alpha_{k+1}\langle s_k, x - x_k\rangle + \beta^2\alpha_{k+1}^2\|s_k\|^2. \end{aligned} \tag{7}$$

Therefore, considering that f is a σ-strongly convex function, it follows from Proposition 1 that $\langle s_k, x - x_k\rangle \leq f(x) - f(x_k) - (\sigma/2)\|x_k - x\|^2$, which substituting into (7) yields (6). The inequality (5) follows from (6) by letting $\sigma = 0$. □

Next we present an important relationship between $(\alpha_k)_{k\in\mathbb{N}}$ and $(\gamma_k)_{k\in\mathbb{N}}$.

Lemma 9 *The following inequality holds*:

$$\alpha_k \geq \min\left\{\alpha_1, c\beta\gamma_k, \frac{\gamma_k}{(1+\rho)L_{f,\mathcal{C}}^2}\right\}, \quad \forall k \in \mathbb{N}. \tag{8}$$

Proof The inequality (8) immediately holds for $k = 1$. Suppose by an absurd that there exists $k \in \mathbb{N}$ such that

$$\alpha_{k+1} < \min\left\{\alpha_1, c\beta\gamma_{k+1}, \frac{\gamma_{k+1}}{(1+\rho)L_{f,\mathcal{C}}^2}\right\}. \tag{9}$$

Since we are supposing that $(\gamma_k)_{k\in\mathbb{N}}$ is a non-increasing sequence, using the definition of α_{k+1} in Step 3 of Algorithm 1 together with (9), we conclude that

$$\beta^{\ell_k-1}\alpha_k = \alpha_{k+1} < \min\left\{\alpha_1, c\beta\gamma_{k+1}, \frac{\gamma_{k+1}}{(1+\rho)L_{f,\mathcal{C}}^2}\right\} \leq \min\left\{c\beta\gamma_k, \frac{\gamma_k}{(1+\rho)L_{f,\mathcal{C}}^2}\right\}. \tag{10}$$

Thus, in particular, we have

$$\beta^{\ell_k-1}\alpha_k \leq c\beta\gamma_k. \tag{11}$$

Considering that f is $L_{f,\mathcal{C}}$-Lipschitz continuous and $x_k \in \mathcal{C}$, using Proposition 4 we have

$$\begin{aligned} f\big(\mathcal{P}_{\mathcal{C}}(x_k - \beta^{\ell_k-1}\alpha_k s_k)\big) - f(x_k) &\leq L_{f,\mathcal{C}}\|\mathcal{P}_{\mathcal{C}}(x_k - \beta^{\ell_k-1}\alpha_k s_k) - x_k\| \\ &\leq L_{f,\mathcal{C}}\|x_k - \beta^{\ell_k-1}\alpha_k s_k - x_k\| \\ &= L_{f,\mathcal{C}}\beta^{\ell_k-1}\alpha_k\|s_k\|. \end{aligned}$$

Using again the $L_{f,\mathcal{C}}$-Lipschitz continuity, it follows from Proposition 2 that $\|s_k\| \leq L_{f,\mathcal{C}}$. Thus, after some algebraic manipulations, the two previous inequalities imply that

$$\begin{aligned} f\big(\mathcal{P}_{\mathcal{C}}(x_k - \beta^{\ell_k-1}\alpha_k s_k)\big) - f(x_k) + \rho\beta^{\ell_k-1}\alpha_k\|s_k\|^2 &\leq L_{f,\mathcal{C}}\beta^{\ell_k-1}\alpha_k\|s_k\| + \rho\beta^{\ell_k-1}\alpha_k\|s_k\|^2 \\ &\leq \beta^{\ell_k-1}\alpha_k(1+\rho)L_{f,\mathcal{C}}^2. \end{aligned}$$

Hence, using (10) we obtain that $f(\mathcal{P}_{\mathcal{C}}(x_k - \beta^{\ell_k-1}\alpha_k s_k)) - f(x_k) + \rho\beta^{\ell_k-1}\alpha_k\|s_k\|^2 < \gamma_k$, or equivalently

$$f\big(\mathcal{P}_{\mathcal{C}}(x_k - \beta^{\ell_k-1}\alpha_k s_k)\big) < f(x_k) - \rho\beta^{\ell_k-1}\alpha_k\|s_k\|^2 + \gamma_k,$$

which, together with (11), contradicts the definition of ℓ_k in (2). Thus, (8) holds for all k and the proof is complete. □

Remark 3 It is worth to noting that the choice of α_1 is crucial for the performance of the method. However, we are not aware of any theoretically founded criterion for choosing the initial stepsize in the context of line search methods. In the cases in which $L_{f,\mathcal{C}}$ is known, a conservative choice is

$$\alpha_1 = \min\left\{c\beta\gamma_1, \frac{\gamma_1}{(1+\rho)L_{f,\mathcal{C}}^2}\right\}$$

since in this case it follows from Lemma 9 that the line search condition at iteration 1 of Algorithm 1 will be directly satisfied (with $\ell_1 = 0$).

In the following we combine the inequalities (3) in Lemma 7 with those in Lemmas 8 and 9 to provide an inequality that will allow us to prove the convergence of $(x_k)_{k\in\mathbb{N}}$ and obtain some iteration-complexity bounds. For that, it is convenient to define the following positive constants for $\rho > 1/2$:

$$\Theta := \min\left\{\frac{\alpha_1}{\gamma_1}, c\beta, \frac{1}{(1+\rho)L_{f,\mathcal{C}}^2}\right\}, \qquad \Gamma := \Theta\left(2\beta - \frac{\beta}{\rho}\right). \tag{12}$$

Lemma 10 *Assume that $\Omega^* \neq \emptyset$. Let $(x_k)_{k\in\mathbb{N}}$ be generated by Algorithm 1 and $x^* \in \Omega^*$.Then, following inequality holds*:

$$\Gamma\gamma_{k+1}(f(x_k) - f^*) \leq \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + \frac{1}{\rho}\beta c\gamma_k^2, \quad \forall k \in \mathbb{N}. \tag{13}$$

In addition, if f is a σ-strongly convex function then there holds

$$\Gamma\gamma_{k+1}(f(x_k) - f^*) \leq (1 - \sigma\beta\Theta\gamma_{k+1})\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + \frac{1}{\rho}\beta c\gamma_k^2, \quad \forall k \in \mathbb{N}. \tag{14}$$

Proof First of all, note that the inequality (14) becomes (13) for $\sigma = 0$. Then, it is sufficient to prove the inequality (14). It follows from Lemma 7 that $\beta\alpha_{k+1}\|s_k\|^2 \leq (f(x_k) - f(x_{k+1}) + \gamma_k)/\rho$, which combined with inequality (6) in Lemma 8 yields

$$\left(2\beta - \frac{\beta}{\rho}\right)\alpha_{k+1}(f(x_k) - f^*) \leq (1 - \sigma\beta\alpha_{k+1})\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + \frac{1}{\rho}\beta\alpha_{k+1}\gamma_k. \tag{15}$$

On the other hand, by using Lemma 9, considering that $(\gamma_k)_{k\in\mathbb{N}}$ is a non-increasing sequence and also using the first equality in (12) we obtain that

$$\alpha_{k+1} \geq \min\left\{\frac{\alpha_1}{\gamma_1}, c\beta, \frac{1}{(1+\rho)L_{f,\mathcal{C}}^2}\right\}\gamma_{k+1} = \Theta\gamma_{k+1}. \tag{16}$$

Besides, we know from Lemma 7 that $\alpha_{k+1} \leq c\gamma_k$, which combined with (15) and (16) yield

$$\left(2\beta - \frac{\beta}{\rho}\right)\Theta\gamma_{k+1}(f(x_k) - f^*) \leq (1 - \sigma\beta\Theta\gamma_{k+1})\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + \frac{1}{\rho}\beta c\gamma_k^2,$$

Therefore, taking into account (12), the last inequality implies (14) and the proof is concluded. □

Remark 4 It is worth to compare the classical inequalities (5) and (6) in Lemma 8 with, respectively, the inequalities (13) and (14) in Lemma 10. This comparison shows that the latter inequalities allow transfer to the sequence of non-monotonicity parameters $(\gamma_k)_{k\in\mathbb{N}}$ the classical conditions usually imposed on the sequence of step sizes $(\alpha_k)_{k\in\mathbb{N}}$ that control the behavior of $(x_k)_{k\in\mathbb{N}}$, see for example [1, 2]. This way,

the method itself will select the step sizes α_k, which are usually prefixed in the classical formulations of sugbgradient method. In fact, for each prefixed non-increasing exogenous sequence $(\gamma_k)_{k\in\mathbb{N}}$, it follows from Lemmas 7, 9 and first equality in (12) that Algorithm 1, by performing a non-monotone line search, select the step sizes α_k satisfying the following inequalities

$$\Theta\gamma_{k+1} \leq \alpha_{k+1} \leq c\gamma_k, \qquad k \in \mathbb{N}, \tag{17}$$

which shows that our method is different from the ones that appeared in [17–19]. Moreover, our line search allows different choices for the sequence $(\gamma_k)_{k\in\mathbb{N}}$ that controls the non-monotonicity.

3.1 Convergence analysis

In this section we analyze the behavior of the sequence $(x_k)_{k\in\mathbb{N}}$ under assumptions (H1), (H2) and more two additional assumptions. *The additional assumptions will be used separately and only when explicitly stated*. The new assumptions are as follows:

(H3) The sequence of non-monotonicity parameters $(\gamma_k)_{k\in\mathbb{N}}$ satisfies

$$\lim_{N\to+\infty} \frac{\sum_{k=1}^N \gamma_k^2}{\sum_{k=1}^N \gamma_{k+1}} = 0.$$

(H4) The sequence of non-monotonicity parameters $(\gamma_k)_{k\in\mathbb{N}}$ satisfies

$$\lim_{N\to+\infty} \frac{\sum_{k=1}^N \gamma_k^2}{N\gamma_{N+1}} = 0.$$

Theorem 11 *Assume that $\Omega^* \neq \emptyset$. Let $(x_k)_{k\in\mathbb{N}}$ be generated by Algorithm 1 with $\rho > 1/2$ and $x^* \in \Omega^*$. Then, for each fixed $N \in \mathbb{N}$, the following inequality hold*:

$$\min\left\{f(x_k) - f^* \,:\, k = 1,\dots,N\right\} \leq \frac{1}{\Gamma}\left(\|x_1 - x^*\|^2 + \beta\rho^{-1}c\sum_{k=1}^N \gamma_k^2\right)\frac{1}{\sum_{k=1}^N \gamma_{k+1}}. \tag{18}$$

Consequently, if (H3) holds then $\lim_{N\to+\infty} \min\left\{f(x_k) - f^* \,:\, k = 1,\dots,N\right\} = 0$.

Proof Let $k \leq N$. Using the inequality (13) in Lemma 10 and taking into account that $\min\left\{f(x_k) - f^* \,:\, k = 1,\dots,N\right\}\sum_{k=1}^N \gamma_{k+1} \leq \sum_{k=1}^N \gamma_{k+1}(f(x_k) - f^*)$, we obtain that

$$\Gamma \min\left\{f(x_k) - f^* \,:\, k = 1,\dots,N\right\}\sum_{k=1}^N \gamma_{k+1} \leq \|x_1 - x^*\|^2 + \frac{1}{\rho}\beta c\sum_{k=1}^N \gamma_k^2,$$

which implies (18). For concluding the proof, first note that assumption (H3) implies that $\lim_{N\to+\infty} \sum_{k=1}^{N} \gamma_{k+1} = +\infty$. Thus, using (18), the last statement follows. □

Let us state and prove a special instance of Theorem 11. For that we need a result, which can be found in [1, Lemma 8.27].

Lemma 12 *Let $a > 0$, $d \geq 0$ and $N \geq 1$. Then,*

$$\frac{d + a\sum_{k=1}^{N} \frac{1}{k}}{\sum_{k=1}^{N} \frac{1}{\sqrt{k+1}}} \leq \frac{4(d + a + a\ln(N))}{\sqrt{N}}.$$

Remark 5 If $(\gamma_k)_{k\in\mathbb{N}}$ satisfies (H4), then $(\gamma_k)_{k\in\mathbb{N}}$ also satisfies (H3). The sequence $(\gamma_k)_{k\in\mathbb{N}}$ with $\gamma_k = 1/(k^{1-\theta/2})$ and $\theta \in (0, 1)$ satisfies (H4). Using Lemma 12 we can also prove that sequence $(\gamma_k)_{k\in\mathbb{N}}$ with $\gamma_k = 1/\sqrt{k}$ satisfies (H3).

The proof of the next theorem follows by combining inequality (18) of Theorem 11 with Lemma (12).

Theorem 13 *Let $x^* \in \Omega^*$, $(x_k)_{k\in\mathbb{N}}$ be generated by Algorithm 1 with $\rho > 1/2$ and*

$$\gamma_k = \frac{1}{\sqrt{k}}, \qquad \forall k \in \mathbb{N}.$$

Then, for each fixed $N \in \mathbb{N}$, the following inequality hold:

$$\min\left\{f(x_k) - f^* \ : \ k = 1, \dots, N\right\} \leq \frac{4}{\Gamma} \frac{\|x_1 - x^*\|^2 + \beta\rho^{-1}c + \beta\rho^{-1}c\ln(N)}{\sqrt{N}}.$$

Consequently, $\lim_{N\to+\infty} \min\left\{f(x_k) - f^* \ : \ k = 1, \dots, N\right\} = 0$.

Remark 6 Since Lemma 12 implies that sequence $(\gamma_k)_{k\in\mathbb{N}}$ with $\gamma_k = 1/\sqrt{k}$ satisfies (H3), we conclude that the inequality in Theorem 13 is an explicit rate of convergence, namely, $\mathcal{O}(\ln(N)/\sqrt{N})$.

We end this section by showing that $(x_k)_{k\in\mathbb{N}}$ generated by Algorithm 1 converges to a solution of the problem (1) whenever $\Omega^* \neq \emptyset$. To this end, we assume that the sequence $(\gamma_k)_{k\in\mathbb{N}}$ satisfies the following conditions:

(H5) $\sum_{k=1}^{+\infty} \gamma_k^2 \leq +\infty$;
(H6) $\sum_{k=1}^{+\infty} \gamma_k = +\infty$.

Remark 7 If $(\gamma_k)_{k\in\mathbb{N}}$ satisfies (H5) and (H6), then $(\gamma_k)_{k\in\mathbb{N}}$ also satisfies (H3). The sequence $(\gamma_k)_{k\in\mathbb{N}}$ with $\gamma_k = 1/k$ satisfies (H5) and (H6).

Theorem 14 *Let $(x_k)_{k\in\mathbb{N}}$ be generated by Algorithm 1 with $\rho > 1/2$. Assume that (H5) holds. If $\Omega^* \neq \emptyset$, then $(x_k)_{k\in\mathbb{N}}$ is bounded. Moreover, if (H6) hold, then $(x_k)_{k\in\mathbb{N}}$ converges to a solution of problem (1).*

Proof Let $x \in \Omega^*$. Using (13) in Lemma 10 we obtain after some algebraic manipulations that

$$\|x_{k+1} - x\|^2 \leq \|x_k - x\|^2 - \Gamma\gamma_{k+1}(f(x_k) - f^*) + \frac{1}{\rho}\beta c\gamma_k^2, \quad \forall k \in \mathbb{N}.$$

Thus, considering that $f(x_k) - f^* \geq 0$, for all $k \in \mathbb{N}$, it follows from the last inequality that

$$\|x_{k+1} - x\|^2 \leq \|x_k - x\|^2 + \frac{1}{\rho}\beta c\gamma_k^2, \quad \forall k \in \mathbb{N}.$$

Hence, (H5) together with Definition 5 implies that the sequence $(x_k)_{k\in\mathbb{N}}$ is quasi-Fejér convergent to Ω^*. Since $\Omega^* \neq \emptyset$, the item (i) of Lemma 6 implies that $(x_k)_{k\in\mathbb{N}}$ is bounded and the first statement is proved. To proceed, define a subsequence $(x_{k_N})_{N\in\mathbb{N}}$ of the sequence $(x_k)_{k\in\mathbb{N}}$ such that

$$f(x_{k_N}) - f^* := \min\left\{f(x_k) - f^* \,:\, k = 1, \ldots, N\right\}, \qquad N \in \mathbb{N}.$$

Since $(x_k)_{k\in\mathbb{N}}$ is bounded, we conclude that $(x_{k_N})_{N\in\mathbb{N}}$ is also bounded. Without loss of generality we can assume that $(x_{k_N})_{N\in\mathbb{N}}$ converges. Set $\bar{x} = \lim_{N\to\infty} x_{k_N}$. Under the assumptions (H3) and (H6) we have from the last part of Theorem 11 that $0 = \lim_{N\to+\infty}(f(x_{k_N}) - f^*)$. Thus, using that $\bar{x} = \lim_{N\to\infty} x_{k_N}$, we conclude that $f(\bar{x}) = f^*$, which implies that $\bar{x} \in \Omega^*$. Therefore, due $(x_k)_{k\in\mathbb{N}}$ be quasi-Fejér convergent to Ω^*, by applying item (ii) of Lemma 6 we obtain the $(x_k)_{k\in\mathbb{N}}$ converges to $\bar{x}$, which completes the proof. □

3.2 Convergence analysis for compact constraint set

The aim of this section is to analyze the behavior of $(x_k)_{k\in\mathbb{N}}$ under assumptions (H1), (H2) (H3) and one new additional assumption. The new assumption is as follows:

(H7) The set $\mathcal{C}$ is compact.

To state the next theorem let us introduce the following auxiliary positive constant

$$D \geq \max_{x,y\in\mathcal{C}} \|x - y\|^2,$$

and to prove it we also need an additional result, which can be found in [1, Lemma 8.27].

Lemma 15 *Let $a > 0$, $d \geq 0$ and $N \geq 2$. Then,*

$$\frac{d + a\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{k}}{\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{\sqrt{k+1}}} \leq \frac{4(d + a\ln(3))}{\sqrt{N+2}}.$$

In the next theorem we show that for suitable choice of the sequence $(\gamma_k)_{k\in\mathbb{N}}$ the rate of convergence of Algorithm 1 is $\mathcal{O}(1/\sqrt{k})$.

Theorem 16 *Let $x^* \in \Omega^*$, $(x_k)_{k\in\mathbb{N}}$ be generated by Algorithm 1 with $\rho > 1/2$ and*

$$\gamma_k = \frac{1}{\sqrt{k}}, \qquad \forall k \in \mathbb{N}. \tag{19}$$

Then, for each fixed $N \in \mathbb{N}$ with $N \geq 2$, the following inequality hold:

$$\min\left\{f(x_k) - f^* \,:\, k = 1, \dots, N\right\} \leq \frac{4\left(D + \frac{\beta c}{\rho}\ln(3)\right)}{\Gamma\sqrt{N+2}}.$$

Consequently, $\lim_{N\to+\infty} \min\left\{f(x_k) - f^* \,:\, k = 1, \dots, N\right\} = 0$.

Proof It follows from (10) in Lemma 13 and definition of γ_k in (19) that

$$\Gamma\frac{1}{\sqrt{k+1}}(f(x_k) - f^*) \leq \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + \frac{\beta c}{\rho}\frac{1}{k}, \quad \forall k \in \mathbb{N}.$$

Thus, summing this inequality over $k = \lceil N/2\rceil, \lceil N/2\rceil + 1, \dots, N$ we conclude that

$$\Gamma\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{\sqrt{k+1}}(f(x_k) - f^*) \leq \|x_{\lceil N/2\rceil} - x^*\|^2 - \|x_{N+1} - x^*\|^2 + \frac{\beta c}{\rho}\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{k}.$$

Since $\min\left\{f(x_k) - f^* \,:\, k = 1, \dots, N\right\}\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{\sqrt{k+1}} \leq \sum_{k=\lceil N/2\rceil}^{N} \frac{1}{\sqrt{k+1}}(f(x_k) - f^*)$ and considering that $D \geq \max_{x,y\in\mathcal{C}} \|x - y\|^2$, we obtain

$$\Gamma\min\left\{f(x_k) - f^* \,:\, k = 1, \dots, N\right\}\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{\sqrt{k+1}} \leq D + \frac{\beta c}{\rho}\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{k}.$$

The last inequality implies that

$$\min\left\{f(x_k) - f^* \,:\, k = 1, \dots, N\right\} \leq \frac{D + \frac{\beta c}{\rho}\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{k}}{\Gamma\sum_{k=\lceil N/2\rceil}^{N} \frac{1}{\sqrt{k+1}}},$$

which combined with Lemma 15 yields the desired inequality. The second statement of theorem is an immediate consequence of the first one. □

Theorem 17 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a σ-strongly convex function and $\sigma > 0$. Let $(x_k)_{k\in\mathbb{N}}$ be generated by Algorithm 1 with $\rho > 1/2$,*

$$\gamma_k = \frac{2}{\sigma\beta\Theta k}, \qquad \forall k \in \mathbb{N}.$$

and $x^* \in \Omega^*$. *Then, for each fixed* $N \in \mathbb{N}$, *the following inequality holds*:

$$\min\left\{f(x_k) - f(x^*) : \ k = 1, \dots, N\right\} \leq \frac{8\beta c}{\rho\sigma\beta\Theta\Gamma}\frac{1}{(N+1)}.$$

As a consequence, $\lim_{N\to+\infty} \min\left\{f(x_k) - f(x^*) : \ k = 1, \dots, N\right\} = 0$.

Proof Since $\gamma_k = 2/(\sigma\beta\Theta k)$ satisfies (H3), it follows from (14) in Lemma 10 and $\sigma > 0$ that

$$\frac{\Gamma}{\sigma\beta\Theta}(f(x_k) - f(x^*)) \leq \frac{1 - \sigma\beta\Theta\gamma_{k+1}}{\sigma\beta\Theta\gamma_{k+1}}\|x_k - x^*\|^2 - \frac{1}{\sigma\beta\Theta\gamma_{k+1}}\|x_{k+1} - x^*\|^2 + \frac{\beta c}{\rho\sigma\beta\Theta}\frac{\gamma_k^2}{\gamma_{k+1}}.$$

Taking into account that $\gamma_{k+1} = 2/(\sigma\beta\Theta(k+1))$, the last last inequality becomes

$$\frac{\Gamma}{\sigma\beta\Theta}(f(x_k) - f(x^*)) \leq \frac{k-1}{2}\|x_k - x^*\|^2 - \frac{k+1}{2}\|x_{k+1} - x^*\|^2 + \frac{2\beta c}{\rho(\sigma\beta\Theta)^2}\frac{2}{k}.$$

Hence, multiplying the last inequality by $2k$ we obtain that

$$\frac{2\Gamma}{\sigma\beta\Theta}k(f(x_k) - f(x^*)) \leq k(k-1)\|x_k - x^*\|^2 - k(k+1)\|x_{k+1} - x^*\|^2 + \frac{8\beta c}{\rho(\sigma\beta\Theta)^2}.$$

Thus, due to $\min\left\{f(x_k) - f(x^*) : \ k = 1, \dots, N\right\}\sum_{k=1}^N k \leq \sum_{k=1}^N k(f(x_k) - f(x^*))$, we have

$$\frac{2\Gamma}{\sigma\beta\Theta}\min\left\{f(x_k) - f(x^*) : \ k = 1, \dots, N\right\}\sum_{k=1}^N k \leq -N(N+1)\|x_{N+1} - x^*\|^2 + \frac{8\beta c}{\rho(\sigma\beta\Theta)^2}N.$$

Therefore, due to $\sum_{k=1}^N k = N(N+1)/2$, we conclude that

$$\Gamma\min\left\{f(x_k) - f(x^*) : \ k = 1, \dots, N\right\} \leq \frac{8\beta c}{\rho\sigma\beta\Theta}\frac{1}{(N+1)},$$

which is equivalent the desired inequality. The second statement of theorem is an immediate consequence of the first one. □

Remark 8 In Theorems 16 and 17 we provide two different choices for γ_k. The one given in Theorem 16 is simpler than in Theorem 17. On the hand, if we know the Lipschitz constant of f, then we can use γ_k given in Theorem 17 to improve the convergence rate given in Theorems 16.

Table 1 Comparison methods: step sizes in the subgradient method

Abbr.	Subgradient method	Step size
Constant step	Constant step size	$\alpha_k = 0.1$
Fixed length	Fixed step length	$\alpha_k = \frac{0.2}{\|\|g^k\|\|}$
Nonsum	Non-summable diminishing step	$\alpha_k = \frac{0.1}{\sqrt{k}}$
Sqrsum nonsum	Square summable but not summable step	$\alpha_k = \frac{0.5}{k}$

4 Illustrative numerical experiments

In this section we present some examples to illustrate the efficiency of the proposed method comparing its performance with other subgradient methods using classical step size rules.[1] It is not our intention to compete with these classical methods or other problem-specific algorithms, but rather to show that a general approach using our method performs remarkably well in a variety of settings. To this end, we consider the same set of constants in all methods and instances. More precisely, we perform Algorithm 1 (subgradient method with non-monotone line search) with $c = 1$, $\beta = 0.9$, $\rho = 0.8$, $\alpha_1 = 0.1$ and $\gamma_k = \frac{\zeta}{\sqrt{k}}$ in all numerical experiments, for all $k \geq 1$ and for some values of ζ (which will be presented in the Sects. 4.1, 4.2 and 4.3). The other four subgradient methods use different step sizes α_k described in Table 1. All the methods start from the same initial point "`zeros(n,1)`" which means the zero vector in $\mathbb{R}^n$

In each case, simple modifications could be made to to improve the performance of our method, but these examples serve to illustrate an implementation of the proposed method and highlight several features. All numerical experiments are implemented in MATLAB R2020b and executed on a personal laptop (Intel Core i7, 2.30 GHz, 8 GB of RAM).

4.1 Maximum of a finite collections of linear functions

The experiments of this section are generated by the class of functions which are point wise maximum of a finite collections of linear functions. These functions are defined as follows:

$$f(x) = \max\{f_j(x) = {a_j}^\top x + b_j \ : \ j = 1, \ldots, m\}, \tag{20}$$

where $a_j \in \mathbb{R}^n$ and $b_j \in \mathbb{R}$. In this case, $\partial f(x) = \text{conv}\ \{\partial f_i(x) \ : \ f_i(x) = f(x)\}$. In this example, we consider the vectors $a_j = (a_{j,1}, \ldots, a_{j,n}) \in \mathbb{R}^n$ and $b_j \in \mathbb{R}$ randomly

[1] An extensive numerical comparison between the proposed method and other nonmonotone subgradient methods is beyond the scope of the present paper and will be left for a future work. The aim of our numerical experiments is just to illustrate the proposed method and its properties.

chosen by "randn", a build-in MATLAB function which returns normally distributed random numbers.

As mentioned before, all the methods start from the same initial point and they stop if the iterate $k = 3000$ is attained. We compare the performance of the methods for different dimensions $n = 2$, $n = 5$, $n = 10$, $n = 20$, $n = 50$ and $n = 100$, where in Algorithm 1 we consider the values of $c = 1$, $\beta = 0.9$, $\rho = 0.8$, $\alpha_1 = 0.1$ and $\gamma_k = \frac{\zeta}{\sqrt{k}}$ with ζ as 0.01, 0.5, 1.0, 0.95, 1.5 and 3.3, respectively. The comparison of the methods is done in terms of the difference $f_{best} - f_{min}$, where the value f_{best} stands to the best value of $f(x^k)$ attained and f_{min} denotes the solution of the problem computed by CVX, a package for specifying and solving convex programming; see [9, 10].

The computation results are displayed in Figs. 1 and 2 and Tables 2 and 3. In these tables, the first column denotes the dimension n and the number of functions f_j, $j = 1, \ldots, m$, in (20). The other columns represent, for each method, the best value obtained for $f_{best} - f_{min}$ and the respective iterate it_{best} where it was attained. As we can see, the results show that Algorithm 1 outperforms the other methods providing a better solution or a similar solution in less iterates in all the test problems. In some instances, the subgradient method with the step sizes "constant step", "fixed length" and "sqrsum nonsum" fail to find an acceptable solution in the sense that these methods stop to decrease the objective function in few iterates. In this sense, Algorithm 1 and the subgradient method with the step size "nonsum" have a better performance than the previous ones.

We also investigate the behavior of the sequences $\{\alpha_k\}$ and $\{\gamma_k\}$ in terms of the inequality 17, i.e.,

$$\Theta\gamma_{k+1} \leq \alpha_{k+1} \leq c\gamma_k, \qquad k \in \mathbb{N},$$

where $\Theta = \min\left\{1, \frac{1}{(1+\rho)L_f^2}\right\}$. In this example, we consider the Lipschitz constant L_f of the function f in (20) as $L_f = \max\{||a_j|| \ : \ j = 1, \ldots, m\}$. The results are reported in Figs. 3 and 4 illustrating the theoretical result stated in Remark 4.

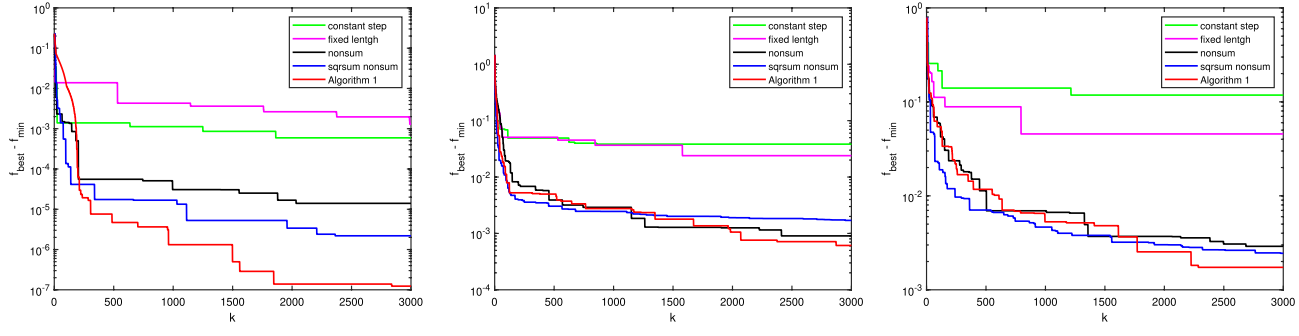
4.2 Fermat–Weber location problem

The experiment of this section is the well known Fermat-Weber location problem; see for instance Brimberg [3]. Let $a_1, \ldots, a_m$ be given points in $\mathbb{R}^n$. The Fermat-Weber location problem is to solve the following minimization problem

$$\min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^{m} w_i ||x - a_i||.$$

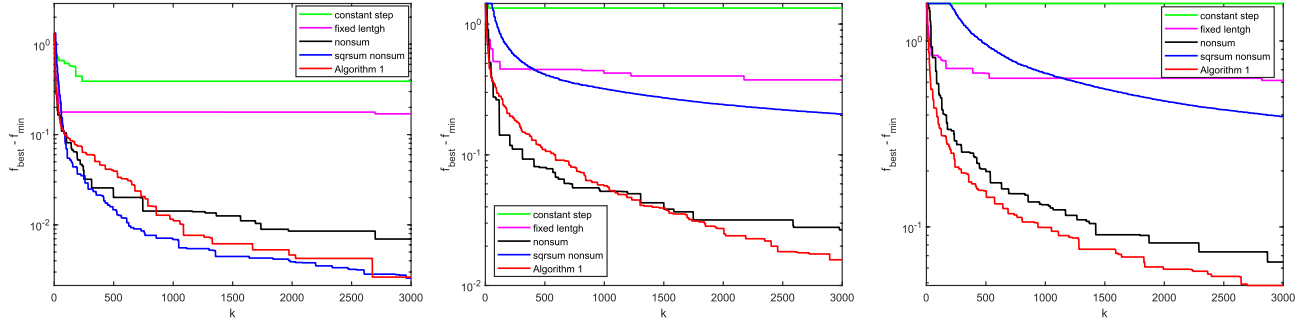
In our particular application, we consider the data points a_i, for $i = 1, \ldots, 27$, given by the coordinate[2] of the cities which are capital of all 26 states of Brazil and

[2] The latitude/longitude coordinates of the Brazilian cities can be found, for instance, at ftp://geoftp.ibge.gov.br/Organizacao/Localidades.



(a) $n = 2$ and $m = 10$ **(b)** $n = 5$ and $m = 30$ **(c)** $n = 10$ and $m = 50$

Fig. 1 Best value of $f(x^k) - f_{min}$ (using log. scale) for Algorithm 1 and each step size in Table 1



(a) $n = 20$ and $m = 100$ **(b)** $n = 50$ and $m = 150$ **(c)** $n = 100$ and $m = 500$

Fig. 2 Best value of $f(x^k) - f_{min}$ (using log. scale) for Algorithm 1 and each step size in Table 1

Table 2 Iteration where each algorithm attains the best value of $f(x^k) - f_{min}$

	Algorithm 1		Constant step		Fixed length	
	$f_{best} - f_{min}$	it_{best}	$f_{best} - f_{min}$	it_{best}	$f_{best} - f_{min}$	it_{best}
$n = 2, m = 10$	1.23289e−07	2838	5.9329e−04	1863	1.29047e−03	2990
$n = 5, m = 30$	6.11231e−04	2872	0.038322	858	0.0239323	1579
$n = 10, m = 50$	1.73369e−03	2286	0.118048	1215	0.0455682	795
$n = 20, m = 100$	2.63594e−03	2675	0.389903	235	0.169888	2698
$n = 50, m = 150$	0.0157351	2895	1.32626	25	0.37411	2175
$n = 100, m = 500$	0.0483826	2696	1.59047	2	0.613463	2820

Table 3 Iteration where each algorithm attains the best value of $f(x^k) - f_{min}$

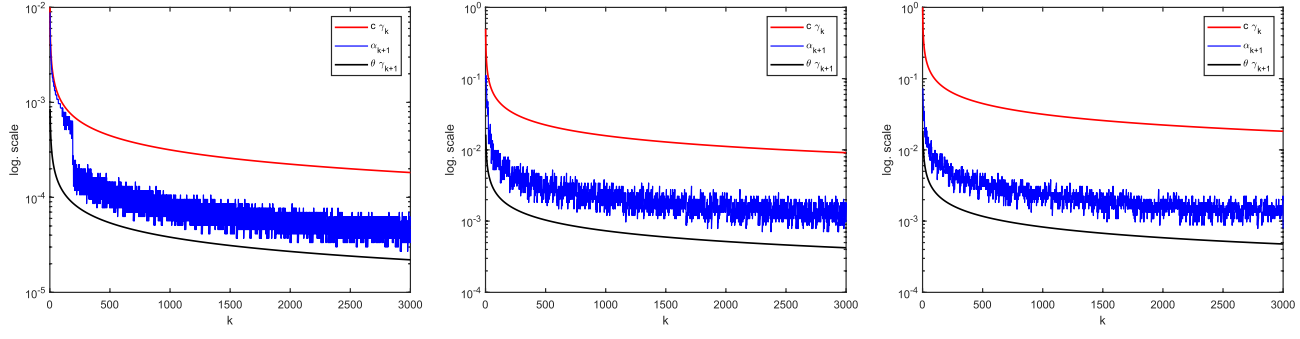
	Nonsum		Sqrsum nonsum	
	$f_{best} - f_{min}$	it_{best}	$f_{best} - f_{min}$	it_{best}
$n = 2, m = 10$	1.4022e−05	2034	2.1927e−06	2363
$n = 5, m = 30$	9.0404e−04	2412	1.7e−03	2949
$n = 10, m = 50$	2.90217e−03	2686	2.4701e−03	2765
$n = 20, m = 100$	6.96234e−03	2699	2.57007e−03	2959
$n = 50, m = 150$	0.0266287	2978	0.205205	2964
$n = 100, m = 500$	0.0646978	2868	0.39232	2972

Brasília (the Federal District, capital of Brazil). We take equally weights for all a_i, namely, $w_i = 1$, $i = 1, \ldots, 27$, and consider the integer part of the coordinates converting it from positive to negative to match with the real data. Our goal is to find a point that minimizes the sum of the distances to the given points representing the cities in order to see how distance is such a point from Brasília (the capital of Brazil). We denote by

$$f_{min} = 312.9232964118977, \qquad x_{min} = (-45.9630806884547, -12.7465709013343) \tag{21}$$

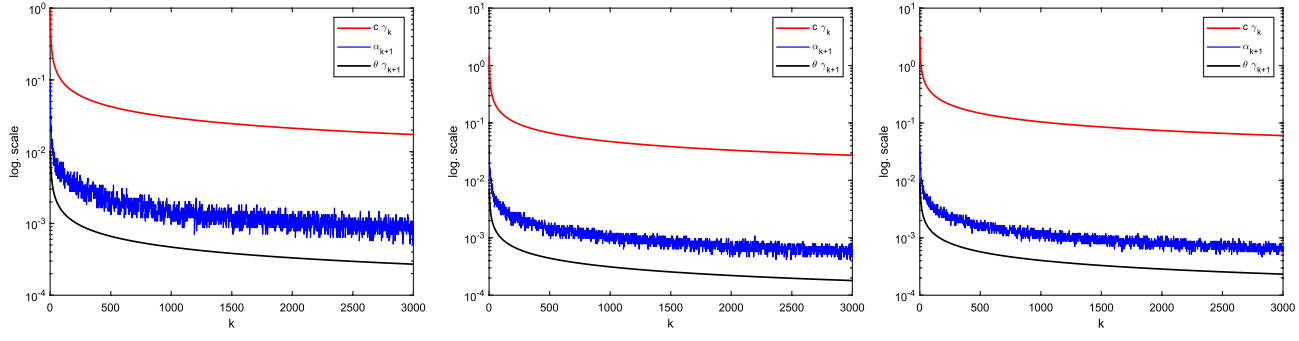
the solution found by the MATLAB package CVX; see [9, 10]. As mentioned in the beginning of this section, we perform Algorithm 1 and other four subgradient methods each of them with different step sizes α_k as described in Table 1. All the methods start from the same initial point and they stop at $k = 200$ iterates. As in the previous example, in Algorithm 1, we take $c = 1$, $\beta = 0.9$, $\rho = 0.8$, $\alpha_1 = 0.1$ and $\gamma_k = \frac{\zeta}{\sqrt{k}}$ with $\zeta = 2$.

In Table 4, we present in the first two columns the solution $x^* = (x_1^*, x_2^*)$ found by each method, in the third column the best value to $|f_{best} - f_{min}|$, where f_{best} stands to the best value of the objective function for each method and f_{min} is given by (21). The last column shows the iterate it_{best} in which the best value $|f_{best} - f_{min}|$ was attained. As we can notice, Algorithm 1 and the subgradient method with constant step size found a better solution compared to the solution known (f_{min} and x_{min}) than the other methods. However, Algorithm 1 found its best value in 29 iterates while



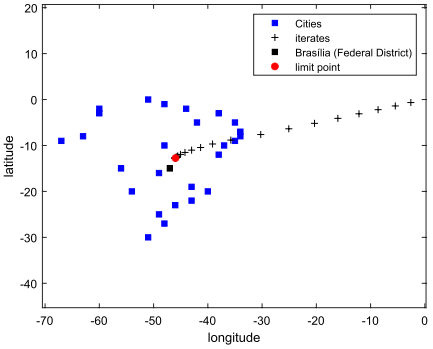
(a) $n = 2$ and $m = 10$ **(b)** $n = 5$ and $m = 30$ **(c)** $n = 10$ and $m = 50$

Fig. 3 Behavior of the sequences $\{\alpha_k\}$ and $\{\gamma_k\}$ (using log. scale) for Algorithm 1

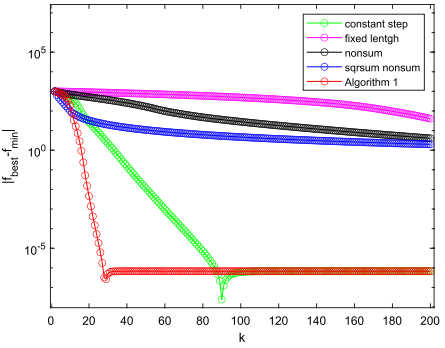


(a) $n = 20$ and $m = 100$ **(b)** $n = 50$ and $m = 150$ **(c)** $n = 100$ and $m = 500$

Fig. 4 Behavior of the sequences $\{\alpha_k\}$ and $\{\gamma_k\}$ (using log. scale) for Algorithm 1



(a) Running Algorithm 1.



(b) Performance of the methods (in log. scale).

Fig. 5 Subgradient methods for solving the Fermat-Weber location problem

Table 4 Solution found for the Fermat-Weber location problem

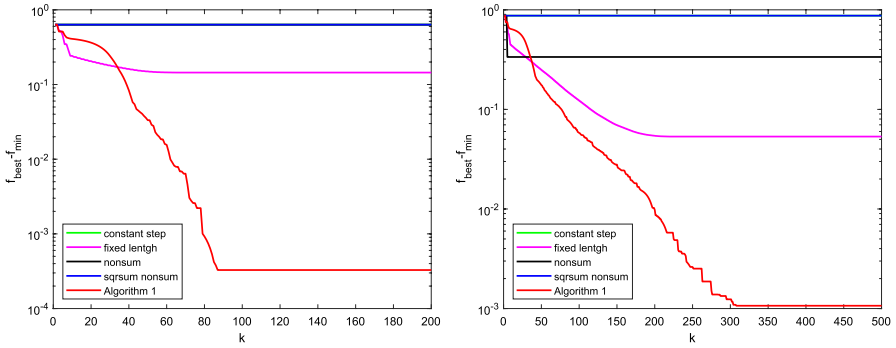
	x_1^*	x_2^*	$\lvert f_{best} - f_{min}\rvert$	it_{best}
Algorithm 1	− 45.963064141347097	− 12.746621089909885	2.66879e−07	29
Constant step	− 45.963064140711523	− 12.746621088320897	2.42824e−08	90
Fixed length	− 38.605444422335090	− 9.623064720309808	40.7379	200
Nonsum	− 43.842367512948982	− 11.429938434104701	4.02647	200
Sqrsum nonsum	− 44.521197252917077	− 11.740733447040283	1.9869	200

the subgradient method with constant step size takes 90 iterates to attain its best value. The performance of each method is presented in Fig. 5b showing the efficiency of the Algorithm 1 for this example. In Fig. 5a, we present the data of this example as well as the iterates of the Algorithm 1 and the solution found by the method.

4.3 Support vector machine

In this section, we apply Algorithm 1 and other four subgradient methods using different step sizes α_k as described in Table 1 for solving a binary classification problem. Support Vector Machine (SVM) is an effective and popular classification learning tool. In its native form, it is an unconstrained empirical loss minimization with a penalty term for the norm of the classifier that is being learned. However, the task of learning a support vector machine is modeled as a constrained quadratic problem formally stated as follows:

$$\min_{w,b} \Phi(w) = \frac{\lambda}{2}||w||^2 \quad \text{subject to} \quad y_i(w^\top x_i + b) \geq 1,\ i = 1, \ldots, m, \tag{22}$$



(a) $\lambda = 0.1$ **(b)** $\lambda = 0.01$

Fig. 6 Subgradient methods for solving the binary classification problem

where $S = \{(x_i, y_i)\}_{i=1}^m$ is a given training set with input features $x_i \in \mathbb{R}^n$, its respective labels $y_i \in \{-1, 1\}$, a regularization parameter $\lambda > 0$ and a bias term b. For simplicity, we omit the bias and consider only linear kernel model throughout our experiments; see more details in [6].

We consider the following optimization problem for learning with a SVM introduced in [25]

$$\min_{w \in \mathcal{C}} f(w) = \frac{\lambda}{2}||w||^2 + \frac{1}{m} \sum_{(x,y) \in S} \mathcal{L}(w;(x,y)), \tag{23}$$

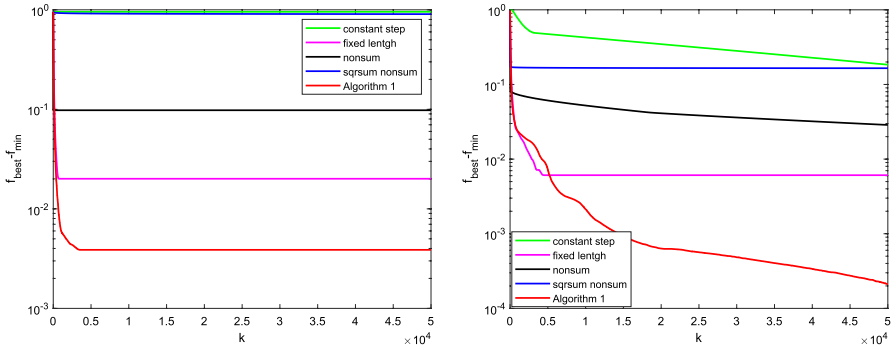
where $\mathcal{C} = \{x \in \mathbb{R}^n \ : \ ||x|| \leq \frac{1}{\sqrt{\lambda}}\}$ and $\mathcal{L}(w;(x,y)) = \max\{0, 1 - yw^\top x\}$ is the hinge loss function. Therefore, the subdifferential of $\mathcal{L}(\cdot\,;(x,y))$ at w is given by

$$\partial \mathcal{L}(w;(x_i,y_i)) = \begin{cases} -y_i x_i, & \text{if} \quad y_i w^\top x_i > 1 \\ 0, & \text{otherwise.} \end{cases}$$

We consider the Iris flower data set (or Ficher's Iris data set).[3] This data set is one of the best known data set found in pattern recognition literature. The data set consists of 3 classes, 50 instances and 4 numeric attributes (sepal length, sepal width, petal length and petal width) where each class refers to a type of Iris plant namely *Iris setosa*, *Iris versicolor* and *Iris verginica*. The first class is linearly separable from others while that latter are not linearly separable. We use the following code to generate our data test.

As in the previous sections, in Algorithm 1 we take $c = 1$, $\beta = 0.9$, $\rho = 0.8$, $\alpha_1 = 0.1$ and $\gamma_k = \frac{\zeta}{\sqrt{k}}$ with $\zeta = 10$. We run all the methods starting from the same initial point "`zeros(n,1)`" and they stop if the iterate $k = 50000$ is attained. As mentioned in [25] and references therein, for different data sets usually small

[3] This data set can be found at http://archive.ics.uci.edu/ml.



(a) $\lambda = 0.001$ **(b)** $\lambda = 0.0001$

Fig. 7 Subgradient methods for solving the binary classification problem

Table 5 Iteration where each algorithm attains the best value of $f(x^k) - f_{min}$

	Algorithm 1		Constant step		Fixed length	
	$f_{best} - f_{min}$	it_{best}	$f_{best} - f_{min}$	it_{best}	$f_{best} - f_{min}$	it_{best}
$\lambda = 0.1$	3.279e−04	87	0.63306518	1	0.1445231	67
$\lambda = 0.01$	1.0672e−03	308	0.87381057	1	0.05335602	221
$\lambda = 0.001$	3.8742e−03	3472	0.95494822	1	0.0200689	751
$\lambda = 0.0001$	2.1166e−04	49,892	0.18407921	50,000	6.08677e−03	4604

Table 6 Iteration where each algorithm attains the best value of $f(x^k) - f_{min}$

	Nonsum		Sqrsum nonsum	
	$f_{best} - f_{min}$	it_{best}	$f_{best} - f_{min}$	it_{best}
$\lambda = 0.1$	0.63306518	1	0.63306518	1
$\lambda = 0.01$	0.33602865	5	0.87381057	1
$\lambda = 0.001$	0.09793562	5	0.90521288	49,998
$\lambda = 0.0001$	0.02867965	49,991	0.16502491	49,998

values of λ are considered, and hence, we run the methods for different values of λ in (22) and (23) (the same value in both problems). As in the previous examples, the comparison of the methods is done in terms of the difference $f_{best} - f_{min}$, where the value f_{best} stands to the best value of $f(x^k)$ attained and f_{min} denotes the solution of the problem (22) computed by CVX; see [9, 10].

The results are presented in Tables 5 and 6. In these tables, the first column denotes the value of λ in (22) and (23). The other columns represent, for each method, the best value obtained for $f_{best} - f_{min}$ and the respective iterate it_{best} where it was attained. The values $f_{best} - f_{min}$ throughout the sequence are shown in Figs. 6

and 7. As we can see, the results show that Algorithm 1 outperforms the other methods providing a better solution in all the test problems.

5 Conclusions

In this paper we have presented a subgradient method with a non-monotone line search for the minimization of convex functions with simple convex constraints. The non-monotone line search allows the method to adaptively select step sizes. As preliminary numerical tests show, this method performs better than the standard subgradient method with prefixed step sizes, which we hope to motivate further research on this subject.

Acknowledgements We would like to thank the referees for their constructive remarks which allow us to improve our work.O. P. Ferreira was partially supported in part by CNPq - Brazil Grants 304666/2021-1, G. N. Grapiglia was partially supported by CNPq - Brazil Grant 312777/2020-5, J.C.O. Souza was supported in part by CNPq Grant 313901/2020-1. The project leading to this publication has received funding from the French government under the "France 2030" investment plan managed by the French National Research Agency (reference: ANR-17-EURE-0020) and from Excellence Initiative of Aix-Marseille University - A*MIDEX.

Data availibility The data that supports the findings of this study is available from the corresponding author upon request.

Code availability The code that supports the findings of this study is available from the corresponding author upon request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Beck, A.: First-Order Methods in Optmization, 1st edn. Society for Industrial and Applied Mathematics-SIAM and Mathematical Optimization Society (2017)
2. Bertsekas, D.P.: Nonlinear Programming, Athena Scientific Optimization and Computation Series, 2nd edn. Athena Scientific, Belmont (1999)
3. Brimberg, J.: The Fermat-Weber location problem revisited. Math. Program. **71**, 71–76 (1995)
4. Combettes, P.L.: Quasi-Fejérian analysis of some optimization algorithms. In: Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications (Haifa, 2000), volume 8 of Stud. Comput. Math., pp. 115–152. North-Holland, Amsterdam (2001)
5. Correa, R., Lemaréchal, C.: Convergence of some algorithms for convex minimization. Math. Program. **62**(2, Ser. B), 261–275 (1993)
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machine and other Kernel-Based Learning Methods. Cambridge University Press (2000)
7. Ermol'ev, Y.M.: Methods of solution of nonlinear extremal problems. Cybernetics **2**(4), 1–14 (1966)
8. Goffin, J.-L., Kiwiel, K.C.: Convergence of a simple subgradient level method. Math. Program. **85**(1, Ser. A), 207–211 (1999)
9. Grant, M., Boyd, S.: Cvx: Matlab software for disciplined convex programming, version 2.1 (2014)
10. Grant, M.C., Boyd, S.P.: Graph implementations for nonsmooth convex programs. In: Recent Advances in Learning and Control, volume 371 of Lect. Notes Control Inf. Sci., pp. 95–110. Springer, London (2008)

11. Grapiglia, G.N., Sachs, E.W.: On the worst-case evaluation complexity of non-monotone line search algorithms. Comput. Optim. Appl. **68**(3), 555–577 (2017)
12. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**(4), 707–716 (1986)
13. Hiriart-Urruty, J.-B., Lemaréchal, C.: Convex analysis and minimization algorithms. I, volume 305 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer, Berlin (1993). Fundamentals
14. Jerinkić, N.K., Ostojić, T.: AN-SPS: Adaptive sample size nonmonotone line search spectral projected subgradient method for convex constrained optimization problems. Preprint arXiv:2208.10616 (2022)
15. Kiwiel, K.C.: Methods of Descent for Nondifferentiable Optimization: Lecture Notes in Mathematics, vol. 1133. Springer, Berlin (1985)
16. Kiwiel, K.C.: Convergence of approximate and incremental subgradient methods for convex optimization. SIAM J. Optim. **14**(3), 807–840 (2003)
17. Krejic, N., Jerinkic, N.K., Ostojic, T.: Spectral projected subgradient method for nonsmooth convex optimization problems. Preprint arXiv:2203.12681, pp. 1–17 (2022)
18. Loreto, M., Crema, A.: Convergence analysis for the modified spectral projected subgradient method. Optim. Lett. **9**(5), 915–929 (2015)
19. Loreto, M., Xu, Y., Kotval, D.: A numerical study of applying spectral-step subgradient method for solving nonsmooth unconstrained optimization problems. Comput. Oper. Res. **104**, 90–97 (2019)
20. Nedić, A., Bertsekas, D.: Convergence rate of incremental subgradient algorithms. In Stochastic optimization: algorithms and applications (Gainesville, FL, 2000), volume 54 of Appl. Optim., pp. 223–264. Kluwer Acad. Publ., Dordrecht (2001)
21. Nedić, A., Bertsekas, D.P.: The effect of deterministic noise in subgradient methods. Math. Program. **125**(1, Ser. A), 75–99 (2010)
22. Nesterov, Y.: Subgradient methods for huge-scale optimization problems. Math. Program. **146**(1–2, Ser. A), 275–297 (2014)
23. Polyak, B.T., Introduction to optimization. Translations Series in Mathematics and Engineering. Optimization Software Inc, Publications Division, New York,: Translated from the Russian. With a foreword by Dimitri P, Bertsekas (1987)
24. Sachs, E.W., Sachs, S.M.: Nonmonotone line searches for optimization algorithms. Control Cybernet. **40**(4), 1059–1075 (2011)
25. Shalev-Shwartz, S., Singer, Y., Srebro, N., Pegasos, N.: Primal Estimated sub-GrAdient SOlver for SVM. In: Proceedings of the 24th International Conference on Machine Learning, pp. 807–814 (2007)
26. Shor, N.Z., Minimization methods for nondifferentiable functions, volume 3 of Springer Series in Computational Mathematics. Springer-Verlag, Berlin,: Translated from the Russian by K. C. Kiwiel and A, Ruszczyński (1985)
27. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**(4), 1043–1056 (2004)