Mathematical Programming (2020) 180:417–450
https://doi.org/10.1007/s10107-018-01361-0

FULL LENGTH PAPER

Series A

Faster subgradient methods for functions with Hölderian growth

Patrick R. Johnstone[1] · **Pierre Moulin**[2]

Received: 29 May 2017 / Accepted: 24 December 2018 / Published online: 7 January 2019

Abstract

The purpose of this manuscript is to derive new convergence results for several subgradient methods applied to minimizing nonsmooth convex functions with Hölderian growth. The growth condition is satisfied in many applications and includes functions with quadratic growth and weakly sharp minima as special cases. To this end there are three main contributions. First, for a constant and sufficiently small stepsize, we show that the subgradient method achieves linear convergence up to a certain region including the optimal set, with error of the order of the stepsize. Second, if appropriate problem parameters are known, we derive a decaying stepsize which obtains a much faster convergence rate than is suggested by the classical $O(1/\sqrt{k})$ result for the subgradient method. Thirdly we develop a novel "descending stairs" stepsize which obtains this faster convergence rate and also obtains linear convergence for the special case of weakly sharp functions. We also develop an adaptive variant of the "descending stairs" stepsize which achieves the same convergence rate without requiring an error bound constant which is difficult to estimate in practice.

Mathematics Subject Classification 65K05 · 65K10 · 90C25 · 90C30

✉ Patrick R. Johnstone
patrick.r.johnstone@gmail.com

Pierre Moulin
pmoulin@illinois.edu

[1] Department of Management Sciences and Information Systems, Rutgers Business School Newark and New Brunswick, Rutgers University, New Brunswick, USA

[2] Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801, USA

1 Introduction

1.1 Motivation and background

In this manuscript we consider the following problem:

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

where $\mathcal{C}$ is a convex, closed, and nonempty subset of a real Hilbert space $\mathcal{H}$, and $h : \mathcal{H} \to \mathbb{R}$ is a convex and closed function. We do not assume h is smooth or strongly convex. Problem (1) arises in many applications such as image processing, machine learning, compressed sensing, statistics, and computer vision [1,20,47].

We focus on the class of *subgradient methods* for solving this problem, which were first studied in the 1960s [18,40]. Since then, these methods have been used extensively because of their simplicity and low per-iteration complexity [18,30–32,39,40]. Such methods only evaluate a subgradient of the function at each iteration. However in general they have a slow worst-case convergence rate of $h(\hat{x}_k) - \min_{x \in \mathcal{C}} h(x) \leq O(1/\sqrt{k})$ after k subgradient evaluations for a particular averaged point $\hat{x}_k$. In this manuscript we show how a structural assumption for Problem (1) that is commonly satisfied in practice yields faster subgradient methods.

The structural assumption we consider is the *Hölder error bound* (throughout referred to as either HEB, HEB(c, θ), or Hölderian growth). We assume that h satisfies

$$h(x) - h^* \geq c d(x, \mathcal{X}_h)^{\frac{1}{\theta}}, \quad \forall x \in \mathcal{C}, \tag{HEB}$$

where

- $\theta \in (0, 1]$ is the "error bound exponent",
- $c > 0$ is the "error bound constant",
- $h^* = \min_{x \in \mathcal{C}} h(x)$ is the optimal value,
- $\mathcal{X}_h \triangleq \{x \in \mathcal{C} : h(x) = h^*\}$ is the solution set (assumed to be nonempty), and
- $d(x, \mathcal{X}_h) = \inf_{x^* \in \mathcal{X}_h} \|x - x^*\|$.

In general, an "error bound" is an upper bound on the distance of a point to the optimal set by some residual function. The study of error bounds has a long tradition in optimization, sensitivity analysis, systems of inequalities, projection methods, and convergence rate estimation [4,7,9,10,14,23,25,28,34,42,46,49,50]. In recent years there has been much renewed interest in the topic. HEB is often referred to as the *Łojaziewicz error bound* [6] and is also related to the *Kurdyka–Łojaziewicz (KL) inequality* [7]. In fact in [7] it was shown that the KL inequality is equivalent to HEB for convex, closed, and proper functions.

There are three main motivations for studying the behavior of algorithms for problems satisfying HEB. Firstly HEB holds for problems arising in many applications. In fact for a semialgebraic function, HEB is guaranteed to hold on a compact set for some θ and c [7]. Secondly, many algorithms have been shown to achieve significantly faster convergence behavior when HEB is satisfied. Thirdly, under HEB it has been possible to develop even faster methods.

The two most common instances of HEB in practice are $\theta = 1/2$ and $\theta = 1$. The $\theta = 1/2$ case is often referred to as the *quadratic growth condition* (QG) [23]. The $\theta = 1$ case is often referred to by saying the function has *weakly sharp minima* (WS) [9]. The function itself may also be called a weakly sharp function. There are also a small number of applications where $\theta \neq 1/2$ or 1, such as L_d regression with $d \neq 1, 2$ [1].

Due to its prevalence in applications, many recent papers have studied QG (the $\theta = 1/2$ case). QG has been used to show a *linear* convergence rate of the objective function values for various algorithms, such as the proximal gradient method, that would otherwise only guarantee sublinear convergence [4,23,48,50]. Many papers have discovered connections between QG and other error bounds and conditions known in the literature. Most importantly it was shown in [23, Appendix A] that for convex functions, QG is equivalent to the *Luo-Tseng* error bound [28], the *Polyak-Łojaziewicz* condition [23], and the *restricted secant inequality* [49].

Weakly sharp functions (i.e. HEB with $\theta = 1$) have been studied in many papers, for example [2,9,14,31,34,35,40,41,47]. For such functions [14] showed that the proximal point method converges to a minimum in a *finite* number of iterations. This is interesting because this method would otherwise only have an $O(1/k)$ rate.

1.2 Our contributions

Recall the definition of the subgradient of h at x [3, Def. 16.1]:

$$\partial h(x) \triangleq \{g \in \mathcal{H} : h(y) \geq h(x) + \langle g, y - x\rangle, \forall y \in \mathcal{H}\}.$$

Define the *subgradient method* as

$$x_{k+1} = P_{\mathcal{C}}(x_k - \alpha_k g_k): \quad \forall k \geq 1, g_k \in \partial h(x_k), \ x_1 \in \mathcal{C}, \tag{2}$$

where $P_{\mathcal{C}}$ denotes the projection onto $\mathcal{C}$ and the choice of the *stepsize* $\alpha_k > 0$ is left unspecified. Despite the long history of analysis of subgradient methods, the simplest stepsize choices for (2) have not been studied for objective functions satisfying HEB. These are the constant stepsize, $\alpha_k = \alpha$, and the decaying stepsize, $\alpha_k = \alpha_1 k^{-p}$ for $p > 0$. This brings us to our contributions in this manuscript.

Firstly we determine the convergence rate of a constant stepsize choice which previously had only been determined for the special case of $\theta = 1/2$ (see [30, Prop. 2.4]). Interestingly, *for any* $\theta \in (0, 1]$ the method obtains a linear convergence rate for $d(x_k, \mathcal{X}_h)$, up to a specific tolerance level of order $O(\alpha^\theta)$.

Secondly, we derive decaying stepsizes which obtain much faster rates than the classical subgradient method if appropriate problem parameters are available. The classical analysis of the subgradient method leads to the rate

$$h(\hat{x}_k) - h^* \leq O\left(k^{-\frac{1}{2}}\right),$$

where $\hat{x}_k$ is a specific average of the previous iterates and $\alpha_k = O(1/\sqrt{k})$ [32]. Combining this with HEB yields

$$d(\hat{x}_k, \mathcal{X}_h) \leq O\left(k^{-\frac{\theta}{2}}\right).$$

This rate is slower than the result of our specialized analysis. We show that with stepsize $\alpha_k = \alpha_1 k^{-p}$ and the proper choice of p and α_1, the subgradient method can obtain the convergence rate

$$d(x_k, \mathcal{X}_h) \leq O\left(k^{-\frac{\theta}{2(1-\theta)}}\right), \quad \forall \theta < 1. \tag{3}$$

It can be seen that the absolute value of the exponent is a factor $1/(1-\theta)$ larger in our analysis.

Our third major contribution is a new "descending stairs" stepsize choice for the subgradient method (DS-SG). The method achieves the convergence rate given in (3) for $\theta < 1$. In addition, for the case $\theta = 1$ it achieves linear convergence. Unlike the methods of [37,38] and [5, Exercise 6.3.3], which also obtain linear convergence when $\theta = 1$, our proposal does not require knowledge of h^*. The methods of [18,40,41] have a similar complexity for $\theta = 1$ but cannot handle $\theta < 1$. The Restarted Subgradient method (RSG) [47] obtains the same iteration complexity but requires averaging which is disadvantageous in applications where the solution is sparse (or low rank) because it can spoil this property [13]. (In Sect. 6.2 we discuss other problems with averaging.) An advantage of RSG is it only requires that HEB be satisfied locally, i.e. on a sufficiently-large level set of h. However in the important case where $\theta = 1$ this makes no difference, because if HEB holds with $\theta = 1$ on any compact set, then it holds globally [10]. Furthermore for many applications with $\theta < 1$, HEB is satisfied globally [7,23].

DS-SG, RSG, and several other methods [18,40] require knowledge of the constant c in HEB which can be hard to estimate in practice. This motivates us to develop our final major contribution: a "doubling trick" for DS-SG which automatically adapts to the unknown error bound constant and still obtains the same iteration complexity, up to a small constant. We call this method the "doubling trick descending stairs subgradient method" (DS2-SG). The competing methods of [18,40,41,47] all require knowledge of c. The authors of [47] proposed an adaptive method which does not require c, however it only works for $\theta < 1$.

In summary, our contributions under HEB are as follows:

1. We show that the subgradient method with a constant stepsize obtains linear convergence for $d(x_k, \mathcal{X}_h)$ to within a region of the optimal set for all $\theta \in (0, 1]$.
2. We derive a decaying stepsize with faster convergence rate than the classical subgradient method.
3. We develop a new "Descending Stairs" stepsize with iteration complexity $O(\epsilon^{1-\frac{1}{\theta}})$ when $\theta < 1$ and $\ln \frac{1}{\epsilon}$ when $\theta = 1$ for finding a point such that $d(x_k, \mathcal{X}_h)^2 \leq \epsilon$. We also develop an adaptive variant which does not need c but retains the same iteration complexity up to a small constant.

Our contributions are summarized in Table 1.

The outline for the manuscript is as follows. In Sect. 2 we discuss some previously known results for subgradient methods applied to functions satisfying HEB. In Sect. 3 we derive the key recursion which describes the subgradient method under HEB and

Table 1 Summary of our contributions for constant, decaying (polynomial), DS-SG, and DS2-SG stepsizes

	Constant	Decaying	DS-SG	DS2-SG
$\theta = 1$	$q^k + O\left(\alpha^{2\theta}\right)$	$O\left(q^k\right)$, Goffin [18]	$O\left(q^k\right)$	$O\left(q^k\right)$, c not required
$\theta < 1$	$q^k + O\left(\alpha^{2\theta}\right)$	$O\left(k^{\frac{\theta}{\theta-1}}\right)$	$O\left(k^{\frac{\theta}{\theta-1}}\right)$	$O\left(k^{\frac{\theta}{\theta-1}}\right)$, c not required

The given convergence rates are for $d(x_k, \mathcal{X}_h)^2$. We list the cases $\theta = 1$ and $\theta < 1$ seperately. Goffin [18] developed a geometrically decaying stepsize which obtains geometric convergence rate for the case $\theta = 1$ with known c (see also [40, Sec. 2.3])

allows us to obtain convergence rates. In Sect. 4 we determine the behavior of a constant stepsize. In Sect. 5 we derive a constant stepsize with explicit iteration complexity. In Sect. 6 we develop our proposed DS-SG. In Sect. 7 we develop the variant, DS2-SG, which does not require the error bound constant. In Sect. 8 we derive a decaying stepsize with faster convergence rate than the classical decaying stepsize. Finally, Sect. 9 features numerical experiments to test some of the theoretical findings of this paper.

2 Prior work on subgradient methods under HEB

There were a few early works that studied the subgradient method under conditions related to HEB with $\theta = 1$. In [40, Thm 2.7, Sec. 2.3], Shor proposed a geometrically decaying stepsize which obtains a linear convergence rate under a condition equivalent to HEB with $\theta = 1$. The stepsize depends on explicit knowledge of the error bound constant c, a bound on the subgradients, and the initial distance $d(x_1, \mathcal{X}_h)$. Goffin [18] extended the analysis of [40] to a slightly more general notion than HEB.[1] Note that our optimal decaying stepsize, derived in Sect. 8, is a natural extension of Goffin's geometrically-decaying stepsize to $\theta < 1$. Rosenburg [39] extended Goffin's results to constrained problems. In [35], Polyak showed that Goffin's method still converges linearly when the subgradients are corrupted by bounded, deterministic noise.

The paper [31] also considers functions satisfying HEB with $\theta = 1$ with (deterministically) noisy subgradients. For constant stepsizes, they show convergence of $\liminf h(x_k)$ to h^* plus a tolerance level depending on noise. For diminishing stepsizes, they show that $\liminf h(x_k)$ actually converges to h^* despite the noise. However [31] does not discuss *convergence rates*, which is the topic of our work.

As mentioned in the introduction, [47] introduced the *restarted subgradient method* (RSG) for when h satisfies HEB. The method implements a predetermined number of averaged subgradient iterations with a constant stepsize and then restarts the averaging and uses a new, smaller stepsize. The authors show that after $O(\epsilon^{2(\theta-1)} \log \frac{1}{\epsilon})$ iterations the method is guaranteed to find a point such that $h(x_k) - h^* \leq \epsilon$. For $\theta = 1$ this is a logarithmic iteration complexity. This improves the iteration complexity of the classical subgradient method which is $O(\epsilon^{-2})$. Differences between our results and RSG will be discussed in Sect. 6.2.

[1] Our analysis also holds for Goffin's condition.

The recent paper [46] extends RSG to stochastic optimization. In particular they provide a similar restart scheme that can also handle stochastic subgradient calls, and guarantees $h(x) - h^* \leq \epsilon$ with high probability. The iteration complexity is the same as for RSG, up to a constant. However, this constant is large leading to a large number of inner iterations, making it potentially difficult to implement the method in practice.

For WS functions, the paper [41] introduced a method similar to RSG except it does not require averaging at the end of each constant stepsize phase. The method also obtains a logarithmic iteration complexity in the $\theta = 1$ case. This method is essentially a special case of our proposed DS-SG for $\theta = 1$.

The paper [17] is concerned with a two-person zero-sum game equilibrium problem with a linear payoff structure. The authors show that finding the solution to the equilibrium problem is equivalent to a WS minimization problem. Using this fact, they derive a method based on Nesterov's smoothing technique with logarithmic iteration complexity. This is superior to the $O(1/\epsilon)$ of standard Nesterov smoothing. Connections between our results and [17] are discussed in Sect. 6.2.

The work [27] studies stochastic subgradient descent under the assumption that the function satisfies WS locally and QG globally. They show a faster convergence rate of the iterates to a minimizer, both in expectation and with high probability, than is known under the classical analysis.

The work [15] proposes a new subgradient method for functions satisfying a similar condition to HEB but with h^* replaced by a strict lower bound on h^*. Like RSG, this algorithm has a logarithmic dependence on the initial distance to the solution set. However it still obtains an $O(1/\epsilon^2)$ iteration complexity, which is the same as the classical subgradient method.

In [37,38] Renegar presented a framework for converting a convex conic program to a general convex problem with an affine constraint, to which projected subgradient methods can be applied. He further showed how this can be applied to general convex optimization problems, such as Prob. (1), by representing them as a conic problem. For the special case where the objective and constraint set is polyhedral, one of the subgradient methods proposed by Renegar has a logarithmic iteration complexity [37, Cor. 3.4]. The main drawback of this method is that it requires knowledge of the optimal value, h^*. It also requires a point in the interior of the constraint set. Similarly the stepsizes proposed in Thm. 2 of [36, Sec 5.3.] and [30, Prop. 2.11] depend on exact knowledge of h^* and also obtain a logarithmic iteration complexity under WS.

The work [33] explores subgradient-type algorithms for nonsmooth nonconvex functions satisfying the KL inequality. A procedure was developed for selecting a subgradient at each iteration which results in a decrease in objective value, thereby leading to convergence to a critical point. The selection procedure typically involves either storing a collection of past subgradients and solving a convex program, or suitably backtracking the stepsize until a certain condition is met.

For WS functions, it is known that there are subgradient methods which obtain linear convergence [18,40,47]. A different assumption, known as partial smoothness, has been used to show *local* linear convergence of proximal gradient methods [19,26]. We mention that the partial smoothness property is different to WS: it applies to composite optimization problems with objective: $F = f + h$ where h must be smooth

but f may be nonsmooth. Unlike subgradient methods, in proximal gradient methods the nonsmooth part f is addressed via its proximal operator.

In recent times, convergence analyses for the subgradient method have focused on the objective function rather than the distance of the iterates from the optimal set. However in the early period of development, there were many works focusing on the distance (e.g. [18,30,35,40]). The subgradient method is not a descent method with respect to function values, however it is with respect to the distances to the optimal set. Thus the distance is a natural metric to study for the subgradient method. Furthermore, for some applications, the distance to the solution set arguably matters more than the objective function value. For example in machine learning, the objective function is only a surrogate for the actual objective of interest—expected prediction error.

Without further assumptions, [36, pp. 167–168] showed that the convergence rate of the distance of the iterates of the subgradient method to the optimal set can be made arbitrarily slow. This is true even for smooth convex problems. In this case, gradient descent with a constant stepsize obtains an $O(1/k)$ *objective function* convergence rate, however the iterates can be made to converge arbitrarily slowly to a minimizer. It is our use of HEB which allows us to derive less pessimistic convergence rates for the distance to the optimal set.

3 The key recursion

In this section we derive the recursion which describes the evolution of the squared error $d(x_k, \mathcal{X}_h)^2$ for the iterates of the standard subgradient method under HEB. The same recursion has been derived many times before for the special cases $\theta = \{1/2, 1\}$ (e.g. [18,30,40]).

3.1 Assumptions

The optimality condition for Prob. (1) can be found in [3, Prop. 26.5]. Note that we do not explicitly use this optimality criterion anywhere in our analysis. For Prob. (1), throughout the manuscript we will assume that $\mathcal{C} \subseteq \text{dom}(\partial h)$, so that for any query point $x \in \mathcal{C}$ it is possible to find a $g \in \partial h(x)$. If h is convex and closed, the solution set $\mathcal{X}_h = \{x : h(x) = h^*\}$ is convex and closed [3]. Here are the precise assumptions we will use throughout the manuscript.

Assumption 3 (*Problem* (1)) Assume $\mathcal{C}$ is convex, closed, and nonempty. Assume h is convex, closed, and satisfies HEB(c, θ). Assume $\mathcal{X}_h$ is nonempty. Assume $\mathcal{C} \subseteq \text{dom}(\partial h)$. Assume there exists a constant G such that $\|g\| \leq G$ for all $g \in \partial h(x)$ and $x \in \mathcal{C}$.

Throughout the manuscript let $\kappa \triangleq G/c$.

3.2 The recursion under HEB

Proposition 1 *Suppose Assumption 3 holds. Then for all $k \geq 1$ for the iterates $\{x_k\}$ of* (2)

$$d(x_{k+1}, \mathcal{X}_h)^2 \leq d(x_k, \mathcal{X}_h)^2 - 2\alpha_k c(d(x_k, \mathcal{X}_h)^2)^{\frac{1}{2\theta}} + \alpha_k^2 G^2. \tag{4}$$

Proof For the point x_k let x_k^* be the unique projection of x_k onto $\mathcal{X}_h$. For $k \geq 1$,

$$\begin{aligned} d(x_{k+1}, \mathcal{X}_h)^2 &= \|x_{k+1} - x_{k+1}^*\|^2 \\ &\leq \|x_{k+1} - x_k^*\|^2 \\ &\leq d(x_k, \mathcal{X}_h)^2 - 2\alpha_k \langle g_k, x_k - x_k^* \rangle + \alpha_k^2 \|g_k\|^2 \\ &\leq d(x_k, \mathcal{X}_h)^2 - 2\alpha_k \left(h(x_k) - h^*\right) + \alpha_k^2 G^2 \\ &\leq d(x_k, \mathcal{X}_h)^2 - 2\alpha_k c(d(x_k, \mathcal{X}_h)^2)^{\frac{1}{2\theta}} + \alpha_k^2 G^2. \end{aligned}$$

In the first inequality, we used the fact that x_{k+1}^* is the closest point to x_{k+1} in $\mathcal{X}_h$. In the second inequality, we used the nonexpansiveness of the projection operator. In the third, we used the convexity of h and in the final inequality we used the error bound. □

Let $e_k \triangleq d(x_k, \mathcal{X}_h)^2$ and $\gamma = \frac{1}{2\theta} \in [\frac{1}{2}, +\infty)$ then for all $k \geq 1$

$$0 \leq e_{k+1} \leq e_k - 2\alpha_k c e_k^{\gamma} + \alpha_k^2 G^2. \tag{5}$$

The main effort of our analysis is in deriving convergence rates for this recursion for various stepsizes.

We note that the key recursion (4) can also be derived with different constants in the following extensions:

1. For $\theta = 1$ a small (relative to c) amount of deterministic noise can be added to the subgradient [31];
2. Instead of (2) one can consider the *incremental* subgradient method [30], the *proximal* subgradient method [12]:

$$x_{k+1} = \text{prox}_{\alpha_k f}(x_k - \alpha_k g_k): \quad \forall k \geq 1, g_k \in \partial h(x_k), \ x_1 \in \text{dom}(\partial h),$$

for minimizing $F(x) = f(x) + h(x)$, so long as the composite function F satisfies HEB and $\text{dom}(\partial h) \subseteq \text{dom}(f)$, or the *relaxed* projected subgradient method:

$$x_{k+1} = (1 - \lambda_k)x_k + \lambda_k P_{\mathcal{C}}(x_k - \alpha_k g_k): \quad \forall k \geq 1, g_k \in \partial h(x_k), \ x_1 \in \mathcal{C},$$

so long as $0 < \underline{\lambda} \leq \lambda_k \leq 1$.

4 Constant stepsize

Consider the projected subgradient method with *constant*, or fixed, stepsize α given in Algorithm FixedSG. Previously it was known that if $\theta = 1/2$ then this method achieves linear convergence to within a region of the solution set [23,30]. We show in the next theorem that linear convergence to within a certain region of $\mathcal{X}_h$ occurs for any $\theta \in (0, 1]$ provided α is sufficiently small.

Algorithm 1: (FixedSG)

Require: $K > 0, \alpha > 0, x_1 \in \mathcal{C}$
1: **for** $k = 1, 2, \ldots, K$ **do**
2: $\quad x_{k+1} = P_{\mathcal{C}}\left(x_k - \alpha g_k\right) : \quad g_k \in \partial h(x_k)$
3: **end for**
4: **return** x_{k+1}

Theorem 1 *Suppose Assumption 3 holds. Let* $e_* = \left(\frac{\alpha G^2}{2c}\right)^{2\theta}$.

1. *For all* $k \geq 1$ *the iterates of FixedSG satisfy*

$$d(x_k, \mathcal{X}_h)^2 \leq \max\left\{d(x_1, \mathcal{X}_h)^2, e_* + \alpha^2 G^2\right\}. \tag{6}$$

2. *If* $0 < \theta \leq \frac{1}{2}$ *and*

$$0 < \alpha \leq 2^{\frac{1-2\theta}{2(1-\theta)}} \theta^{\frac{1}{2(1-\theta)}} G^{\frac{2\theta-1}{1-\theta}} c^{\frac{\theta}{\theta-1}} \tag{7}$$

then for all $k \geq 1$ *the iterates of FixedSG satisfy*

$$d(x_k, \mathcal{X}_h)^2 - e_* \leq q_1^{k-1}(d(x_1, \mathcal{X}_h)^2 - e_*) \tag{8}$$

where

$$q_1 = \left(1 - \frac{1}{\theta}\alpha c e_*^{\frac{1-2\theta}{2\theta}}\right) \in [0, 1). \tag{9}$$

3. *Suppose there exists* $D \geq 0$ *s.t.* $d(x_k, \mathcal{X}_h)^2 \leq D$ *for all* k. *If* $\frac{1}{2} \leq \theta \leq 1$ *and the stepsize is chosen s.t.*

$$0 < \alpha \leq \frac{\theta D^{1-\frac{1}{2\theta}}}{c}, \tag{10}$$

then for all $k \geq 1$ *the iterates of FixedSG satisfy*

$$d(x_k, \mathcal{X}_h)^2 - e_* \leq \max\left\{q_2^{k-1}\left(d(x_1, \mathcal{X}_h)^2 - e_*\right), \alpha^2 G^2\right\}, \tag{11}$$

where

$$q_2 = 1 - \frac{\alpha c D^{\frac{1}{2\theta}-1}}{\theta} \in [0, 1).$$

Note that in part 3 of Theorem 1, we assume the existence of a bound D s.t. $d(x_k, \mathcal{X}_h)^2 \leq D$ for all $k \in \mathbb{N}$. Such a bound was provided in part 1 of the theorem. However for the sake of notational clarity we prove part 3 with a generic upper bound D.

Proof Recall our notation $e_k = d(x_k, \mathcal{X}_h)^2$ and let $\gamma = \frac{1}{2\theta}$. Returning to the main recursion (5) derived in Prop. 1 and replacing the stepsize with a constant yields

$$0 \leq e_{k+1} \leq e_k - 2\alpha c e_k^\gamma + \alpha^2 G^2, \tag{12}$$

where $\gamma \geq \frac{1}{2}$. The key to understanding the behavior of this recursion is to write it as

$$e_{k+1} - e_* \leq e_k - e_* - 2\alpha c \left(e_k^\gamma - e_*^\gamma\right) \tag{13}$$

where $e_* = \left(\frac{\alpha G^2}{2c}\right)^{\frac{1}{\gamma}}$. We will show that $e_k - e_*$ must go to 0 and derive the convergence rate.

Boundedness

We first prove (6), which says that e_k is bounded. Considering (13) we see that since $\alpha > 0$ and $c > 0$, if $e_k \geq e_*$ then $e_{k+1} \leq e_k$. On the other hand, if $e_k \leq e_*$, then (12) yields $e_{k+1} \leq e_k + \alpha^2 G^2 \leq e_* + \alpha^2 G^2$. Therefore

$$e_{k+1} \leq \max\left\{e_k, e_* + \alpha^2 G^2\right\} \leq \max\left\{e_1, e_* + \alpha^2 G^2\right\}.$$

Case 1: $\boldsymbol{\theta \leq \frac{1}{2}}$.

Suppose $\theta \leq \frac{1}{2}$ and hence $\gamma \geq 1$. By the convexity of t^γ,

$$e_k^\gamma - e_*^\gamma \geq \gamma e_*^{\gamma-1}(e_k - e_*).$$

Using this in (13) along with the facts that $\alpha > 0$ and $c > 0$ yields

$$e_{k+1} - e_* \leq \left(1 - 2\alpha c\gamma e_*^{\gamma-1}\right)(e_k - e_*).$$

Thus so long as

$$1 - 2\alpha c\gamma e_*^{\gamma-1} \geq 0, \tag{14}$$

we have $q_1 \geq 0$ where q_1 is defined in (9) and

$$e_{k+1} - e^* \leq q_1\left(e_k - e^*\right) \leq q_1^k\left(e_1 - e^*\right)$$

where the second inequality comes from recursing. This proves (8).

Simplifying (14) yields

$$\begin{aligned} 2\alpha c\gamma e_*^{\gamma-1} &\leq 1 \\ &\iff \alpha c\gamma \left(\frac{\alpha G^2}{2c}\right)^{\frac{\gamma-1}{\gamma}} \leq 2^{-1} \end{aligned}$$

$$\iff \alpha \leq \left(\frac{1}{\gamma} G^{\frac{2(1-\gamma)}{\gamma}} 2^{-\frac{1}{\gamma}} c^{-\frac{1}{\gamma}}\right)^{\frac{\gamma}{2\gamma-1}}$$

which is equivalent to (7).

Case 2: $\boldsymbol{\theta \geq \frac{1}{2}}$.
For $\theta \in [\frac{1}{2}, 1]$, $\gamma \in [\frac{1}{2}, 1]$, which implies by concavity

$$e_*^\gamma - e_k^\gamma \leq \gamma e_k^{\gamma-1}(e_* - e_k).$$

Therefore

$$e_k^\gamma - e_*^\gamma \geq \gamma e_k^{\gamma-1}(e_k - e_*).$$

Substituting this inequality into (13) and again using $\alpha > 0$ and $c > 0$ yields

$$e_{k+1} - e_* \leq e_k - e_* - 2\alpha c\gamma e_k^{\gamma-1}(e_k - e_*).$$

Now if $e_* \leq e_k$, then since $e_k \leq D$,

$$e_{k+1} - e_* \leq \left(1 - 2\alpha c\gamma D^{\gamma-1}\right)(e_k - e_*) = q_2(e_k - e_*).$$

So long as

$$1 > 1 - 2\alpha c\gamma D^{\gamma-1} \geq 0$$

[which is implied by (10)], we have $q_2 \in [0, 1)$. On the other hand if $e_k \leq e_*$ then, using (12), $e_{k+1} \leq e_* + \alpha^2 G^2$. Thus for all $k \geq 1$

$$e_{k+1} - e_* \leq \max\left\{q_2(e_k - e_*), \alpha^2 G^2\right\}.$$

Iterating this recursion and using the fact that $q_2 \in [0, 1)$ yields (11). □

5 Iteration complexity for constant stepsize

Using the results of the previous section we can derive the iteration complexity of a constant stepsize for finding a point such that $d(x_k, \mathcal{X}_h)^2 \leq \epsilon$. The basic idea in the following theorem is to pick $\alpha = O(\epsilon^{\frac{1}{2\theta}})$, so that e_* defined in Theorem 1 is equal to ϵ. Then the iteration complexity can be determined from the linear convergence rate of $d(x_k, \mathcal{X}_h)^2$ to e_*.

Theorem 2 *Suppose Assumption 3 holds. Choose $\epsilon > 0$ and set*

$$\alpha = \frac{2c\epsilon^{\frac{1}{2\theta}}}{G^2}. \tag{15}$$

1. *If* $0 < \theta \leq \frac{1}{2}$, *and*

$$0 < \epsilon \leq \left(\frac{\theta\kappa^2}{2}\right)^{\frac{\theta}{1-\theta}}, \tag{16}$$

then for the iterates of FixedSG,

$$d(x_{k+1}, \mathcal{X}_h)^2 \leq 2\epsilon$$

for all $k \geq K$ *where*

$$K \triangleq \frac{1}{2}\theta\kappa^2 \ln\left(\frac{d(x_1, \mathcal{X}_h)^2}{\epsilon}\right)\epsilon^{1-\frac{1}{\theta}}.$$

2. *For* $\frac{1}{2} \leq \theta \leq 1$, *assume* $\hat{D} > 0$ *and* $\epsilon > 0$ *are chosen s.t.*

$$d(x_1, \mathcal{X}_h)^2 \leq \hat{D} \tag{17}$$

$$\epsilon \leq \min\left\{\frac{\hat{D}}{2}, \left(\frac{\theta\kappa^2}{2}\right)^{2\theta}\hat{D}^{2\theta-1}\right\}. \tag{18}$$

We further require

$$\begin{cases} \epsilon \leq \left(\frac{\kappa^2}{4}\right)^{\frac{\theta}{1-\theta}} & : \text{ if } \theta < 1 \\ \kappa \geq 2 & : \text{ if } \theta = 1. \end{cases} \tag{19}$$

Then for the iterates of FixedSG,

$$d(x_{k+1}, \mathcal{X}_h)^2 \leq 2\epsilon$$

for all $k \geq K$, *where*

$$K \triangleq \frac{1}{2}\theta\kappa^2\hat{D}^{1-\frac{1}{2\theta}} \ln\left(\frac{d(x_1, \mathcal{X}_h)^2}{\epsilon}\right)\epsilon^{-\frac{1}{2\theta}}. \tag{20}$$

Proof We consider the two cases, $\theta \leq 1/2$ and $\theta \geq 1/2$, separately.

Case 1: $\boldsymbol{\theta \leq \frac{1}{2}}$.
From Theorem 1, the convergence factor in the constant stepsize case is $q_1 = 1 - \frac{\alpha c}{\theta}e_*^{\frac{1}{2\theta}-1}$ where $e_* = \left(\frac{\alpha G^2}{2c}\right)^{2\theta} = \epsilon$ for this choice of α given in (15). Recall the notation $e_k = d(x_k, \mathcal{X}_h)^2$. Since ϵ satisfies (16), $0 \leq q_1 < 1$. Thus from Theorem 1 we know that for all $k \geq 1$

$$e_{k+1} - e_* \leq q_1^k(e_1 - e^*) \leq q_1^k e_1$$

which implies

$$\max\{e_{k+1} - e_*, 0\} \leq q_1^k e_1. \tag{21}$$

This means that

$$\ln(\max\{0, e_{k+1} - e_*\}) \leq k \ln q_1 + \ln e_1$$

using the convention, $\ln(0) = -\infty$. Thus $e_{k+1} - e_* \leq \epsilon$ is implied by

$$k \ln q_1 + \ln e_1 \leq \ln \epsilon \iff k \geq \frac{\ln \frac{e_1}{\epsilon}}{\ln \frac{1}{q_1}}.$$

Now

$$\ln q_1 = \ln\left(1 - \frac{\alpha c}{\theta} e_*^{\frac{1}{2\theta}-1}\right) \leq -\frac{\alpha c}{\theta} e_*^{\frac{1}{2\theta}-1} \iff \ln \frac{1}{q_1} \geq \frac{\alpha c}{\theta} e_*^{\frac{1}{2\theta}-1}.$$

Therefore if

$$k \geq \frac{\theta \ln \frac{e_1}{\epsilon}}{\alpha c e_*^{\frac{1}{2\theta}-1}} = \frac{\theta G^2 \ln \frac{e_1}{\epsilon}}{2c^2 \epsilon^{\frac{1}{\theta}-1}} = \frac{1}{2}\theta \kappa^2 \ln\left(\frac{e_1}{\epsilon}\right) \epsilon^{1-\frac{1}{\theta}}$$

then using the fact that for this choice of α, $e_* = \epsilon$, we arrive at

$$e_{k+1} \leq e_* + \epsilon = 2\epsilon.$$

Case 2: $\boldsymbol{\theta \geq \frac{1}{2}}$.

As before, $\alpha = \frac{2c\epsilon^{\frac{1}{2\theta}}}{G^2}$ which implies $e_* = \epsilon$. First note that by Part 1 of Theorem 1,

$$\begin{aligned} d(x_k, \mathcal{X}_h)^2 &\leq \max\left\{d(x_1, \mathcal{X}_h)^2, e_* + \alpha^2 G^2\right\} \\ &= \max\left\{d(x_1, \mathcal{X}_h)^2, \epsilon + \frac{4c^2}{G^2}\epsilon^{\frac{1}{\theta}}\right\} \\ &\leq \max\{d(x_1, \mathcal{X}_h)^2, 2\epsilon\} \\ &\leq \hat{D} \end{aligned}$$

for all $k \geq 1$, where in the second inequality we used (17) and (19). Therefore $\hat{D}$ is a valid upper bound for the sequence $\{d(x_k, \mathcal{X}_h)^2\}$ and can be used in place of D in Theorem 1. Now from Theorem 1 the convergence factor is

$$q_2 = 1 - \frac{\alpha c}{\theta} \hat{D}^{\frac{1}{2\theta}-1}$$

which is greater than or equal to 0 (and less than 1) because ϵ satisfies (18). Recalling (11) we see that

$$e_{k+1} \leq \max\left\{e_* + q_2^k\left(d(x_1, \mathcal{X}_h)^2 - e_*\right), e_* + \alpha^2 G^2\right\}. \tag{22}$$

We have already shown that the second argument in the max above is upper bounded by 2ϵ. Consider the first argument in the max in (22). Now $q_2^k(d(x_1, \mathcal{X}_h)^2 - e_*) \leq q_2^k e_1$, thus this argument can be dealt with the same way as Case 1 for $\theta \leq 1/2$, except for a different convergence factor. Thus we observe

$$\ln q_2 = \ln\left(1 - \frac{\alpha c}{\theta}\hat{D}^{\frac{1}{2\theta}-1}\right) \leq -\frac{\alpha c}{\theta}\hat{D}^{\frac{1}{2\theta}-1} \iff \ln\frac{1}{q_2} \geq \frac{\alpha c}{\theta}\hat{D}^{\frac{1}{2\theta}-1}.$$

Therefore if

$$k \geq \frac{\theta G^2 \hat{D}^{1-\frac{1}{2\theta}}}{2c^2} \ln\left(\frac{e_1}{\epsilon}\right)\epsilon^{-\frac{1}{2\theta}}$$

then the first argument in the max in (22) is upper bounded by 2ϵ. □

Rather surprisingly, Theorem 2 shows that a restarting strategy is not necessary for $\theta \leq \frac{1}{2}$. This is because for $\theta \leq \frac{1}{2}$ the iteration complexity for a constant stepsize is equal to the complexity of RSG derived in [47]. It is also matched by the optimal decaying stepsize we will derive in Sect. 8. To compare with RSG in more detail, [47] showed that RSG requires $O(\epsilon'^{2(\theta-1)})$ iterations (suppressing constants and a $\ln\frac{1}{\epsilon}$ factor) to achieve $h(x) - h^* \leq \epsilon'$. Now, using the error bound, in order to guarantee $d(x_k, \mathcal{X}_h)^2 \leq \epsilon$, we need $h(x) - h^* \leq \epsilon' = \epsilon^{\frac{1}{2\theta}}$. Using this in the iteration complexity from [47] yields an expression which is $O(\epsilon^{1-\frac{1}{\theta}})$, which is the same as what we derived for the constant stepsize for $\theta \leq 1/2$. However, for $\theta > \frac{1}{2}$, RSG, our DS-SG method, and our optimal decaying stepsize are significantly faster than the constant stepsize choice.

The comparison with the classical result for the subgradient method is as follows. It is easy to show that for the subgradient method with a constant stepsize α:

$$\frac{1}{k}\sum_{i=1}^{k}(h(x_i) - h^*) \leq \frac{d(x_1, \mathcal{X}_h)^2}{2\alpha k} + \frac{\alpha}{2}G^2.$$

Setting

$$\alpha = \frac{c\epsilon^{\frac{1}{2\theta}}}{G^2}$$

and

$$k \geq \kappa^2 d(x_1, \mathcal{X}_h)^2 \epsilon^{-1/\theta}$$

implies

$$h\left(x_k^{av}\right) - h^* \leq \frac{1}{k}\sum_{i=1}^{k}(h(x_i) - h^*) \leq c\epsilon^{1/2\theta}$$

where $x_k^{av} = \frac{1}{k}\sum_{i=1}^{k} x_i$. Now using the error bound, this yields $d(x_k^{av}, \mathcal{X}_h)^2 \leq \epsilon$. With respect to ϵ, this classical iteration complexity is clearly worse than the result of Theorem 1 for all $\theta \in (0, 1]$. Furthermore, the dependence on $d(x_1, \mathcal{X}_h)$ is worse. For $\theta \leq 1/2$, the fixed stepsize depends on $\ln d(x_1, \mathcal{X}_h)$, whereas the classical stepsize has iteration complexity which depends linearly on $d(x_1, \mathcal{X}_h)$.

We note that as $\theta \to 0$ the iteration complexity can be made arbitrarily large. This is not suprising, as it has been proved in [36, pp. 167–168] that the convergence rate of $x_k \to x^*$ can be made arbitrarily bad for gradient methods.

6 A "descending stairs" stepsize with better iteration complexity for $1/2 \leq \theta \leq 1$

6.1 The method

In this section we propose a new stepsize for the subgradient method (DS-SG) which obtains a better iteration complexity than the fixed stepsize for functions satisfying HEB with $1/2 \leq \theta \leq 1$. In fact for $\theta = 1$ the iteration complexity is logarithmic, i.e. $O(\ln \frac{1}{\epsilon})$. The basic idea is to use a constant stepsize in the subgradient method and every K iterations reduce the stepsize by a factor of $\beta_{ds}^{\frac{1}{2\theta}} > 1$. Also the number of iterations K increases by a factor $\beta_{ds}^{\frac{1}{\theta}-1}$. Our analysis allows us to determine good choices for the initial stepsize and number of iterations which lead to an improved rate.

The algorithm is similar to RSG [47]. However our method has some important advantages, which will be discussed in Sect. 6.2, and a different analysis. As was mentioned earlier, the method of [41, Sec. V] is essentially a special case of DS-SG for $\theta = 1$.

DS-SG requires an upper bound on the distance of the starting point to the solution, i.e. $\Omega_1 \geq d(x_{\text{init}}, \mathcal{X}_h)^2$. If $\mathcal{C}$ is bounded then one can use the diameter of $\mathcal{C}$. If a lower bound on the optimal value is known, i.e. $h_l \leq h^*$, then by the error bound $d(x_1, \mathcal{X}_h) \leq c^{-\theta}(h(x_1) - h^*)^\theta \leq c^{-\theta}(h(x_1) - h_l)^\theta$ implies we can use $\Omega_1 = c^{-2\theta}(h(x_1) - h_l)^{2\theta}$.

Theorem 3 *Suppose Assumption 3 holds and $\frac{1}{2} \leq \theta \leq 1$. Choose $x_{init} \in \mathcal{C}$ and Ω_1 such that $d(x_{init}, \mathcal{X}_h)^2 \leq \Omega_1$. If $\theta < 1$, choose $\beta_{ds} > 1$ so that*

$$\beta_{ds} \geq \max\left\{\frac{1}{2}\left(\frac{\kappa^2}{4}\right)^{\frac{\theta}{\theta-1}}\Omega_1, \theta^{-2\theta}\kappa^{-4\theta}\Omega_1^{2(1-\theta)}\right\}. \tag{23}$$

Algorithm 2: (DS-SG) Descending Stairs Subgradient Method for $1/2 \leq \theta \leq 1$

Require: β_{ds}, M, x_{init}, Ω_1, G, c, θ.
$\kappa = \frac{G}{c}$
$\tilde{K}_1 = \theta \kappa^2 \beta_{ds}^{\frac{1}{2\theta}} \ln(2\beta_{ds}) \, \Omega_1^{1-\frac{1}{\theta}}$
$K_1 = \lceil \tilde{K}_1 \rceil$
$\alpha(1) = \frac{2c}{G^2} \left(\frac{\Omega_1}{2\beta_{ds}} \right)^{\frac{1}{2\theta}}$
$\hat{x}_0 = x_{\text{init}}$
for $m = 1, 2, \ldots, M$ **do**
  $\hat{x}_m = \text{FixedSG}(K_m, \alpha(m), \hat{x}_{m-1})$
  $\alpha(m+1) = \beta_{ds}^{-\frac{1}{2\theta}} \alpha(m)$
  $K_{m+1} = \left\lceil \beta_{ds}^{\frac{m(1-\theta)}{\theta}} \tilde{K}_1 \right\rceil$
end for
return $\hat{x}_M$

If $\theta = 1$, assume $\kappa \geq 2$ and choose any $\beta_{ds} > 1$. Fix $\epsilon > 0$ and choose $M \geq \left\lceil \frac{\ln \frac{\Omega_1}{\epsilon}}{\ln \beta_{ds}} \right\rceil$. Then for $\hat{x}_M$ returned by Algorithm DS-SG, $d(\hat{x}_M, \mathcal{X}_h)^2 \leq \epsilon$. The iteration complexity is as follows:

1. *If $\theta = 1$ this requires fewer than*

$$\left(\beta_{ds}^{\frac{1}{2}} \kappa^2 \ln(2\beta_{ds}) + 1 \right) \left(\frac{\ln \frac{\Omega_1}{\epsilon}}{\ln \beta_{ds}} + 1 \right) \tag{24}$$

   subgradient evaluations. This simplifies to

$$O\left(\kappa^2 \ln \frac{\Omega_1}{\epsilon} \right) \tag{25}$$

   as $\kappa, \Omega_1 \to \infty$, and $\epsilon \to 0$.
2. *If $\frac{1}{2} \leq \theta < 1$, this requires fewer than*

$$\frac{\theta \beta_{ds}^{\frac{3}{2\theta}-1} \ln(2\beta_{ds})}{\beta_{ds}^{\frac{1}{\theta}-1} - 1} \kappa^2 \epsilon^{1-\frac{1}{\theta}} + \frac{\ln \frac{\Omega_1}{\epsilon}}{\ln \beta_{ds}} + 1 \tag{26}$$

   subgradient evaluations. If κ is chosen large enough so that $\Omega_1 = O(\kappa^{\frac{2\theta}{1-\theta}})$, this simplifies to

$$O\left(\max\{\kappa^2, \Omega^{\frac{1}{\theta}-1}\} \epsilon^{1-\frac{1}{\theta}} \right) \tag{27}$$

   as $\kappa, \Omega_1 \to \infty$, and $\epsilon \to 0$.

Proof We need some new notation. For $\hat{x}_m$ defined in line 7 of DS-SG, let $\hat{e}_m = d(\hat{x}_m, \mathcal{X}_h)^2$. We will use a sequence of tolerances $\{\epsilon_m\}$ defined as $\epsilon_m = \beta_{ds}^{-m}\Omega_1$. Another sequence $\{D_m\}$ is chosen as

$$D_m = 2\beta_{ds}\epsilon_m.$$

For each $m \geq 1$, the set $\{\epsilon_m/2, D_m, \alpha(m)\}$ will be used in statement 2 of Theorem 2 in place of $\{\epsilon, \hat{D}, \alpha\}$. Furthermore we will show that K_m is greater than the corresponding expression for K in (20). This will show that $\hat{e}_m \leq 2(\epsilon_m/2) = \epsilon_m$.

We now show that $\{\epsilon_m/2, D_m, \alpha(m)\}$ satisfies (15), (17), (18), and (19), and that K_m is greater than K given in (20). Now the stepsize $\alpha(m)$, defined on lines 4 and 8 of DS-SG, can be written as

$$\alpha(m) = \frac{2c}{G^2}\left(\frac{\epsilon_m}{2}\right)^{\frac{1}{2\theta}}.$$

Thus $\alpha(m)$ satisfies (15) for all $m \geq 1$. Next we prove that for $\frac{1}{2} \leq \theta < 1$, condition (23) ensures that (18)–(19) are satisfied for all $m \geq 1$.

To establish (18), we will prove that both arguments in the min in (18) individually satisfy the inequality when $\hat{D}$ and ϵ are replaced by D_m and $\epsilon_m/2$. Since $\beta_{ds} > 1$, it is clear that the first argument in the min in (18) satisfies the inequality. Now for the second argument in the min in (18) to satisfy the inequality we require

$$\frac{\epsilon_m}{2} \leq \left(\frac{\theta\kappa^2}{2}\right)^{2\theta} D_m^{2\theta-1} = \frac{1}{2}\left(\theta\kappa^2\right)^{2\theta}\beta_{ds}^{2\theta-1}\epsilon_m^{2\theta-1}.$$

Using $\epsilon_m = \beta_{ds}^{-m}\Omega_1$ and rearranging this yields

$$\beta_{ds}^{2m(1-\theta)+2\theta-1} \geq \theta^{-2\theta}\kappa^{-4\theta}\Omega_1^{2(1-\theta)}. \tag{28}$$

In order to hold for all $m \geq 1$ it suffices to show it holds for $m = 1$, which is implied by the second argument in the max in (23). In the case $\theta = 1$, (28) reduces to

$$\beta_{ds} \geq \frac{1}{\kappa^4}.$$

Note for $\theta = 1$, by assumption $\kappa \geq 2$. Thus any $\beta_{ds} > 1$ satisfies this.

Now we establish (19). For $\theta = 1$ we only require that $\kappa \geq 2$ which is true by assumption. We now establish (19) for $\frac{1}{2} \leq \theta < 1$. In this case, (19) requires that

$$\frac{\epsilon_m}{2} = \frac{1}{2}\beta_{ds}^{-m}\Omega_1 \leq \left(\frac{\kappa^2}{4}\right)^{\frac{\theta}{1-\theta}}.$$

In order for this to be satisfied for all m, it suffices to show that it holds for $m = 1$. This is implied by the first argument in the max function in (23).

Finally we prove by induction that (17) holds and that K_m is greater than K defined in (20). For $m = 1$, D_1 clearly satisfies (17). Also K_1, given in Line 1 of Algorithm DS-SG, satisfies (20). Altogether this implies $\hat{e}_1 \leq \epsilon_1$ by Theorem 2.

Next, assume (17) is true and K_{m-1} is greater than K in (20) at iteration $m-1$. Since we have established (15), (18), and (19) hold for all $m \geq 1$, part 2 of Theorem 1 implies that $\hat{e}_{m-1} \leq \epsilon_{m-1}$. At iteration m, FixedSG is initialized at $\hat{x}_{m-1}$, and $d(\hat{x}_{m-1}, \mathcal{X})^2 \leq \epsilon_{m-1}$, thus

$$D_m = 2\beta_{ds}\epsilon_m = 2\epsilon_{m-1} \geq d(\hat{x}_{m-1}, \mathcal{X})^2$$

which establishes (17) at iteration m. Next, substituting D_m and $\epsilon_m/2$ in for $\hat{D}$ and ϵ in (20), we see that K_m needs to be greater than

$$\frac{1}{2}\theta\kappa^2 \ln\left(\frac{2d(\hat{x}_{m-1}, \mathcal{X}_h)^2}{\epsilon_m}\right) D_m^{1-\frac{1}{2\theta}} (\epsilon_m/2)^{-\frac{1}{2\theta}}$$

which is indeed true since K_m can be re-expressed as

$$\begin{aligned} K_m &= \left\lceil \theta\kappa^2 \beta_{ds}^{\frac{1}{2\theta}} \ln(2\beta_{ds}) \Omega_1^{1-\frac{1}{\theta}} \beta_{ds}^{-(m-1)(1-\frac{1}{\theta})} \right\rceil \\ &= \left\lceil \theta\kappa^2 \beta_{ds}^{1-\frac{1}{2\theta}} \ln(2\beta_{ds}) \epsilon_m^{1-\frac{1}{\theta}} \right\rceil \\ &\geq \frac{1}{2}\theta\kappa^2 \ln\left(\frac{2d(\hat{x}_{m-1}, \mathcal{X}_h)^2}{\epsilon_m}\right) (2\beta_{ds}\epsilon_m)^{1-\frac{1}{2\theta}} (\epsilon_m/2)^{-\frac{1}{2\theta}}. \end{aligned}$$

We have shown that $\{\epsilon_m/2, D_m, \alpha(m)\}$ satisfies (15), (17), (18), and (19), and that K_m is greater than K defined in (20). Thus by part 2 of Theorem 1, for all $m \geq 1$ $\hat{e}_m \leq 2(\epsilon_m/2) = \epsilon_m$. Finally the choice $M = \left\lceil \frac{\ln \frac{\Omega_1}{\epsilon}}{\ln \beta_{ds}} \right\rceil$ implies $\epsilon_M = \beta_{ds}^{-M}\Omega_1 \leq \epsilon$.

If $\theta = 1$, the total number of subgradient evaluations is

$$MK_1 \leq \left(\kappa^2 \beta_{ds}^{\frac{1}{2}} \ln(2\beta_{ds}) + 1\right)\left(\frac{\ln \frac{\Omega_1}{\epsilon}}{\ln \beta_{ds}} + 1\right)$$

where we have used $\lceil x \rceil < x + 1$. Further note that for $\theta = 1$, β_{ds} is a constant that can be chosen independently of κ, Ω, and ϵ, which implies (25).

We now establish the iteration complexity when $\frac{1}{2} \leq \theta < 1$. For $m \geq 0$, let

$$\tilde{K}_{m+1} = \beta_{ds}^{\frac{m(1-\theta)}{\theta}} \theta\kappa^2 \beta_{ds}^{\frac{1}{2\theta}} \ln(2\beta_{ds}) \Omega_1^{1-\frac{1}{\theta}} \tag{29}$$

then $K_m = \lceil \tilde{K}_m \rceil$ where K_m is defined on Line 9 of Algorithm 2. If $\theta < 1$ the total number of subgradient evaluations is

$$\begin{aligned} K_1 + K_2 + \cdots + K_M &= \lceil \tilde{K}_1 \rceil + \lceil \tilde{K}_2 \rceil + \cdots + \lceil \tilde{K}_M \rceil \\ &< \tilde{K}_1 + \tilde{K}_2 + \cdots + \tilde{K}_M + M \\ &= \tilde{K}_1 \left(1 + \beta_{ds}^{\frac{1}{\theta}-1} + \left(\beta_{ds}^{\frac{1}{\theta}-1} \right)^2 + \cdots + \left(\beta_{ds}^{\frac{1}{\theta-1}} \right)^{M-1} \right) + M \\ &= \tilde{K}_1 \frac{\left(\beta_{ds}^{\frac{1}{\theta}-1} \right)^M - 1}{\beta_{ds}^{\frac{1}{\theta}-1} - 1} + M \\ &\leq \tilde{K}_1 \frac{\left(\beta_{ds}^{\frac{1}{\theta}-1} \right)^M}{\beta_{ds}^{\frac{1}{\theta}-1} - 1} + M. \end{aligned} \tag{30}$$

Now since

$$M \leq \frac{\ln \frac{\Omega_1}{\epsilon}}{\ln \beta_{ds}} + 1 \tag{31}$$

it follows that

$$(\beta_{ds}^{\frac{1}{\theta}-1})^M \leq \beta_{ds}^{\frac{1}{\theta}-1} \left(\frac{\Omega_1}{\epsilon} \right)^{\frac{1}{\theta}-1}. \tag{32}$$

Finally, substitute (31), (32), and the expression for $\tilde{K}_1$ into (30) to obtain the iteration complexity

$$\frac{\theta \beta_{ds}^{\frac{3}{2\theta}-1} \ln(2\beta_{ds})}{\beta_{ds}^{\frac{1}{\theta}-1} - 1} \kappa^2 \epsilon^{1-\frac{1}{\theta}} + \frac{\ln \frac{\Omega_1}{\epsilon}}{\ln \beta_{ds}} + 1 \tag{33}$$

total subgradient evaluations, which is (26).

Now onto (27). We derive the limiting behavior of (33) as $\epsilon \to 0$, and Ω_1 and κ approach ∞. In order to do this, we will prove that if $\Omega_1 = O(\kappa^{\frac{2\theta}{1-\theta}})$, the requisite lower bound on β_{ds} in (23) is $O(1)$, which implies that β_{ds} can be chosen as an $O(1)$ constant. If κ is too small, then it is enlarged to size $\Theta(\Omega^{\frac{1-\theta}{2\theta}})$ so that this does hold.

Considering each argument in the max in (23), the first is

$$\frac{1}{2} \left(\frac{\kappa^2}{4} \right)^{\frac{\theta}{\theta-1}} \Omega_1 = O\left(\kappa^{\frac{2\theta}{\theta-1}} \Omega_1 \right) = O(1)$$

and the second is

$$\theta^{-2\theta} \kappa^{-4\theta} \Omega_1^{2(1-\theta)} = O\left(\kappa^{-4\theta} \Omega_1^{2(1-\theta)} \right) = O(1)$$

where we have used the assumption that $\Omega_1 = O(\kappa^{\frac{2\theta}{1-\theta}})$. Since β_{ds} is $O(1)$ under this assumption, (33) implies the number of subgradient evaluations behaves as $O(\kappa^2\epsilon^{1-\frac{1}{\theta}})$. Since κ may have to be enlargened to $\Theta(\Omega^{\frac{1-\theta}{2\theta}})$, this implies the subgradient evaluations actually behave as $O(\max\{\kappa^2, \Omega^{\frac{1-\theta}{\theta}}\}\epsilon^{1-\frac{1}{\theta}})$, which yields (27). □

6.2 Discussion

The optimal choice for β_{ds} can be found by minimizing the iteration complexities given in (24) and (26) w.r.t. β_{ds}. However the closed form expression is complicated and not particularly enlightening. Solving it numerically, we find it is typically between 2 and 2.5.

Regarding RSG [47], the iteration complexity is very similar to ours, even though the analysis is different. There are several points to note in comparing the two. First is that their error metric is $h(x)-h^*$. On the other hand our error metric is $d(x_k, \mathcal{X}_h)^2$. Furthermore their iteration complexity is for finding $h(x)-h^* \leq 2\epsilon$. To do a fair comparison, we can convert their error metric to $d(x_k, \mathcal{X}_h)^2$ by using $\epsilon' = 2^{-1}\epsilon^{\frac{1}{2\theta}}$ in their iteration complexity. As we mentioned earlier, their iteration complexity is $O(\epsilon'^{2(\theta-1)} \ln \frac{1}{\epsilon'})$. Thus, if we make the substitution, we see that their iteration complexity is the same as ours except they have an extra $\log \frac{1}{\epsilon}$ term. The dependence on $\kappa = G/c$ is the same.

With respect to their algorithm implementation as given in [47, Algorithm 2], the major difference to DS-SG is that [47] requires averaging to be done after every inner loop. As mention before, this may be undesirable on problems where nonergodic methods are preferable. For instance, in problems where $\mathcal{C}$ enforces sparsity or low-rank, the averaging phase spoils this property [13]. Another situation in which averaging is undesirable is when learning with reproducing kernels [24]. In such problems, the variable is represented as a linear combination of a kernel evaluated at different points. After t iterations of the subgradient method, the solution is $\sum_{i=1}^{t-1} \alpha_i k(x_i, \cdot)$ where $k : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is the kernel function. Thus it is necessary to store the $t-1$ points $\{x_i\}$ after t iterations which is infeasible. The key to making the method practical is that for certain objectives the coefficients α_i decay geometrically and the early iterations can be safely ignored. Thus only a small fraction of the last t points are recorded. However, if averaging is used, the earlier coefficients are no longer negligible which compromises the feasibility of the method. Another advantage of our approach over [47] will arise in the next section, where we develop a method for adapting to unknown c.

7 Double descending stairs stepsize method for unknown c

7.1 The method

In our method DS-SG (Algorithm 2), the initial number of inner iterations is

$$K_1 = \left\lceil \theta\kappa^2\beta_{ds}^{\frac{1}{2\theta}} \ln(2\beta_{ds})\, \Omega_1^{1-\frac{1}{\theta}} \right\rceil, \tag{34}$$

where $\kappa = G/c$. The initial stepsize $\alpha(1)$, given in line 4, and the lower bound on β_{ds}, given in (23), also depend on c. If a lower bound for c is known, then using this value in (4), (23), and (34) ensures convergence. However in many problems c is unknown. Furthermore if c is greatly underestimated then this will lead to many more inner iterations and a much smaller initial stepsize than is necessary. For the case where no accurate lower bound for c is known, we propose the following "doubling trick" which still guarantees essentially the same iteration complexity. The analysis only holds when $\mathcal{C}$ is bounded. Let the diameter of $\mathcal{C}$ be $\Omega_{\mathcal{C}} = \max_{x,x' \in \mathcal{C}} \|x - x'\|^2$. The basic idea is to repeat DS-SG with a new c which is half the old estimate, which quadruples the number of inner iterations and halves the initial stepsize. In this way it takes only $O(\log_2(\frac{c_1}{c}))$ trial choices for for the error bound constant until it lower bounds the true constant. Furthermore, if the initial estimate c_1 is much larger than the true c, then the number of inner iterations is relatively small, which is why the overall iteration complexity comes out to be only a factor of (4/3) times larger than that of DS-SG. This means it is advantageous to use a large overestimate of c. In fact one can safely use the initial estimate $c_1 = G\Omega_{\mathcal{C}}^{\frac{1}{2} - \frac{1}{2\theta}}$. We call the method the "Doubling trick Descending Stairs" subgradient method (DS2-SG).

Algorithm 3: Double Descending Stairs subgradient method for unknown c (DS2-SG), $\frac{1}{2} \leq \theta \leq 1$

Require: $\beta_{ds}, G, M, c_1, \Omega_{\mathcal{C}}, x_1$, *stopping criterion*.
1: $l = 1$
2: **while** *stopping criterion* not satisfied **do**
3: $\tilde{x}_l$ =DS-SG($\beta_{ds}, M, \tilde{x}_{l-1}, \Omega_{\mathcal{C}}, G, c_l, \theta$)
4: $c_{l+1} = c_l/2$
5: $l = l + 1$
6: **end while**
7: **return** $\tilde{x}_{l-1}$

Theorem 4 *Suppose Assumption 3 holds and* $1/2 \leq \theta \leq 1$. *Suppose* $\|x - y\|^2 \leq \Omega_{\mathcal{C}}$ *for all* $x, y \in \mathcal{C}$ *and fix* $c_1 > 0$. *Let* $\kappa_1 = G/c_1$. *If* $\theta < 1$, *choose* $\beta_{ds} > 1$ *s.t.*

$$\beta_{ds} \geq \max \left\{ \frac{1}{2} \left(\frac{\kappa_1^2}{4} \right)^{\frac{\theta}{\theta - 1}} \Omega_{\mathcal{C}}, \theta^{-2\theta} \kappa_1^{-4\theta} \Omega_{\mathcal{C}}^{2(1-\theta)} \right\}. \tag{35}$$

If $\theta = 1$, *choose* c_1 *so that* $\kappa_1 \geq 2$ *and choose any* $\beta_{ds} > 1$. *Fix* $\epsilon > 0$ *and choose*

$$M \geq \left\lceil \frac{\ln \frac{\Omega_{\mathcal{C}}}{\epsilon}}{\ln \beta_{ds}} \right\rceil. \tag{36}$$

For the output of Algorithm DS2-SG, if $l \geq L = \max\{0, \lceil \log_2 c_1/c \rceil\} + 1$, *then* $d(\tilde{x}_l, \mathcal{X}_h)^2 \leq \epsilon$. *The number of subgradient evaluations is upper bounded by the following quantities (where* $\overline{\kappa} = \max\{\kappa, \kappa_1\}$*):*

1. *If* $\theta = 1$*:*

$$\frac{4}{3}\left(\beta_{ds}^{\frac{1}{2}}\overline{\kappa}^2\ln(2\beta_{ds}) + \left(\frac{\overline{\kappa}}{\kappa_1}\right)^2 + \log_2\left(\frac{\overline{\kappa}}{\kappa_1}\right) + 1\right)\left(\frac{\ln\frac{\Omega_{\mathcal{C}}}{\epsilon}}{\ln\beta_{ds}} + 1\right) \tag{37}$$

which simplifies to

$$O\left(\overline{\kappa}^2\ln\frac{\Omega_{\mathcal{C}}}{\epsilon}\right)$$

as $\kappa, \kappa_1, \Omega_{\mathcal{C}} \to \infty$*, and* $\epsilon \to 0$*.*

2. *If* $\frac{1}{2} \leq \theta < 1$*:*

$$\frac{4\theta\beta_{ds}^{\frac{3}{2\theta}-1}\ln(2\beta_{ds})}{3(\beta_{ds}^{\frac{1}{\theta}-1}-1)}\overline{\kappa}^2\epsilon^{1-\frac{1}{\theta}} + \left(\frac{4\overline{\kappa}^2}{3\kappa_1^2} + \log_2\left(\frac{\overline{\kappa}}{\kappa_1}\right) + 2\right)\left(\frac{\ln\frac{\Omega_{\mathcal{C}}}{\epsilon}}{\ln\beta_{ds}} + 1\right). \tag{38}$$

If κ_1 *is chosen large enough so that* $\Omega_{\mathcal{C}} = O(\kappa_1^{\frac{2\theta}{1-\theta}})$*, this simplifies to*

$$O\left(\max\{\overline{\kappa}^2, \Omega_{\mathcal{C}}^{\frac{1}{\theta}-1}\}\epsilon^{1-\frac{1}{\theta}}\right) \tag{39}$$

as $\kappa, \kappa_1, \Omega_{\mathcal{C}} \to \infty$*, and* $\epsilon \to 0$*.*

Note that if $c_1 = G\Omega_{\mathcal{C}}^{\frac{1}{2}-\frac{1}{2\theta}}$*, then* $\overline{\kappa} = \kappa$*.*

Proof For all l it is clear that since the iterates remain in the constraint set $\mathcal{C}$, $d(\tilde{x}_l, \mathcal{X}_h)^2 \leq \Omega_{\mathcal{C}}$. Now by the choice of L, $c_l \leq c$ for all $l \geq L$. Therefore we can apply Theorem 3 to the iterations within the while loop when $l \geq L$, which implies $d(\tilde{x}_l, \mathcal{X}_h)^2 \leq \epsilon$ for $l \geq L$. Note that, since the R.H.S. of (35) decreases if you replace c_1 with a smaller error bound constant, β_{ds} will satisfy (23) with c_l in place of c_1 for all $l \geq 2$.

We now determine the overall iteration complexity. Let K_j^l for $l = 1, 2, \ldots, L$ and $j = 1, 2, \ldots M$ be the number of iterations passed to FixedSG within the jth call to FixedSG in DS-SG, during the lth loop in DS2-SG. For all such l and j, $K_j^l = \lceil \tilde{K}_j^l \rceil$ where $\tilde{K}_j^1 = \tilde{K}_j$ defined in (29), and $\tilde{K}_j^l = 4^{l-1}\tilde{K}_j^1$. Thus using the fact that $\lceil x \rceil < x + 1$, the total number of subgradient calls of DS2-SG can be upper bounded as

$$\begin{aligned}\sum_{l=1}^{L}\sum_{j=1}^{M} K_j^l &< \sum_{l=1}^{L}\sum_{j=1}^{M}\tilde{K}_j^l + LM \\ &= \left(1 + 4 + 16 + \cdots + 4^{L-1}\right)\sum_{j=1}^{M}\tilde{K}_j^1 + LM\end{aligned}$$

$$
\begin{aligned}
&= \frac{(4^L - 1)}{3} \sum_{j=1}^{M} \tilde{K}_j^1 + LM \\
&= \frac{4}{3} \max\left\{ \left(\frac{c_1}{c}\right)^2, 1 \right\} \sum_{j=1}^{M} \tilde{K}_j^1 \\
&\quad + (\max\{0, \lceil \log_2 c_1/c \rceil\} + 1) \left\lceil \frac{\ln \frac{\Omega_{\mathcal{C}}}{\epsilon}}{\ln \beta_{ds}} \right\rceil. \qquad (40)
\end{aligned}
$$

By noting that

$$
\max\left\{ \left(\frac{c_1}{c}\right)^2, 1 \right\} = \frac{\overline{\kappa}^2}{\kappa_1^2}
$$

and with the aid of (24) and (26), (40) reduces to the iteration complexities given in (37) and (38).

Now

$$
cd(x, \mathcal{X}_h)^{\frac{1}{\theta}} \leq h(x) - h^* \leq \langle g, x - x^* \rangle \leq \|g\| \|x - x^*\|
$$

for all $x \in \mathcal{C}$, $g \in \partial h(x)$, and $x^* \in \mathcal{X}_h$. If $x^* = \text{proj}_{\mathcal{X}_h}(x)$ then this implies

$$
cd(x, \mathcal{H}_h)^{\frac{1}{\theta}} \leq Gd(x, \mathcal{H}_h) \implies c \leq Gd(x, \mathcal{H}_h)^{1-\frac{1}{\theta}} \quad \forall x \in \mathcal{C}.
$$

Minimizing the R.H.S. yields $c \leq G\Omega_{\mathcal{C}}^{\frac{1}{2}-\frac{1}{2\theta}}$. Therefore the choice $c_1 = G\Omega_{\mathcal{C}}^{\frac{1}{2}-\frac{1}{2\theta}}$ guarantees $\kappa_1 \leq \kappa$. For $\theta = 1$, choosing $c_1 = G$ implies $\kappa_1 = 1$, which violates the requirement: $\kappa_1 \geq 2$. Thus one should instead choose $c_1 = G/2$. □

7.2 Discussion

The authors of RSG [47] proposed a variant, R^{2}SG, which can adapt to unknown c when $\theta < 1$. It also uses geometrically increasing number of inner iterations, however the initial stepsize remains the same. An advantage of that method is it does not require the constraint set to be bounded. However since their analysis is only valid for $\theta < 1$, it cannot be directly applied to important problems such as polyhedral convex optimization, and requires using a surrogate $\theta < 1$.

For $\theta = 1$, the subgradient methods of [40, Sec. 2.3] and [18] choose geometrically decaying stepsizes which depend on the error bound constant c. It is plausible that our "doubling trick" idea can be employed to accelerate these methods when c is unknown, by starting with an estimate for c and repeatedly halving it. This should lead to linear convergence with only a slightly larger iteration complexity than the original methods. Thus our doubling trick can be thought of as a "meta-acceleration" technique with potentially large scope.

A drawback of DS2-SG is it does not have an explicit stopping rule. In particular, the number of "wrapper" iterations, L, depends on the true error bound constant c, which is unknown. This is also the main drawback of R^{2}SG [47] (along with the fact it cannot be applied when $\theta = 1$). As was suggested in [47], we suggest using an independent stopping criterion. For example on a machine learning problem, one could use the error on a small validation set as an indication the algorithm has converged. If a lower bound $h_{LB} \leq h^*$ is known, then $c^{-\theta}\,(h(x_k) - h_{LB})^{\theta} < \sqrt{\epsilon}$ can be used as a stopping criterion. This is because $d(x_k, \mathcal{X}_h) \leq c^{-\theta}\,(h(x_k) - h_{LB})^{\theta}$. Furthermore since, $cd(x_k, \mathcal{X}_h)^{\frac{1}{\theta}-1} \leq \|g\|$ for $g \in \partial h(x)$, the norm of the subgradient could be used as a stopping criterion for $\theta < 1$. Another possibility is to use the fact that $cd(x_k, \mathcal{X}_h) \leq \|g\|^{\theta}\Omega_{\mathcal{C}}^{\theta}$. Exploring these stopping criteria is a topic for future work.

In practice for DS2-SG, we often observe an increase in the objective function value whenever a new trial error bound constant is used resulting in a larger stepsize. It is therefore a good strategy to keep track of the iterate $\tilde{x}_l$ with the smallest objective function value so far. This does not change the overall iteration complexity and only requires storing one additional iterate.

8 Faster rates for decaying stepsizes for $\theta < 1$

If $\theta < 1$, an upper bound for G is known, a lower bound for c is known, and the constraint set is compact, then it is possible to obtain the same iteration complexity as DS-SG using decaying stepsizes. We consider $\theta \geq 1/2$ and $\theta < 1/2$ in separate theorems.

Theorem 5 *Suppose Assumption 3 holds and $\frac{1}{2} \leq \theta < 1$. Suppose $\|x - y\|^2 \leq \Omega_{\mathcal{C}}$ for all $x, y \in \mathcal{C}$. Choose c small enough (or G large enough) so that*

$$\kappa \geq \sqrt{3}\Omega_{\mathcal{C}}^{\frac{1-\theta}{2\theta}}. \tag{41}$$

For the iterates of the subgradient method (2), let $\alpha_k = \alpha_1 k^{-p}$ where

$$p = \frac{1}{2(1-\theta)} \tag{42}$$

and

$$\alpha_1 = \frac{c}{G^2}\left(\frac{\theta\kappa^2}{1-\theta}\right)^{p}. \tag{43}$$

Then, for all $k \geq \lceil \frac{2\theta}{1-\theta} \rceil$

$$d(x_k, \mathcal{X}_h)^2 \leq \left(\frac{\theta}{1-\theta}\right)^{\frac{\theta}{1-\theta}}\left(\frac{k}{\kappa^2}\right)^{\frac{\theta}{\theta-1}}. \tag{44}$$

Proof The recursion describing the subgradient method is, for $k \geq 1$,

$$e_{k+1} \leq e_k - 2\alpha_k c e_k^{\gamma} + \alpha_k^2 G^2, \tag{45}$$

where $e_k = d(x_k, \mathcal{X})^2$ and $\gamma = \frac{1}{2\theta}$. Let $\alpha_k = \alpha_1 k^{-p}$. We wish to prove that if

$$p = \frac{\gamma}{2\gamma - 1}$$

and the constant α_1 is chosen as in (43), then

$$e_k \leq C_e k^{-b} \tag{46}$$

where

$$b \triangleq \frac{p}{\gamma} = \frac{1}{2\gamma - 1},$$

for all $k \geq k_0 \triangleq \lceil 2b \rceil$, and C_e is given by $C_e = (\kappa^2 b)^b$.

We will prove this result by induction. The initial condition is

$$e_{k_0} \leq C_e k_0^{-b}$$

which is implied by

$$\Omega_{\mathcal{C}} \leq C_e k_0^{-b} \iff C_e = (\kappa^2 b)^b \geq \Omega_{\mathcal{C}} k_0^b. \tag{47}$$

Since $k_0 = \lceil 2b \rceil \leq 2b + 1 \leq 3b$, this is implied by

$$\left(b\kappa^2\right)^b \geq \Omega_{\mathcal{C}} (3b)^b.$$

Dividing by b^b and taking the bth root yields

$$\kappa^2 \geq 3\Omega_{\mathcal{C}}^{\frac{1}{b}},$$

which is (41).

Next, assume (46) is true for some $k \geq k_0$. That is, assume $e_k = a C_e k^{-b}$, where $0 \leq a \leq 1$. We will show that this implies $e_{k+1} \leq C_e (k+1)^{-b}$. Substituting $e_k = a C_e k^{-b}$ and $\alpha_k = \alpha_1 k^{-p}$ into the right hand side of (45) yields

$$\begin{aligned} e_{k+1} &\leq a C_e k^{-b} - 2\alpha_1 c a^{\gamma} C_e^{\gamma} k^{-(p+\gamma b)} + \alpha_1^2 G^2 k^{-2p} \\ &= a C_e k^{-b} + \left(\alpha_1^2 G^2 - 2\alpha_1 c a^{\gamma} C_e^{\gamma}\right) k^{-2p} \end{aligned}$$

using the fact that $p + \gamma b = 2p$. Thus we wish to enforce the inequality:

$$aC_e k^{-b} + \left(\alpha_1^2 G^2 - 2\alpha_1 c a^\gamma C_e^\gamma\right) k^{-2p} \leq C_e (k+1)^{-b}. \tag{48}$$

We need (48) to hold for all $a \in [0, 1]$. Since $\frac{1}{2} \leq \theta < 1$ which implies $\frac{1}{2} < \gamma \leq 1$, the L.H.S. is a convex function of a for $a \geq 0$. Therefore if the inequality holds for $a = 0$ and $a = 1$, then it holds for all $a \in [0, 1]$.

Consider first, $a = 0$. The condition is

$$\alpha_1^2 G^2 k^{-2\gamma b} \leq C_e (k+1)^{-b}.$$

This is equivalent to

$$\alpha_1 \leq G^{-1} C_e^{\frac{1}{2}} k^{\gamma b} (k+1)^{-\frac{b}{2}}. \tag{49}$$

Note that α_1, given in (43), can be rewritten as

$$\alpha_1 = \frac{c C_e^\gamma}{G^2}.$$

Substituting α_1 into (49) yields

$$\frac{c}{G^2} C_e^\gamma \leq G^{-1} C_e^{\frac{1}{2}} k^{\gamma b} (k+1)^{-\frac{b}{2}}$$

which can be rearranged to

$$G \geq c C_e^{\gamma - \frac{1}{2}} k^{-\gamma b} (k+1)^{\frac{b}{2}}. \tag{50}$$

Now

$$C_e^{\frac{2\gamma - 1}{2}} = \kappa \sqrt{b}.$$

Substituting this into (50) yields

$$k^{\gamma b} (k+1)^{-\frac{b}{2}} \geq \sqrt{b}. \tag{51}$$

Now

$$\begin{aligned}(k+1)^{-\frac{b}{2}} &= k^{-\frac{b}{2}} (1 + k^{-1})^{-\frac{b}{2}} \\ &\geq k^{-\frac{b}{2}} \left(1 - \frac{b}{2} k^{-1}\right) \\ &= k^{-\frac{b}{2}} - \frac{b}{2} k^{-\frac{b}{2} - 1}.\end{aligned}$$

Therefore (51) is implied by

$$k^{b(\gamma-\frac{1}{2})} - \frac{b}{2}k^{b(\gamma-\frac{1}{2})-1} \geq \sqrt{b}.$$

Now substituting $b = (2\gamma - 1)^{-1}$ into the two exponents yields

$$k^{\frac{1}{2}} - \frac{b}{2}k^{-\frac{1}{2}} \geq \sqrt{b}$$

which is equivalent to

$$t^2 - \sqrt{b}t - \frac{b}{2} \geq 0$$

with the substitution $t = \sqrt{k}$. Thus we require

$$t \geq \frac{1+\sqrt{3}}{2}\sqrt{b}$$

which is implied by $k \geq 2b$. Thus $k \geq \lceil 2b \rceil$ implies (48) holds with $a = 0$.

Now consider $a = 1$ in (48). We again simplify (48) using

$$C_e(k+1)^{-b} = C_e k^{-b}(1+k^{-1})^{-b} \geq C_e k^{-b} - bC_e k^{-(b+1)}.$$

Therefore in the case $a = 1$, (48) is implied by

$$\left(\alpha_1^2 G^2 - 2\alpha_1 c C_e^{\gamma}\right) k^{-2p} \leq -bC_e k^{-(b+1)}. \tag{52}$$

Now $2p = b + 1$, therefore (52) is equivalent to

$$\alpha_1^2 G^2 - 2\alpha_1 c C_e^{\gamma} + bC_e \leq 0$$

for all $k \geq 1$. The L.H.S. is a positive-definite quadratic in α_1. Solving it yields the two solutions

$$\frac{2cC_e^{\gamma} \pm \sqrt{4c^2 C_e^{2\gamma} - 4G^2 bC_e}}{2G^2}.$$

The quadratic has a real solution if

$$4c^2 C_e^{2\gamma} - 4G^2 bC_e \geq 0 \iff C_e \geq \left(\kappa^2 b\right)^b. \tag{53}$$

Thus since $C_e = (\kappa^2 b)^b$, the only valid choice for α_1 is

$$\alpha_1 = \frac{cC_e^{\gamma}}{G^2}$$

which corresponds to (43). This completes the proof. □

The convergence rate given in (44) yields the following iteration complexity: The subgradient method with this stepsize yields a point such that $d(x_k, \mathcal{X}_h)^2 \leq \epsilon$ for all

$$k \geq \frac{2\theta}{1-\theta} \max\left\{\kappa^2, 3\Omega_{\mathcal{C}}^{\frac{1}{\theta}-1}\right\} \epsilon^{1-\frac{1}{\theta}}.$$

This is equal (up to constants) to the iteration complexity derived for DS-SG in Theorem 3. The main drawback versus DS-SG is that the analysis only holds for a bounded constraint set. It is also trivial to embed this stepsize into the "doubling" framework used in DS2-SG so that one does not need a lower bound for c. Since the analysis is the same as given in Theorem 4, we omit the details. The proof of Theorem 5 is inspired by [18] which considered geometrically decaying stepsizes when $\theta = 1$. Theorem 5 is a natural extension of [18] to $\theta < 1$.

The optimal stepsize given in Theorem 5 requires knowledge of G, c, and $\Omega_{\mathcal{C}}$ in order to set α_1. In the longer version of this paper [22] we show that the stepsizes $\alpha_k = \alpha_1 k^{-p}$ with $p < 1$ are convergent for any $\alpha_1 > 0$ when $\theta \geq 1/2$.

We can obtain the same rate for the choice of α_1 and p in Theorem 5 when $\theta < 1/2$. In this case, the convergence rate holds for all $k \geq 2$ under a slightly different condition on κ.

Theorem 6 *Suppose Assumption 3 holds and $0 < \theta < \frac{1}{2}$. Suppose $\|x - y\|^2 \leq \Omega_{\mathcal{C}}$ for all $x, y \in \mathcal{C}$. Choose c small enough (or G large enough) so that*

$$\kappa^2 \geq \frac{2(1-\theta)}{\theta} \Omega_{\mathcal{C}}^{\frac{1-\theta}{\theta}}. \tag{54}$$

For the iterates of the subgradient method (2), let $\alpha_k = \alpha_1 k^{-p}$ where p and α_1 are defined in (42) and (43). Then, for all $k \geq 2$, $d(x_k, \mathcal{X})^2$ satisfies (44).

Proof Recall $\gamma = 1/(2\theta)$ and note that $\gamma > 1$ since $\theta < 1/2$. Recall

$$b = \frac{1}{2\gamma - 1} \leq 1 \text{ and } p = \gamma b.$$

As with the proof of Theorem 5, this will be a proof by induction. We wish to prove that $e_k \leq C_e k^{-b}$ for all $k \geq 2$ for the constant defined as $C_e = \left(\kappa^2 b\right)^b$. The initial condition is $e_2 \leq C_e 2^{-b}$ which is implied by $C_e \geq \Omega_{\mathcal{C}} 2^b$. This in turn is implied by (54).

Now we assume $e_k = a C_e k^{-b}$ for some $k \geq 2$ and $a \in [0, 1]$ and will show that $e_{k+1} \leq C_e (k+1)^{-b}$. Using the inductive assumption in the main recursion (45) yields the following inequality, which we would like to enforce for all $a \in [0, 1]$:

$$\begin{aligned} e_{k+1} &\leq a C_e k^{-b} + \left(\alpha_1^2 G^2 - 2\alpha_1 c a^\gamma C_e^\gamma\right) k^{-2p} \\ &\leq C_e (k+1)^{-b}, \end{aligned} \tag{55}$$

where we once again used the fact that $p + \gamma b = 2p$. We require (55) to hold for all $a \in [0, 1]$. The L.H.S. is concave in a (since $\gamma > 1$), so we will compute the maximizer w.r.t. a. Let $D_1 = \alpha_1^2 G^2 k^{-2p}$, $D_2 = C_e k^{-b}$, and $D_3 = 2\alpha_1 c C_e^{\gamma} k^{-2\gamma b}$. Then let

$$f(a) = D_1 + D_2 a - D_3 a^{\gamma}$$

which is the L.H.S. of (55). Let a_* be the solution to

$$0 = f'(a_*) = D_2 - \gamma D_3 a_*^{\gamma 1},$$

which implies

$$\begin{aligned} a_* &= \left(\frac{D_2}{\gamma D_3}\right)^{\frac{1}{\gamma-1}} \\ &= C_e^{-1}(2\alpha_1 \gamma c)^{\frac{1}{1-\gamma}} k^{\frac{1}{\gamma-1}} = C_e^{-1} D_4 \alpha_1^{\frac{1}{1-\gamma}} k^{\frac{1}{\gamma-1}} \end{aligned}$$

where $D_4 = (2\gamma c)^{\frac{1}{1-\gamma}}$. But recall that $a \in [0, 1]$ therefore the maximizer of $f(a)$ in $[0, 1]$ is given by

$$\min\left\{1, C_e^{-1} D_4 \alpha_1^{\frac{1}{1-\gamma}} k^{\frac{1}{\gamma-1}}\right\}.$$

Thus if

$$k \geq \left(C_e D_4^{-1}\right)^{\gamma-1} \alpha_1 \tag{56}$$

then the maximizer in $[0, 1]$ is equal to 1. Substituting the values for α_1 and C_e into (56) yields

$$k \geq \left(C_e D_4^{-1}\right)^{\gamma-1} \frac{c}{G^2} C_e^{\gamma} = \frac{2\gamma}{2\gamma - 1}.$$

Since $\gamma > 1$ this is implied by $k \geq 2$. Thus we only need to consider $a = 1$ in (55).

The analysis with $a = 1$ substituted into (55) was carried out in the proof of Theorem 5. Recall that for this choice of stepsize and constant, the inequality (55) is satisfied with $a = 1$ for all $k \geq 1$, which completes the proof. □

9 Numerical experiments

In this section we present two numerical experiments to demonstrate some of the theoretical findings in this manuscript. For more experiments, see the long version of this paper [22].

9.1 Least absolute deviations

We consider an example satisfying HEB(c, θ) with $\theta = 1$. Consider the following problem:

$$\min_x \|Ex - b\|_1 : \quad \|x\|_1 \leq \tau. \tag{57}$$

This objective function is used in regression problems and in machine learning [16, 20,43,44]. The ℓ_1 constraint is used to encourage a sparse solution x. Besides the subgradient techniques considered in this manuscript, there are a few other methods which can tackle Prob. (57) such as ADMM [8], primal dual splitting [11], projective splitting [21], and two specialized algorithms [44,45]. In this experiment, we will only consider subgradient methods, which have considerable advantages over these alternatives in terms of scalability and theoretical guarantees.[2]

Problem (57) is a polyhedral optimization problem therefore HEB(c, θ) is satisfied for all x with $\theta = 1$ [47]. However, it is not easy to compute c. Projection onto the ℓ_1 ball can be done in linear time by using the method of [29] among others.

To test the subgradient methods we consider a synthetic instance of Problem (57). We set $m = 100$ and $n = 50$ and construct E of size $m \times n$ with i.i.d. $\mathcal{N}(0, 1)$ entries. We construct b of size $m \times 1$ with i.i.d. $\mathcal{N}(0, 1)$ entries. We set $\tau = 1$. All tested algorithms were initialized to the same point. First we test the performance of the descending stairs stepsize in DS-SG, the restarted subgradient method RSG of [47], and Shor's method of [40, Sec. 2.3] (which is very similar to Goffin's stepsize [18]), alongside two decaying stepsizes: $\alpha_k = 0.1k^{-1}$ and $\alpha_k = 0.01k^{-0.5}$, where the constant α_1 was tuned for fast convergence. For DS-SG we used $\beta_{ds} = 4$, $\epsilon = 10^{-5}$, $\Omega_{\mathcal{C}} = 4\tau^2$, and $G = \sigma_1(E)\sqrt{m}$. For the other methods we chose the parameters in the way suggested by the authors. Since c is difficult to estimate, we tuned it to get the best performance in each algorithm (see below for our approach, DS2-SG, which estimates c on the fly). For DS-SG, RSG, and Shor's algorithm, these were $c = 22$, 15, and 11 respectively.

The log of $d(x_k, \mathcal{X}_h)^2$ for each of these algorithms is plotted in Fig. 1 (Left) versus the number k of subgradient evaluations. Figure 1 (Left) confirms that DS-SG has a linear convergence rate, verifying Theorem 3. Its performance is very similar to Shor's method. While RSG does appear to obtain linear convergence, its rate is slower than DS-SG and Shor's method.

As was mentioned, we had to tune c to get good performance of DS-SG, RSG, and Shor's method. We now compare these three methods with our proposed "doubling trick" variant, DS2-SG, which does not need the value of c. We also applied our "doubling trick" of Sect. 7 to Shor's method of [40, Sec. 2.3.] and call the method "Shor2". Shor2 works in much the same way as DS2-SG: we start with a candidate $c = G$ and run Shor's method for enough iterations to reach $\epsilon = 10^{-8}$ error in terms of distance to the solution, according to the theory of [40, Sec. 2.3]. We then halve the candidate error bound constant c and repeat.

[2] see [22] for a more detailed comparison with these alternative methods.

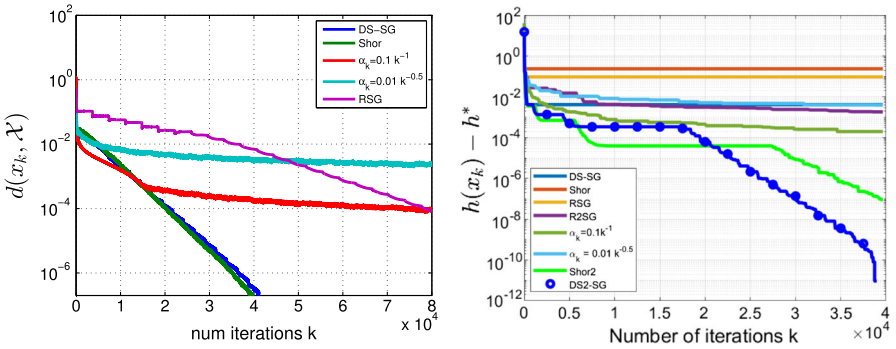

Fig. 1 Problem (57). (Left): Log of square distance to the solution versus number of subgradient evaluations for DS-SG, RSG, Shor's method, and two decaying stepsizes. (Right): Log of $h(x) - h^*$ versus number of subgradient evaluations for DS-SG, RSG, Shor's method, R^{2}SG, DS2-SG, Shor2, which is Shor's method with our doubling trick, and two decaying polynomial stepsizes

We also compare with the method R^{2}SG proposed in [47]. Note that this method only works for $\theta < 1$ so following the advice of [47], we use the approximate value of $\hat{\theta} = 0.8$. We initialize DS2-SG with the same parameters as DS-SG but with $c_1 = G/2 = 80$. To demonstrate the effect of poorly chosen c in DS-SG, RSG, and Shor's method (without doubling trick), we set $c = 100$ for all these methods (recall the tuned values were smaller). The results are given in Fig. 1 (Right). For each algorithm we keep track of the iterate with the smallest function value encountered so far. We see that DS-SG, RSG, and Shor's method converge to suboptimal solutions due to the incorrect value of c. However DS2-SG and Shor2 find the correct solution to within an objective function error of 10^{-10}. R^{2}SG has slower convergence, which is not surprising since it is not guaranteed to obtain linear convergence when $\theta = 1$. It is also encouraging that both DS2-SG and Shor2 are faster than the decaying stepsize choices, $\alpha_k = O(k^{-1})$ and $\alpha_k = O(k^{-0.5})$, since these choices are commonly used in practice.

9.2 Sparse SVM

The ℓ_1-regularized Support Vector Machine (SVM) Problem [51] is

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \max\left\{0, 1 - y_i c_i^\top x\right\} + \rho \|x\|_1$$

for a dataset $\{c_i, y_i\}_{i=1}^m$ with $c_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$. We will consider the equivalent constrained version

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \max\left\{0, 1 - y_i c_i^\top x\right\} : \|x\|_1 \leq \tau. \tag{58}$$

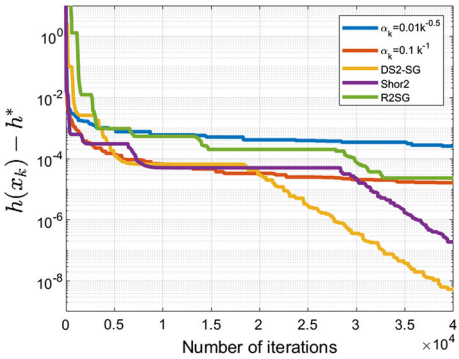

Fig. 2 Problem (58) with randomly generated data: Log of $h(x) - h^*$ versus number of subgradient evaluations for DS2-SG, R^{2}SG, Shor2, and two decaying stepsizes

Since the objective function is polyhedral it satisfies HEB with $\theta = 1$ for some unknown $c > 0$. Since c is unknown, we only consider the subgradient methods DS2-SG (ours), Shor2 (the method of [40, Sec. 2.3] with our doubling trick applied to it), R^{2}SG [47], and the following decaying stepsizes: $\alpha_k = 0.1k^{-1}$ and $\alpha_k = 0.01k^{-0.5}$, where the constants 0.1 and 0.01 were tuned to give fast convergence. R^{2}SG only works for $\theta < 1$ so cannot be directly applied to this problem. Instead we selected $\hat{\theta} < 1$ which gave the fastest convergence. Surprisingly, $\hat{\theta} = 0.5$ performed the best on this problem, even though one might expect $\hat{\theta} \approx 1$ to perform better. For DS2-SG and Shor2 we initialize with $c_1 = G/2$ where $G = \sum_{i=1}^{m} \|c_i\|$. We used $\beta_{ds} = 4$, $\epsilon = 10^{-8}$, and $\Omega_{\mathcal{C}} = 4\tau^2$. All algorithms had the same starting point.

A random instance of Prob. (58) was generated as follows: $n = 50$, $m = 100$, the entries of c_i are drawn from $\mathcal{N}(0, 1)$, the $y_i = \pm 1$ with equal probability, and $\tau = 2$. The results are plotted in Fig. 2. We see that our proposal, DS2-SG, and Shor2, which uses our doubling trick, outperform the others.

References

1. Agro, G.: Maximum likelihood and L_p norm estimators. Stat. Appl. **4**(1), 7 (1992)
2. Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. Math. Program. **137**(1–2), 91–129 (2013)
3. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, Berlin (2011)
4. Beck, A., Shtern, S.: Linearly convergent away-step conditional gradient for non-strongly convex functions. Math. Program. **164**, 1–27 (2015)
5. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Nashua (1999)
6. Bolte, J., Daniilidis, A., Lewis, A.: The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM J. Optim. **17**(4), 1205–1223 (2007)
7. Bolte, J., Nguyen, T.P., Peypouquet, J., Suter, B.W.: From error bounds to the complexity of first-order descent methods for convex functions. Math. Program. **165**, 1–37 (2015)
8. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
9. Burke, J., Deng, S.: Weak sharp minima revisited part i: basic theory. Control Cybern. **31**, 439–469 (2002)
10. Burke, J., Ferris, M.C.: Weak sharp minima in mathematical programming. SIAM J. Control Optim. **31**(5), 1340–1359 (1993)

11. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
12. Cruz, J.Y.B.: On proximal subgradient splitting method for minimizing the sum of two nonsmooth convex functions. Set-Valued Var. Anal. **25**(2), 245–263 (2017)
13. Davis, D., Yin, W.: A three-operator splitting scheme and its optimization applications. Set-Valued Var. Anal. **25**(4), 829–858 (2017)
14. Ferris, M.C.: Finite termination of the proximal point algorithm. Math. Program. **50**(1), 359–366 (1991)
15. Freund, R.M., Lu, H.: New computational guarantees for solving convex optimization problems with first order methods, via a function growth condition measure. Math. Program. **170**, 1–33 (2015)
16. Gao, X., Huang, J.: Asymptotic analysis of high-dimensional LAD regression with LASSO. Stat. Sin. **20**, 1485–1506 (2010)
17. Gilpin, A., Pena, J., Sandholm, T.: First-order algorithm with $O(\ln(1/\epsilon))$ convergence for ϵ-equilibrium in two-person zero-sum games. Math. Program. **133**(1–2), 279–298 (2012)
18. Goffin, J.L.: On convergence rates of subgradient optimization methods. Math. Program. **13**(1), 329–347 (1977)
19. Hare, W., Lewis, A.S.: Identifying active constraints via partial smoothness and prox-regularity. J. Convex Anal. **11**(2), 251–266 (2004)
20. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R.: The Elements of Statistical Learning. Springer, Berlin (2009)
21. Johnstone, P.R., Eckstein, J.: Projective splitting with forward steps: asynchronous and block-iterative operator splitting. arXiv:1803.07043 (2018)
22. Johnstone, P.R., Moulin, P.: Faster subgradient methods for functions with Hölderian growth. arXiv:1704.00196 (2017)
23. Karimi, H., Nutini, J., Schmidt, M.: Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 795–811. Springer (2016)
24. Kivinen, J., Smola, A.J., Williamson, R.C.: Online learning with kernels. IEEE Trans. Signal Process. **52**(8), 2165–2176 (2004)
25. Li, G.: Global error bounds for piecewise convex polynomials. Math. Program. **137**(1–2), 37–64 (2013)
26. Liang, J., Fadili, J., Peyré, G.: Activity identification and local linear convergence of forward–backward-type methods. SIAM J. Optim. **27**(1), 408–437 (2017)
27. Lim, E.: On the convergence rate for stochastic approximation in the nonsmooth setting. Math. Oper. Res. **36**(3), 527–537 (2011)
28. Luo, Z.Q., Tseng, P.: Error bounds and convergence analysis of feasible descent methods: a general approach. Ann. Oper. Res. **46**(1), 157–178 (1993)
29. Maculan, N., Santiago, C.P., Macambira, E., Jardim, M.: An $O(n)$ algorithm for projecting a vector on the intersection of a hyperplane and a box in $\mathbb{R}^n$. J. Optim. Theory Appl. **117**(3), 553–574 (2003)
30. Nedić, A., Bertsekas, D.: Convergence rate of incremental subgradient algorithms. In: Stochastic Optimization: Algorithms and Applications, pp. 223–264. Springer (2001)
31. Nedić, A., Bertsekas, D.P.: The effect of deterministic noise in subgradient methods. Math. Program. **125**(1), 75–99 (2010)
32. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIAM J. Optim. **19**(4), 1574–1609 (2009)
33. Noll, D.: Convergence of non-smooth descent methods using the Kurdyka–Łojasiewicz inequality. J. Optim. Theory Appl. **160**(2), 553–572 (2014)
34. Pang, J.S.: Error bounds in mathematical programming. Math. Program. **79**(1–3), 299–332 (1997)
35. Poljak, B.: Nonlinear programming methods in the presence of noise. Math. Program. **14**(1), 87–97 (1978)
36. Polyak, B.T.: Introduction to Optimization. Optimization Software Inc., New York (1987)
37. Renegar, J.: A framework for applying subgradient methods to conic optimization problems. arXiv:1503.02611 (2015)
38. Renegar, J.: "Efficient" subgradient methods for general convex optimization. SIAM J. Optim. **26**(4), 2649–2676 (2016)
39. Rosenberg, E.: A geometrically convergent subgradient optimization method for nonlinearly constrained convex programs. Math. Oper. Res. **13**(3), 512–523 (1988)
40. Shor, N.Z.: Minimization Methods for Non-differentiable Functions, vol. 3. Springer, Berlin (2012)

41. Supittayapornpong, S., Neely, M.J.: Staggered time average algorithm for stochastic non-smooth optimization with $O(1/T)$ convergence. arXiv:1607.02842 (2016)
42. Tseng, P.: Approximation accuracy, gradient methods, and error bound for structured convex optimization. Math. Program. **125**(2), 263–295 (2010)
43. Wang, L.: The ℓ_1 penalized LAD estimator for high dimensional linear regression. J. Multivar. Anal. **120**, 135–151 (2013)
44. Wang, L., Gordon, M.D., Zhu, J.: Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In: Sixth International Conference on Data Mining, ICDM'06, 2006, pp. 690–700. IEEE (2006)
45. Wu, T.T., Lange, K.: Coordinate descent algorithms for lasso penalized regression. Ann. Appl. Stat. **2**, 224–244 (2008)
46. Xu, Y., Lin, Q., Yang, T.: Accelerate stochastic subgradient method by leveraging local error bound. hyperimagehttp://arxiv.org/abs/1607.01027arXiv:1607.01027 (2016)
47. Yang, T., Lin, Q.: RSG: beating subgradient method without smoothness and strong convexity. arXiv:1512.03107 (2015)
48. Zhang, H.: New analysis of linear convergence of gradient-type methods via unifying error bound conditions. arXiv:1606.00269 (2016)
49. Zhang, H., Yin, W.: Gradient methods for convex minimization: better rates under weaker conditions. arXiv:1303.4645 (2013)
50. Zhou, Z., So, A.M.C.: A unified approach to error bounds for structured convex optimization problems. Math. Program. **165**, 689–728 (2017)
51. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm support vector machines. In: NIPS, vol. 15, pp. 49–56 (2003)