Open Journal of Mathematical Optimization

Nicholas J. A. Harvey, Chris Liaw & Sikander Randhawa

Tight analyses for subgradient descent I: Lower bounds

Volume 5 (2024), article no. 4 (17 pages)

https://doi.org/10.5802/ojmo.31

Article submitted on May 24, 2023, revised on May 30, 2024,
accepted on May 30, 2024.

Open Journal of Mathematical Optimization is a member of the
Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/
e-ISSN: 2777-5860

Tight analyses for subgradient descent I: Lower bounds

Nicholas J. A. Harvey
University of British Columbia
nickhar@mail.ubc.ca

Chris Liaw
Google Research
cvliaw@google.com

Sikander Randhawa
University of British Columbia
srand@cs.ubc.ca

Abstract

Consider the problem of minimizing functions that are Lipschitz and convex, but not necessarily differentiable. We construct a function from this class for which the T^{th} iterate of subgradient descent has error $\Omega(\log(T)/\sqrt{T})$. This matches a known upper bound of $O(\log(T)/\sqrt{T})$. We prove analogous results for functions that are additionally *strongly* convex. There exists such a function for which the error of the T^{th} iterate of subgradient descent has error $\Omega(\log(T)/T)$, matching a known upper bound of $O(\log(T)/T)$. These results resolve a question posed by Shamir (2012).

Digital Object Identifier 10.5802/ojmo.31

Keywords Gradient descent, First-order methods, Lower bounds.

Acknowledgments The first author is supported by an NSERC Discovery Grant.

1 Introduction

Subgradient descent (henceforth, GD) is a very simple and widely used iterative method for minimizing a non-smooth convex function. In a nutshell, the method works by querying an oracle for a subgradient, then taking a small step in the opposite direction. The simplicity and effectiveness of this algorithm has established it as an essential tool in numerous applications.

The efficiency of GD is usually measured by the rate of decrease of the *error* — the difference in value between the algorithm's output and the function's infimum. The optimal error rate is known under various assumptions on f, the function to be minimized. In addition to convexity, common assumptions are that f is *smooth* (the gradient is Lipschitz) or *strongly convex* (locally lower-bounded by a quadratic). In applications, strongly convex functions often arise due to regularization, whereas smooth functions can sometimes be obtained by smoothening approximations (e.g., convolution).

This paper focuses on the setting in which the function is non-smooth and Lipschitz, and the domain is convex and compact. A difficulty with this setting is that the successive iterates of GD might not have monotonically decreasing error. Consequently the final iterate might not have the lowest error. A workaround, known to Nemirovski and Yudin [10], is to output the *average* of the iterates. Existing analyses [10] show that after T iterations of GD, the error of the average is $\Theta(1/\sqrt{T})$, assuming that the function is Lipschitz and the step size is chosen appropriately. This error rate is optimal for first-order algorithms. For functions that are also strongly convex [5, 13] the average has error $O(\log(T)/T)$, although other algorithms [6] and averaging schemes [9, 13, 17] achieve error $\Theta(1/T)$. The latter error rate is also optimal for first-order algorithms.

Shamir [16] asked the very natural question of whether the *final* iterate of GD achieves the optimal rate in the non-smooth scenario, as it does in the smooth scenario. Substantial progress on this question was made by Shamir and Zhang [17], who showed that the final iterate has error $O(\log(T)/\sqrt{T})$ for Lipschitz f, and

$O(\log(T)/T)$ for f that is also strongly convex. Both of these bounds are a $\log(T)$ factor worse than the optimal rate, so Shamir and Zhang [17] write:

> An important open question is whether the $O(\log(T)/T)$ rate we obtained on [the last iterate], for strongly-convex problems, is tight. This question is important, because running SGD for T iterations, and returning the last iterate, is a very common heuristic... In fact, even for the simpler case of (non-stochastic) gradient descent, we do not know whether the behavior of the last iterate... is tight.

Our work shows that the $\log(T)$ factor is necessary, both for Lipschitz functions and for strongly convex functions. Thus, both of the upper bounds due to Shamir and Zhang are actually tight. This resolves the first question of Shamir [16]. In fact, we show a much stronger statement: *any convex combination* of the last k iterates must incur a $\log(T/k)$ factor. Thus, if an averaging scheme is used, then a constant fraction of the iterates must be averaged to achieve the optimal rate.

2 Preliminaries

Let $\mathcal{X}$ be a convex, compact and non-empty subset of $\mathbb{R}^n$ and let $f\colon \mathcal{X} \to \mathbb{R}$ be a convex function. We will assume[1] that f is subdifferentiable on $\mathcal{X}$, meaning that the subdifferential $\partial f(x)$ is non-empty for all $x \in \mathcal{X}$. The goal is to solve the convex program $\min_{x\in\mathcal{X}} f(x)$. We will assume that a minimizer exists[2]. We do not assume that an explicit representation of f is provided. Instead, the algorithm can only query f via a subgradient oracle, which is a subroutine that, given $x \in \mathcal{X}$, returns any vector $g \in \partial f(x)$. The set $\mathcal{X}$ is represented by a projection oracle, which is a subroutine that, given $x \in \mathbb{R}^n$, returns the point in $\mathcal{X}$ that is closest in Euclidean norm to x. The function f is called α-strongly convex if

$$f(y) \;\geq\; f(x) + \langle\, g,\, y - x \,\rangle + \frac{\alpha}{2}\,\|y-x\|^2 \qquad \forall\, y, x \in \mathcal{X}, g \in \partial f(x). \tag{1}$$

Throughout this paper, $\|\cdot\|$ denotes the *Euclidean* norm in $\mathbb{R}^n$ and $[T]$ denotes the set $\{1,\ldots,T\}$.

We will say that f is L-Lipschitz[3] on $\mathcal{X}$ if $\|g\| \leq L$ for all $x \in \mathcal{X}$ and $g \in \partial f(x)$. Let $\Pi_{\mathcal{X}}$ denote the projection operator on $\mathcal{X}$, which is defined by $\Pi_{\mathcal{X}}(y) = \operatorname{argmin}_{x\in\mathcal{X}} \|x-y\|$. The projected subgradient descent algorithm is given in Algorithm 1. Notice that there the algorithm maintains a sequence of points and there are several strategies to output a single point. The simplest strategy is to simply output x_{T+1}. However, one can also consider averaging all the iterates [12, 15] or averaging only a fraction of the final iterates [13].

Notice that the final iteration number T could be chosen in advance and provided as input, or could be determined dynamically during the course of the algorithm. We will refer to the former as the *fixed-time* setting and the latter as the *anytime* setting. In the fixed-time setting the sequence η_t of step sizes has length T and its values can depend on T, whereas in the anytime setting it should have infinite length and the values cannot depend on T.

For Lipschitz functions, uniform averaging with $\eta_t = \Theta(1/\sqrt{T})$ (fixed-time setting) [10] or $\eta_t = \Theta(1/\sqrt{t})$ (anytime setting) [4, Theorem 3.1] are known to achieve error rate $O(1/\sqrt{T})$. For functions that are also strongly convex, uniform averaging with $\eta_t = \Theta(1/T)$ (fixed-time setting) and suffix averaging with $\eta_t = \Theta(1/t)$ (anytime setting) are known [13] to achieve error rate $O(1/T)$. Recently Jain et al. [8] considered the error of the final iterate, in the *fixed-time* setting only. They showed that a non-obvious choice of step size gives error rate of $O(1/\sqrt{T})$ for Lipschitz functions and $O(1/T)$ for functions that are also strongly-convex. Nesterov and Shikhman [11] described an algorithm different than GD for which the t^{th} iterate has error rate $O(1/\sqrt{T})$ in the Lipschitz setting.

3 Statement of results

This paper proves the following lower bounds on the error of the final iterate for GD for non-smooth, convex functions.

[1] This holds, for example, if f is finite and convex on an open superset of $\mathcal{X}$ [14, Theorem 23.4].
[2] This holds, for example, if f is continuous, by Weierstrass' theorem.
[3] Our definition is slightly stronger than the standard definition $|f(x) - f(y)| \leq L\,\|x-y\|$ for all $x, y \in \mathcal{X}$. However, if the latter inequality holds on an open superset of $\mathcal{X}$, then this implies our definition.

Algorithm 1 Projected subgradient descent for minimizing a non-smooth, convex function. The final iteration number T could either be predetermined, or determined during the course of the algorithm.

```
1: procedure SubgradientDescent($\mathcal{X} \subseteq \mathbb{R}^n$, $x_1 \in \mathcal{X}$, step sizes $\eta_1, \eta_2, \ldots$)
2:     for $t \leftarrow 1, 2, \ldots$ do
3:         Query subgradient oracle at $x_t$ for $g_t \in \partial f(x_t)$
4:         $y_{t+1} \leftarrow x_t - \eta_t g_t$ (take a step in the opposite direction)
5:         $x_{t+1} \leftarrow \Pi_{\mathcal{X}}(y_{t+1})$ (project $y_{t+1}$ onto the set $\mathcal{X}$)
6:     $T \leftarrow t$ (the final iteration number)
7:     return either $\begin{cases} x_{T+1} & \text{(final iterate)} \\ \frac{1}{T+1}\sum_{t=1}^{T+1} x_t & \text{(uniform averaging)} \\ \frac{1}{T/2+1}\sum_{t=T/2+1}^{T+1} x_t & \text{(suffix averaging)} \end{cases}$
```

3.1 Strongly convex and Lipschitz functions

Theorem 1. *For any T and any constant $c > 0$, there exists a convex function $f_T : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the unit Euclidean ball in $\mathbb{R}^T$, such that f_T is $(3/c)$-Lipschitz and $(1/c)$-strongly convex, and satisfies the following. Suppose that Algorithm 1 is executed from the initial point $x_1 = 0$ with step sizes $\eta_t = c/t$. Let $x^* = \operatorname{argmin}_{x \in \mathcal{X}} f_T(x)$. Then*

$$f_T(x_{T+1}) - f_T(x^*) \;\geq\; \frac{\log T}{4cT} \tag{2}$$

More generally, any convex combination $\bar{x}$ of the last k iterates has

$$f_T(\bar{x}) - f_T(x^*) \;\geq\; \frac{\ln(T) - \ln(k)}{4cT}. \tag{3}$$

Thus, suffix averaging must average a constant fraction of iterates to achieve the optimal $O(1/T)$ error.

The assumptions of this theorem and the lower bound that it provides asymptotically match the following upper bound.

Theorem 2 (Shamir and Zhang [17, Theorem 1])**.** *Consider any function f that is G-Lipschitz and λ-strongly convex, and any closed convex set $\mathcal{X}$. Suppose that Algorithm 1 is executed from the initial point $x_1 = 0$ with step sizes $\eta_t = 1/\lambda t$. Let $x^* = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$. Then*

$$f(x_T) - f(x^*) \;\leq\; \frac{17G^2(1 + \log T)}{\lambda T} \qquad \forall\, T > 1.$$

Setting $G = (3/c)$ and $\lambda = 1/c$, the lower bound from Eq. (2) can be rewritten as $\frac{G^2}{36\lambda}\frac{\log T}{T}$, which is within a constant factor of the upper bound in Theorem 2.

▶ Remark 3. Theorem 1 proves a lower bound for the anytime setting. An analogous statement for the fixed-time setting is discussed in Section 4.2.

The function f_T from Theorem 1 is not difficult to state. Recall that $[n] = \{1, \ldots, n\}$. Focusing on the case $c = 1$, we define $f_T : \mathbb{R}^T \to \mathbb{R}$ and $h_i \in \mathbb{R}^T$ for $i \in [T+1]$ by

$$f_T(x) \;=\; \max_{i \in [T+1]} H_i(x) \qquad \text{where} \qquad H_i(x) = h_i^{\mathsf{T}} x + \frac{1}{2}\|x\|^2$$

$$h_{i,j} \;=\; \begin{cases} a_j & \text{(if } 1 \leq j < i) \\ -1 & \text{(if } i = j \leq T) \\ 0 & \text{(if } i < j \leq T) \end{cases} \qquad \text{and} \qquad a_j \;=\; \frac{1}{2(T+1-j)} \qquad (\text{for } j \in [T]).$$

Here we use the notation $h_{i,j}$ to denote the j^{th} component of the vector h_i.

3.2 Lipschitz functions

Theorem 4. *For any T and any constant $c > 0$, there exists a convex function $f_T : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the unit Euclidean ball in $\mathbb{R}^T$, such that f_T is $1/c$-Lipschitz, and satisfies the following. Suppose that Algorithm 1 is* *executed from the initial point $x_1 = 0$ with step sizes $\eta_t = c/\sqrt{t}$ with $c > 0$. Let $x^* = \operatorname{argmin}_{x \in \mathcal{X}} f_T(x)$. Then*

$$f_T(x_{T+1}) - f_T(x^*) \;\geq\; \frac{\log T}{32c\sqrt{T}}. \tag{4}$$

More generally, any weighted average $\bar{x}$ of the last k iterates has

$$f_T(\bar{x}) - f_T(x^*) \;\geq\; \frac{\ln(T) - \ln(k)}{32c\sqrt{T}}. \tag{5}$$

Furthermore, the value of f_T strictly monotonically increases *for the first T iterations:*

$$f_T(x_{t+1}) \;\geq\; f_T(x_t) + \frac{1}{64c\sqrt{T}(T-t+1)} \qquad \forall\, t \in [T]. \tag{6}$$

The assumptions of this theorem and the lower bound that it provides asymptotically match the following upper bound.

Theorem 5 (Shamir and Zhang [17, Theorem 2])**.** *Consider any function f that is G-Lipschitz and any closed convex set $\mathcal{X}$ with diameter D. Suppose that Algorithm 1 is executed from the initial point $x_1 = 0$ with step sizes $\eta_t = c/\sqrt{t}$. Let $x^* = \operatorname{argmin}_{x \in \mathcal{X}} f_T(x)$. Then*

$$f(x_T) - f(x^*) \;\leq\; \left(\frac{D^2}{c} + cG^2\right)\frac{2 + \log T}{\sqrt{T}} \qquad \forall\, T > 1.$$

Setting $G = (1/c)$ and $D = 2$, the lower bound from Eq. (4) can be rewritten as $\left(\frac{D^2}{c} + cG^2\right)\frac{\log T}{160\sqrt{T}}$, which is within a constant factor of the upper bound in Theorem 5.

▶ Remark 6. A weaker form of Eq. (4), with the constant 64 instead of 32, follows by summing Eq. (6).

▶ Remark 7. Theorem 4 proves a lower bound for the anytime setting. An analogous statement for the fixed-time setting is discussed in Section 5.1.

The function f_T from Theorem 4 is not difficult to state. We focus on the case $c = 1$. For $i \in [T]$, define the positive scalar parameters

$$a_i \;=\; \frac{1}{8(T-i+1)} \qquad\qquad b_i \;=\; \frac{\sqrt{i}}{2\sqrt{T}}.$$

Define $f : \mathbb{R}^T \to \mathbb{R}$ and $h_i \in \mathbb{R}^T$ for $i \in [T+1]$ by

$$f(x) \;=\; \max_{i \in [T+1]} h_i^{\mathsf{T}} x \qquad \text{where} \qquad h_{i,j} \;=\; \begin{cases} a_j & (\text{if } 1 \leq j < i) \\ -b_i & (\text{if } i = j \leq T) \\ 0 & (\text{if } i < j \leq T) \end{cases}.$$

As above, we use the notation $h_{i,j}$ to denote the j^{th} component of the vector h_i.

3.3 A construction independent of T

In order to incur a $\log T$ factor in the error of the T^{th} iterate, Theorem 1 and Theorem 4 construct a function f_T parameterized by T. It is also possible to create a single function f, *independent* of T, which incurs an additional factor very slightly below $\log T$ for infinitely many T. Theorem 24 constructs such a function that is both Lipschitz and strongly convex; this function is infinite-dimensional. This construction gives an analogue of Theorem 1 with a function independent of T. A trivial modification of that argument gives an analogue of Theorem 4.

3.4 Discussion of step size

In Theorem 1, the step size η_t is chosen to have the very specific form $\eta_t = c/t$. It seems conceivable to prove an analogous lower bound for all step sizes of the form $\eta_t = \Theta(1/t)$, however we have not been able to accomplish that. In particular, Theorem 1 requires that the constant c used in the definition of η_t is related to the Lipschitz parameter $(3/c)$ and the strong convexity parameter $(1/c)$. It seems conceivable to allow these parameters to be independent, although we have been unable to accomplish that.

Similarly, Theorem 4 requires a step size of $\eta_t = c/\sqrt{t}$. It seems conceivable to generalize the lower bound for all step sizes of the form $\eta_t = \Theta(1/\sqrt{t})$. In particular, Theorem 4 requires that the constant c used in the definition of η_t is related to the Lipschitz parameter $(1/c)$. It also seems conceivable to allow these parameters to be independent.

4 Proof of Theorem 1

In this section we will prove Theorem 1 in the case where $c = 1$. This implies the general statement by applying the following reduction, which is easily verifiable via induction.

Lemma 8. *Consider executing Algorithm 1 on the convex function $f : \mathcal{X} \mapsto \mathbb{R}$, using initial point x_1, step-sizes η_t, and subgradient oracle σ for which $\sigma(x) \in \partial f(x)$. Suppose that it produces the iterates $x_1, x_2, \ldots$. Then, for any $c > 0$, executing Algorithm 1 on the function $(1/c) \cdot f$, using initial point x_1, step-sizes $c \cdot \eta_t$, and subgradient oracle $(1/c) \cdot \sigma$ also yields the iterates $x_1, x_2, \ldots$.*

Henceforth assume that $c = 1$. For convenience, we reiterate the definition of $f = f_T$ that was given in Section 3.1, and we emphasize that this function depends on T. We will show that the final iterate produced by Algorithm 1 has $f(x_T) = \log(T)/4T$ and $\min_{x \in \mathcal{X}} f(x) \leq 0$, thereby proving (2). Let $\mathcal{X}$ be the Euclidean unit ball in $\mathbb{R}^T$. Define $f : \mathcal{X} \to \mathbb{R}$ and $h_i \in \mathbb{R}^T$ for $i \in [T+1]$ by

$$
\begin{aligned}
f(x) &= \max_{i \in [T+1]} H_i(x) && (7)\\
\text{where} \quad H_i(x) &= h_i^\mathsf{T} x + \frac{1}{2}\|x\|^2 \\
h_{i,j} &= \begin{cases} a_j & (\text{if } 1 \leq j < i) \\ -1 & (\text{if } i = j \leq T) \\ 0 & (\text{if } i < j \leq T) \end{cases} \\
a_j &= \frac{1}{2(T+1-j)} \qquad (\text{for } j \in [T]).
\end{aligned}
$$

Recall that $h_{i,j}$ denotes the j^{th} component of the vector h_i.

It is easy to see that f is 1-strongly convex due to the $\frac{1}{2}\|x\|^2$ term. Furthermore f is 3-Lipschitz over $\mathcal{X}$ because $\|\nabla H_i(x)\| \leq \|h_i\| + \|x\| \leq \|h_i\| + 1$ and $\|h_i\|^2 \leq 1 + \frac{1}{4}\sum_{j=1}^{T} \frac{1}{(T+1-j)^2} < 1 + \frac{1}{2}$. Finally, the minimum value of f over $\mathcal{X}$ is non-positive because $H_i(0) = 0$ for all $i \in [T+1]$, so $f(0) = 0$.

Subgradient oracle. In order to execute Algorithm 1 on f we must specify a subgradient oracle. First, we require the following claim, which follows from standard facts in convex analysis [7, Theorem 4.4.2].

▷ Claim 9. $\partial f(x)$ is the convex hull of $\{\, h_i + x \,:\, i \in \mathcal{I}(x) \,\}$, where $\mathcal{I}(x) = \{\, i \,:\, H_i(x) = f(x) \,\}$.

Our subgradient oracle is simple: given x, it returns $h_{i'} + x$ where $i' = \min \mathcal{I}(x)$.

Explicit description of iterates. Next we will explicitly describe the iterates produced by executing Algorithm 1 on f. Define the points $z_t \in \mathbb{R}^T$ for $t \in [T+1]$ by $z_1 = 0$ and

$$
z_{t,j} = \begin{cases} \dfrac{1 - (t-j-1)a_j}{t-1} & (\text{if } 1 \leq j < t) \\ 0 & (\text{if } t \leq j \leq T) \end{cases} \qquad (\text{for } t > 1).
$$

We will show inductively that these are precisely the iterates produced by Algorithm 1 when using $x_1 = 0$ and the subgradient oracle defined above. First some preliminary claims are necessary.

▷ Claim 10. For $t \in [T+1]$, z_t is non-negative. In particular, $z_{t,j} \geq \frac{1}{2(t-1)}$ for $j < t$ and $z_{t,j} = 0$ for $j \geq t$.

Proof. By definition, $z_{t,j} = 0$ for all $j \geq t$. For $j < t$, we use that $t - 1 < T + 1$ to obtain

$$
z_{t,j} = \frac{1 - (t-j-1)a_j}{t-1} > \frac{1}{t-1} \cdot \big(1 - (T+1-j)a_j\big) = \frac{1}{t-1} \cdot \frac{1}{2}. \qquad ◀
$$

▷ Claim 11. $\|z_1\| = 0$ and $\|z_t\|^2 \leq \frac{1}{t-1}$ for $t > 1$. Thus $z_t \in \mathcal{X}$ for all $t \in [T+1]$.

Proof. The claim obviously holds for $z_1 = 0$, so assume $t \geq 2$. We have $z_{t,j} = 0$ for all $j \geq t$, and for $j < t$, we have

$$z_{t,j} \;=\; \frac{1-(t-j-1)a_j}{t-1} \;\leq\; \frac{1}{t-1}.$$

Since z_t is non-negative by Claim 10, it follows that $\|z_t\|^2 \leq \frac{1}{t-1}$. ◀

Using the definition of the h_i vectors we can determine the value and subdifferential at z_t.

▷ Claim 12. $f(z_t) = H_t(z_t)$ for all $t \in [T+1]$. The subgradient oracle for f at z_t returns the vector $h_t + z_t$.

Proof. We claim that $h_t^\mathsf{T} z_t = h_i^\mathsf{T} z_t$ for all $i > t$. This follows since z_t is supported on its first $t-1$ coordinates and since h_t and h_i agree on the first $t-1$ coordinates (for $i > t$).

Next we claim that $h_t^\mathsf{T} z_t > h_i^\mathsf{T} z_t$ for all $1 \leq i < t$.

$$\begin{aligned}
(h_t - h_i)^\mathsf{T} z_t &= \sum_{j=1}^{t-1} (h_{t,j} - h_{i,j}) z_{t,j} && (z_t \text{ is supported on first } t-1 \text{ coordinates})\\
&= \sum_{j=i}^{t-1} (h_{t,j} - h_{i,j}) z_{t,j} && (h_i \text{ and } h_t \text{ agree on first } i-1 \text{ coordinates})\\
&= (a_i + 1) z_{t,i} + \sum_{j=i+1}^{t-1} a_j z_{t,j}.
\end{aligned}$$

This is strictly positive by the definition of a_j and since $z_t \geq 0$ by Claim 10.

These two statements imply that $H_t(z_t) \geq H_i(z_t)$ for all $i \in [T+1]$, and therefore $f(z_t) = H_t(z_t)$. Moreover $\mathcal{I}(z_t) = \{\, i : H_i(z_t) = f(z_t) \,\} = \{t, \ldots, T+1\}$. Thus, when evaluating the subgradient oracle at the vector z_t, it returns the vector $h_t + z_t$. ◀

Since the subgradient returned at z_t is determined by Claim 12, and the next iterate of GD arises from a step in the opposite direction, a straightforward induction proof allows us to show the following lemma.

Lemma 13. *For the function f defined in* (7)*, the vector x_t in Algorithm 1 equals z_t, for every $t \in [T+1]$.*

Proof. By definition, $z_1 = x_1 = 0$. By Claim 12, the subgradient returned at x_1 is $h_1 + x_1 = h_1$, so Algorithm 1 sets $y_2 = x_1 - \eta_1 h_1 = e_1$, the first standard basis vector (since $h_1 = -e_1$). Then Algorithm 1 projects onto the feasible region, obtaining $x_2 = \Pi_{\mathcal{X}}(y_2)$, which also equals e_1 since $y_2 \in \mathcal{X}$. Since z_2 also equals e_1, the base case is proven.

So assume $z_t = x_t$ for $2 \leq t < T$; we will prove that $z_{t+1} = x_{t+1}$. By Claim 12, the subgradient returned at x_t is $g_t = h_t + z_t$. Then Algorithm 1 sets $y_{t+1} = x_t - \eta_t g_t$. Since $x_t = z_t$ and $\eta_t = 1/t$, we obtain

$$\begin{aligned}
y_{t+1,j} &= z_{t,j} - \frac{1}{t}(h_{t,j} + z_{t,j})\\
&= \frac{t-1}{t} z_{t,j} - \frac{1}{t} h_{t,j}\\
&= \frac{t-1}{t} \begin{Bmatrix} \frac{1-(t-j-1)a_j}{t-1} & (\text{for } j < t)\\ 0 & (\text{for } j \geq t) \end{Bmatrix} - \frac{1}{t} \begin{Bmatrix} a_j & (\text{for } j < t)\\ -1 & (\text{for } j = t)\\ 0 & (\text{for } j > t) \end{Bmatrix}\\
&= \frac{1}{t} \begin{Bmatrix} 1-(t-j-1)a_j & (\text{for } j < t)\\ 0 & (\text{for } j \geq t) \end{Bmatrix} - \frac{1}{t} \begin{Bmatrix} a_j & (\text{for } j < t)\\ -1 & (\text{for } j = t)\\ 0 & (\text{for } j > t) \end{Bmatrix}\\
&= \frac{1}{t} \begin{Bmatrix} 1-(t-j)a_j & (\text{for } j < t)\\ 1 & (\text{for } j = t)\\ 0 & (\text{for } j \geq t+1) \end{Bmatrix}
\end{aligned}$$

So $y_{t+1} = z_{t+1}$. Since $x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1})$ is defined to be the projection onto $\mathcal{X}$, and $y_{t+1} \in \mathcal{X}$ by Claim 11, we have $x_{t+1} = y_{t+1} = z_{t+1}$. ◀

Now that we have determined the exact sequence of iterates chosen by the algorithm, the following claim proves (3) for the case $c = 1$. Inequality (2) is simply the special case where $k = 1$.

▷ Claim 14. For $k \in [T]$, let $\bar{x} = \sum_{t=T-k+2}^{T+1} \lambda_t x_t$ be any convex combination of the last k iterates. Then

$$f(\bar{x}) \;\geq\; \frac{\ln(T) - \ln(k)}{4T}.$$

Proof. By Lemma 13, $x_t = z_t$ for all $t \in [T+1]$. By Claim 10, every $z_t \geq 0$ so $\bar{x} \geq 0$. Moreover, by Claim 10 again, $z_{t,j} \geq 1/2T$ for all $T-k+2 \leq t \leq T+1$ and $1 \leq j \leq T-k+1$. Consequently, $\bar{x}_j \geq 1/2T$ for all $1 \leq j \leq T-k+1$. Thus,

$$\begin{aligned}
f(\bar{x}) &\geq h_{T+1}^{\mathsf{T}} \bar{x} \qquad \text{(by definition of } f\text{)} \\
&= \sum_{j=1}^{T-k+1} h_{T+1,j} \underbrace{\bar{x}_j}_{\geq 1/2T} + \sum_{j=T-k+2}^{T} \underbrace{h_{T+1,j}\, \bar{x}_j}_{\geq 0} \\
&\geq \sum_{j=1}^{T-k+1} a_j \cdot \frac{1}{2T} \\
&= \frac{1}{4T} \sum_{j=1}^{T-k+1} \frac{1}{T+1-j} \\
&\geq \frac{1}{4T} \int_1^{T-k+1} \frac{1}{T+1-x}\, dx \\
&= \frac{\log(T) - \log(k)}{4T}
\end{aligned}$$

◀

4.1 The origin is the minimizer

Above we have argued that $f(0) = 0$ but we did not determine the minimum value of f. In fact, the origin is the unique minimizer.

Lemma 15. *The origin is the unique minimizer of f.*

Proof. We have shown that $f(0) = 0$, so it remains to show that $f(x) > 0$ for all non-zero x. There are two cases to consider.

The first case is that every coordinate of x is non-negative. From the definition of h_{T+1} we see that all of its coordinates are strictly positive. Since x is non-zero, we have $h_{T+1}^{\mathsf{T}} x > 0$. Thus we have that

$$f(x) \;\geq\; H_{T+1}(x) \;\geq\; h_{T+1}^{\mathsf{T}} x \;>\; 0.$$

In the other case, let $i \in [T]$ be the smallest value for which $x_i < 0$. By definition we have $h_{i,j} > 0$ for all $j < i$. Thus

$$h_i^{\mathsf{T}} x \;=\; \underbrace{\sum_{1 \leq j < i} h_{i,j} x_j}_{\geq 0} + \underbrace{h_{i,i}}_{=-1} \underbrace{x_i}_{<0} \;>\; 0.$$

As in the first case, we obtain that

$$f(x) \;\geq\; H_i(x) \;\geq\; h_i^{\mathsf{T}} x \;>\; 0.$$

◀

4.2 Fixed-time setting

An analog of Theorem 1 holds in the fixed-time setting, using step sizes $\eta_t = 1/T$. The main change to the proof is that we must define

$$z_{t,j} \;=\; \begin{cases} \dfrac{1-(t-j-1)a_j}{T-1} & (\text{if } 1 \leq j < t) \\ 0 & (\text{if } t \leq j \leq T). \end{cases} \qquad (\text{for } t > 1).$$

This definition satisfies $z_{t,j} \geq 1/2(T-1)$ for $j < t$ and $\|z_t\|^2 \leq 1/(T-1)$. The same proof, mutatis mutandis, shows that

$$f_T(x_{T+1}) - f_T(x^*) \;\geq\; \frac{\log T}{4(T-1)}$$

5 Proof of Theorem 4

This section is similar to the previous one, the main difference being that we define a function that is not strongly convex, which yields a stronger lower bound.

To prove Theorem 4 it again suffices to consider the case $c = 1$ since the general statement again follows by Lemma 8. We define a function $f = f_T$, depending on T, for which the final iterate produced by Algorithm 1 has $f(x_T) = \log(T)/32\sqrt{T}$ and $\min_{x \in \mathcal{X}} f(x) \leq 0$, thereby proving (4).

For convenience, we reiterate the definition of f that was given in Section 3.2. For $i \in [T]$, define the positive scalar parameters

$$a_i \;=\; \frac{1}{8(T-i+1)} \qquad\qquad b_i \;=\; \frac{\sqrt{i}}{2\sqrt{T}}.$$

As before, $\mathcal{X}$ denotes the Euclidean unit ball in $\mathbb{R}^T$. Define $f : \mathcal{X} \to \mathbb{R}$ and $h_i \in \mathbb{R}^T$ for $i \in [T+1]$ by

$$f(x) \;=\; \max_{i \in [T+1]} h_i^{\mathsf{T}} x \tag{8}$$

$$\text{where} \qquad h_{i,j} \;=\; \begin{cases} a_j & (\text{if } 1 \leq j < i) \\ -b_i & (\text{if } i = j \leq T) \\ 0 & (\text{if } i < j \leq T) \end{cases}.$$

This function f is 1-Lipschitz over $\mathcal{X}$ because

$$\|h_i\|^2 \;\leq\; \sum_{j=1}^{T} a_j^2 + b_T^2 \;=\; \frac{1}{64} \sum_{j=1}^{T} \frac{1}{j^2} + \frac{1}{4} \;<\; \frac{1}{2}.$$

The minimum value of f over $\mathcal{X}$ is non-positive because $f(0) = 0$.

Subgradient oracle. Similar to Claim 9, [7, Theorem 4.4.2] implies

▷ Claim 16. $\partial f(x)$ is the convex hull of $\{\, h_i \,:\, i \in \mathcal{I}(x) \,\}$, where $\mathcal{I}(x) = \{\, i \,:\, h_i^{\mathsf{T}} x = f(x) \,\}$.

Our subgradient oracle is simple: given x, it returns $h_{i'}$ where $i' = \min \mathcal{I}(x)$.

Explicit description of iterates. Next we will explicitly describe the iterates produced by executing Algorithm 1 on f. Define the points $z_t \in \mathbb{R}^T$ for $t \in [T+1]$ by $z_1 = 0$ and

$$z_{t,j} \;=\; \begin{cases} \left(\dfrac{b_j}{\sqrt{j}} - a_j \displaystyle\sum_{k=j+1}^{t-1} \dfrac{1}{\sqrt{k}} \right) & (\text{if } 1 \leq j < t) \\ 0 & (\text{if } t \leq j \leq T). \end{cases} \qquad (\text{for } t > 1).$$

We will show inductively that these are precisely the iterates produced by Algorithm 1 when using $x_1 = 0$ and the subgradient oracle defined above.

▷ Claim 17. For $t \in [T+1]$, z_t is non-negative. In particular, $z_{t,j} \geq \frac{1}{4\sqrt{T}}$ for $j < t$ and $z_{t,j} = 0$ for $j \geq t$.

Proof. By definition, $z_{t,j} = 0$ for all $j \geq t$. For $j < t$,

$$\begin{aligned} z_{t,j} &= \left(\frac{b_j}{\sqrt{j}} - a_j \sum_{k=j+1}^{t-1} \frac{1}{\sqrt{k}} \right) \\ &= \left(\frac{1}{2\sqrt{T}} - \frac{1}{8(T-j+1)} \sum_{k=j+1}^{t-1} \frac{1}{\sqrt{k}} \right) \quad \text{(by definition of } a_j \text{ and } b_j) \\ &\geq \frac{1}{2\sqrt{T}} - \frac{1}{4(T-j+1)} \frac{t-1-j}{\sqrt{t-1}} \quad \text{(by Claim 30)} \\ &\geq \frac{1}{2\sqrt{T}} - \frac{1}{4\sqrt{T}} \quad \text{(by Claim 31, replacing } t \text{ with } t-1) \\ &= \frac{1}{4\sqrt{T}}. \end{aligned}$$

◀

▷ Claim 18. $z_{t,j} \leq 1/\sqrt{T}$ for all j. In particular, $z_t \in \mathcal{X}$ (the unit ball in $\mathbb{R}^T$).

Proof. We have $z_{t,j} = 0$ for all $j \geq t$, and for $j < t$, we have

$$z_{t,j} = \left(\frac{b_j}{\sqrt{j}} - a_j \sum_{k=j+1}^{t} \frac{1}{\sqrt{k}} \right) \leq \frac{b_j}{\sqrt{j}} = \frac{1}{2\sqrt{T}}.$$

Since Claim 17 shows that $z_t \geq 0$, we have $\|z_t\| \leq 1$, and therefore $z_t \in \mathcal{X}$. ◀

Using the definition of the h_i vectors we can determine the value and subdifferential at z_t.

▷ Claim 19. $f(z_t) = h_t^\mathsf{T} z_t$ for all $t \in [T+1]$. The subgradient oracle for f at z_t returns the vector h_t.

Proof. We claim that $h_t^\mathsf{T} z_t = h_i^\mathsf{T} z_t$ for all $i > t$. This follows since z_t is supported on its first $t-1$ coordinates, and since h_t and h_i agree on the first $t-1$ coordinates (for $i > t$). Next we claim that $h_t^\mathsf{T} z_t > h_t^\mathsf{T} z_i$ for all $1 \leq i < t$. This also follows from the definition of z_t and h_i:

$$\begin{aligned} (h_t - h_i)^\mathsf{T} z_t &= \sum_{j=1}^{t-1} (h_{t,j} - h_{i,j}) z_{t,j} \quad (z_t \text{ is supported on first } t-1 \text{ coordinates}) \\ &= \sum_{j=i}^{t-1} (h_{t,j} - h_{i,j}) z_{t,j} \quad (h_i \text{ and } h_t \text{ agree on first } i-1 \text{ coordinates}) \\ &= (a_i + b_i) z_{t,i} + \sum_{j=i+1}^{t-1} a_j z_{t,j} \\ &> 0, \end{aligned}$$

since z_t is non-negative by Claim 17.

These two claims imply that $h_t^\mathsf{T} z_t \geq h_i^\mathsf{T} z_t$ for all $i \in [T+1]$, and therefore $f(z_t) = h_t^\mathsf{T} z_t$. Moreover $\mathcal{I}(z_t) = \{i : h_i^\mathsf{T} z_t = f(z_t)\} = \{t, \ldots, T+1\}$. Thus, when evaluating the subgradient oracle at the vector z_t, it returns the vector h_t. ◀

Since the subgradient returned at z_t is determined by Claim 19, and the next iterate of SGD arises from a step in the opposite direction, a straightforward induction proof allows us to show the following lemma.

Lemma 20. *For the function f defined in* (8)*, the vector x_t in Algorithm 1 equals z_t, for every $t \in [T+1]$.*

Proof. The proof is by induction. By definition $x_1 = 0$ and $z_1 = 0$, establishing the base case.

Assume $z_t = x_t$ for $t \leq T$; we will prove that $z_{t+1} = x_{t+1}$. Recall that Algorithm 1 sets $y_{t+1} = x_t - \eta_t g_t$, and that $\eta_t = \frac{1}{\sqrt{t}}$. By the inductive hypothesis, $x_t = z_t$. By Claim 19, the algorithm uses the subgradient $g_t = h_t$.

Thus,

$$\begin{aligned} y_{t+1,j} &= z_{t,j} - \frac{1}{\sqrt{t}} h_{t,j} \\ &= \left\{ \begin{array}{ll} \frac{b_j}{\sqrt{j}} - a_j \sum_{k=j+1}^{t-1} \frac{1}{\sqrt{k}} & \text{(for } 1 \le j < t) \\ 0 & \text{(for } j \ge t) \end{array} \right\} - \frac{1}{\sqrt{t}} \left\{ \begin{array}{ll} a_j & \text{(for } 1 \le j < t) \\ -b_t & \text{(for } j = t) \\ 0 & \text{(for } j > t) \end{array} \right\} \\ &= \left\{ \begin{array}{ll} \frac{b_j}{\sqrt{j}} - a_j \sum_{k=j+1}^{t} \frac{1}{\sqrt{k}} & \text{(for } j < t) \\ \frac{b_t}{\sqrt{t}} & \text{(for } j = t) \\ 0 & \text{(for } j > t) \end{array} \right\} \end{aligned}$$

So $y_{t+1} = z_{t+1}$. Since $x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1})$ by definition, and $y_{t+1} \in \mathcal{X}$ by Claim 18, we have $x_{t+1} = y_{t+1} = z_{t+1}$. ◀

Now that we have determined the exact sequence of iterates chosen by the algorithm, the following claim proves (5) for the case $c = 1$. Inequality (4) is simply the special case where $k = 1$.

▷ Claim 21. For $k \in [T]$, let $\bar{x} = \sum_{t=T-k+2}^{T+1} \lambda_t x_t$ be any convex combination of the last k iterates. Then

$$f(\bar{x}) \ \ge\ \frac{\ln(T) - \ln(k)}{32\sqrt{T}}.$$

Proof. By Lemma 20, $x_t = z_t$ for all $t \in [T+1]$. By Claim 17, every $z_t \ge 0$ so $\bar{x} \ge 0$. Moreover, by Claim 17 again, $z_{i,j} \ge 1/4\sqrt{T}$ for all $T-k+2 \le t \le T+1$ and $1 \le j \le T-k+1$. Consequently, $\bar{x}_j \ge 1/4\sqrt{T}$ for all $1 \le j \le T-k+1$. Thus,

$$\begin{aligned} f(\bar{x}) &\ge h_{T+1}^{\mathsf{T}} \bar{x} \quad \text{(by definition of } f) \\ &= \sum_{j=1}^{T-k+1} h_{T+1,j}\bar{x}_j + \sum_{j=T-k+2}^{T} \underbrace{h_{T+1,j}\bar{x}_j}_{\ge 0} \\ &\ge \sum_{j=1}^{T-k+1} a_j \frac{1}{4\sqrt{T}} \\ &= \frac{1}{4\sqrt{T}} \sum_{j=1}^{T-k+1} \frac{1}{8(T-j+1)} \\ &\ge \frac{1}{32\sqrt{T}} \int_1^{T-k+1} \frac{1}{T-x+1}\,dx \\ &= \frac{\log(T) - \log(k)}{32\sqrt{T}} \end{aligned}$$

◀

The following claim completes the proof of (6), for the case $c = 1$.

▷ Claim 22. For any $t \in [T]$, we have $f(x_{t+1}) \ge f(x_t) + 1/64\sqrt{T}(T-t+1)$.

Proof.

$$\begin{aligned} f(x_{t+1}) - f(x_t) &= h_{t+1}^{\mathsf{T}} z_{t+1} - h_t^{\mathsf{T}} z_t \qquad \text{(by Claim 19)} \\ &= \sum_{j=1}^{t} (h_{t+1,j} z_{t+1,j} - h_{t,j} z_{t,j}) \qquad \text{(due to support of } z_{t+1} \text{ and } z_t) \\ &= \sum_{j=1}^{t-1} (h_{t+1,j} z_{t+1,j} - h_{t,j} z_{t,j}) + (h_{t+1,t} z_{t+1,t} - h_{t,t} \underbrace{z_{t,t}}_{=0}) \\ &= \sum_{j=1}^{t-1} a_j (z_{t+1,j} - z_{t,j}) + a_t z_{t+1,t} \qquad \text{(by definition of } h_{t+1} \text{ and } h_t) \\ &\ge \sum_{j=1}^{t-1} a_j \cdot \Big(\frac{-a_j}{\sqrt{t}}\Big) + a_t \Big(\frac{1}{4\sqrt{T}}\Big) \qquad \text{(by definition of } z_{t+1} \text{ and } z_t \text{, and Claim 17)} \end{aligned}$$

$$\geq -\frac{1}{64\sqrt{t}} \sum_{j=1}^{t-1} \Big(\frac{1}{T-j+1}\Big)^2 + \frac{1}{32\sqrt{T}(T-t+1)} \qquad \text{(by definition of } a_j)$$

$$\geq \frac{1}{64\sqrt{T}(T-t+1)} \qquad \text{(by Claim 23)}$$ ◀

▷ Claim 23. For any $t \in [T]$,

$$\frac{1}{\sqrt{t}} \sum_{j=1}^{t-1} \Big(\frac{1}{T-j+1}\Big)^2 \leq \frac{1}{\sqrt{T}} \cdot \frac{1}{T-t+1}.$$

Proof. If $t = 1$, the sum is empty so the left-hand side is zero. If $t > 1$, Claim 32 yields

$$\sum_{j=1}^{t-1} \Big(\frac{1}{T-j+1}\Big)^2 = \sum_{\ell=T-t+2}^{T} \frac{1}{\ell^2} \leq \frac{1}{T-t+1} - \frac{1}{T} < \frac{t}{T(T-t+1)}.$$

So it suffices to prove that

$$\frac{\sqrt{t}}{T(T-t+1)} \leq \frac{1}{\sqrt{T}} \cdot \frac{1}{T-t+1}.$$

This obviously holds, since $t \leq T$. ◀

5.1 Fixed-time setting

An analog of Theorem 4 holds in the fixed-time setting, using step sizes $\eta_t = 1/\sqrt{T}$. The main change to the proof is that we must define $b_i = 1/2$ and

$$z_{t,j} = \begin{cases} \dfrac{b_i - (t-j-1)a_j}{\sqrt{T}} & (\text{if } 1 \leq j < t) \\ 0 & (\text{if } t \leq j \leq T). \end{cases} \qquad (\text{for } t > 1).$$

This definition satisfies $z_{t,j} \geq 1/4\sqrt{T}$ for $j < t$ and $\|z_t\|^2 \leq 1$. The same proof, mutatis mutandis, shows that

$$f_T(x_{T+1}) - f_T(x^*) \geq \frac{\log T}{32\sqrt{T}}$$

6 A construction independent of T

In order to achieve large error after T iterations of GD, Theorem 1 constructs a function parameterized by T. One may wonder whether a single function could achieve error $\Omega(\log(T)/T)$ for *every* $T \geq 1$. Unfortunately this is not possible. If such a function existed then, for even T, we would have

$$\frac{1}{T/2} \sum_{i=1+T/2}^{T} \big(f(x_i) - f(x^*)\big) = \Omega(\log(T)/T).$$

However, the analysis of suffix averaging [13] proves that

$$\frac{1}{T/2} \sum_{i=1+T/2}^{T} \big(f(x_i) - f(x^*)\big) = O(1/T),$$

which is a contradiction. So no such function can exist.

In this section, we will show a slightly weaker result: for every function $g(T) = o(\log(T)/T)$, we can construct a strongly convex function for which the final iterate has error at least $C \cdot g(T)$, for every $C > 0$ and for infinitely many T.

In this section we will use convex functions in Hilbert space. The key definitions (convexity, strong convexity, subgradients, etc.) are essentially unchanged from the finite dimensional setting; see, e.g., Bauschke and Combettes [3] or Barbu and Precupanu [1]. As usual, ℓ_2 denotes the space of square-summable sequences in $\mathbb{R}^{\mathbb{N}}$.

The main result of this section is the following.

Theorem 24. *For every $c > 0$, there exists $\mathcal{X} \subset \ell_2$, a convex function $f : \mathcal{X} \to \mathbb{R}$, and a subgradient oracle for f such that f is $(3/c)$-Lipschitz, f is $(1/c)$-strongly convex, $\inf_{x\in\mathcal{X}} f(x) = 0$, and with the following property. Suppose that Algorithm 1 is executed from the initial point $x_1 = 0$ with step sizes $\eta_t = c/t$. Then, for every non-negative $g(T) = o(\log(T)/T)$,*

$$\limsup_{T\to\infty} \frac{f(x_T)}{g(T)} \;=\; \infty. \tag{9}$$

For functions that are not strongly convex, the following analogous result holds.

Theorem 25. *For every $c > 0$, there exists $\mathcal{X} \subset \ell_2$, a convex function $f : \mathcal{X} \to \mathbb{R}$, and a subgradient oracle for f such that f is $(1/c)$-Lipschitz, $\inf_{x\in\mathcal{X}} f(x) = 0$, and with the following property. Suppose that Algorithm 1 is executed from the initial point $x_1 = 0$ with step sizes $\eta_t = c/\sqrt{t}$. Then, for every non-negative $g(T) = o\big(\log(T)/\sqrt{T}\big)$,*

$$\limsup_{T\to\infty} \frac{f(x_T)}{g(T)} \;=\; \infty.$$

The remainder of this section proves Theorem 24 for the case $c = 1$. We omit the proof for arbitrary $c > 0$ and the proof of Theorem 25 because they are trivial modifications.

The main tool we use to prove Theorem 24 is Lemma 26, whose statement appears technical, but is actually quite intuitive. In a nutshell, this lemma states that running Algorithm 1 on an infinite sum of convex functions is equivalent to running an instance of Algorithm 1 for each summand in parallel. The proof of Lemma 26 appears in Subsection 6.2.

Lemma 26. *Let $C_1, C_2, \ldots$ be positive integers satisfying $\sum_{i=1}^{\infty} 1/C_i \leq 1$. Let $T_1, T_2, \ldots$ be positive integers. Let $\{f^{(i)}\}_{i=1}^{\infty}$ be a family of non-negative, convex functions where $f^{(i)} : \mathbb{R}^{T_i} \to \mathbb{R}$. Let $R > 0$. Let $\mathcal{X}_i = \mathcal{B}_{T_i}(0, R)$, the closed Euclidean ball of radius R in $\mathbb{R}^{T_i}$. Assume that $f^{(i)}$ is L-Lipschitz on $\mathcal{X}_i$, and that $f^{(i)}(0) = 0$. For any $x \in \ell_2$, we will decompose it into finite-dimensional vectors as*

$$x \;=\; (x^{[1]}, x^{[2]}, \ldots) \qquad \text{where} \qquad x^{[i]} \in \mathbb{R}^{T_i}. \tag{10}$$

Then $f : \ell_2 \to \mathbb{R}$ and $\mathcal{X}$ are defined as

$$f(x) \;=\; f\left(x^{[1]}, x^{[2]}, \ldots\right) \;=\; \sum_{i=1}^{\infty} \frac{1}{C_i^2} f^{(i)}\left(C_i x^{[i]}\right) \qquad \text{and} \qquad \mathcal{X} = \prod_{i=1}^{\infty} \frac{\mathcal{X}_i}{C_i}. \tag{11}$$

The following hold:

P1. *$\mathcal{X} \subset \ell_2$*

P2. *f is well-defined and finite on all of ℓ_2.*

P3. *f is convex on ℓ_2.*

P4. *f is subdifferentiable on $\mathcal{X}$.*

P5. *If $f^{(i)}$ is α-strongly convex on $\mathcal{X}_i$ for every i, then f is α-strongly convex on $\mathcal{X}$.*

P6. *f is L-Lipschitz on $\mathcal{X}$. That is, for every $x \in \mathcal{X}$ and $g \in \partial f(x)$, we have $\|g\| \leq L$.*

P7. *Let σ_i be an arbitrary subgradient oracle for $f^{(i)}$ (i.e., $\sigma_i(x) \in \partial f^{(i)}(x)\ \forall\, x \in \mathcal{X}_i$). Let $x_t^{(i)}$ denote the t^{th} iterate of Algorithm 1 on the function $f^{(i)}$ using the feasible region $\mathcal{X}_i$, step sizes η_t, initial point $x_1^{(i)}$ and the subgradient oracle σ_i. Then, there is a subgradient oracle σ on $\mathcal{X}$ such that, when executing Algorithm 1 on f with initial point*

$$x_1 \;=\; (x_1^{[1]}, x_1^{[2]}, \ldots) \;=\; \Big(\frac{x_1^{(1)}}{C_1}, \frac{x_1^{(2)}}{C_2}, \ldots\Big) \tag{12}$$

and step sizes η_t, then the t^{th} iterate satisfies

$$x_t \;=\; (x_t^{[1]}, x_t^{[2]}, \ldots) \;=\; \Big(\frac{x_t^{(1)}}{C_1}, \frac{x_t^{(2)}}{C_2}, \ldots\Big) \qquad \forall\, t \in \mathbb{N}. \tag{13}$$

In other words, $x_1^{[i]} = x_1^{(i)}/C_i$ for all $i \in \mathbb{N}$ implies $x_t^{[i]} = x_t^{(i)}/C_i$ for all $t \in \mathbb{N}$ and all $i \in \mathbb{N}$.

Applying Lemma 26. Lemma 26 constructs a single infinite dimensional function from many finite dimensional functions (see (11)) while maintaining crucial properties such as convexity, Lipschitzness, and boundedness. Importantly, running Algorithm 1 on this infinite dimensional function is "equivalent" to running an instance of Algorithm 1 for each finite dimensional function in parallel: The value of the t^{th} iterate of the infinite dimensional instance can be obtained by a weighted sum of the values of t^{th} iterates of the finite dimensional instances. To prove Theorem 24 we will construct a single function f using infinitely many instances of the function f_T from Section 4, with different values of T.

6.1 Proof of Theorem 24

Defining f. We would like to apply Lemma 26, so we must first satisfy its hypotheses. The simplest step is defining the constants $C_i = 2^i$; this clearly satisfies the requirement $\sum_{i=1}^{\infty} \frac{1}{C_i} \leq 1$. Next, since $g = o(\log(t)/t)$, Claim 34 implies existence of a positive function h such that $g(t) = o\big(\log(t)/(t \cdot h(t))\big)$ and $\lim_{t\to\infty} h(t) = \infty$. Thus, there exists a value T_i such that

$$T_i \geq i \qquad \text{and} \qquad g(t) \leq \frac{1}{4C_i^2}\left(\frac{\log t}{t \cdot h(t)}\right) \quad \forall\, t \geq T_i. \tag{14}$$

The set $\mathcal{X}_i$ is simply the unit ball $\mathcal{B}_{T_i}(0,1)$ in $\mathbb{R}^{T_i}$. The function f is defined as in (11), where $f^{(i)}$ is the T_i-dimensional function f_{T_i} defined in Section 3.1. Since f is a conic combination of the $f^{(i)}$ (see (11)), and each $f^{(i)}$ is non-negative (see Section 4.1), it follows that f is non-negative and $f(0) = 0$. Thus 0 is a minimizer of f over $\mathcal{X}$.

Applying Lemma 26. Recall that each $f^{(i)}$ is 3-Lipschitz and 1-strongly convex over $\mathcal{X}_i$. Furthermore $\partial f^{(i)}(x) \neq \emptyset$ for all $x \in \mathcal{X}_i$. Let σ_i be the subgradient oracle for $f^{(i)}$ described in Section 4. Let $x_t^{(i)}$ denote the t^{th} iterate of Algorithm 1 when executed on $f^{(i)}$ using the subgradient oracle σ_i, initial point $x_1^{(i)} = 0$, and step size $\eta_t = 1/t$. The conclusions of Lemma 26 are:

- f is well defined over $\mathcal{X}$.
- f is 3-Lipschitz and 1-strongly convex over $\mathcal{X}$.
- $\partial f(x) \neq \emptyset$ for all $x \in \mathcal{X}$.
- There exists a subgradient oracle σ for f over $\mathcal{X}$ such that, when executing Algorithm 1 on f with subgradient oracle σ, initial point $x_1 = 0$, and step size $\eta_t = 1/t$, the t^{th} iterate $x_t \in \mathcal{X}$ satisfies $x_t^{[i]} = x_t^{(i)}$ for all $i \in \mathbb{N}$.

The key point is: after running GD on the infinite-dimensional function f, the t^{th} iterate x_t has its i^{th} component $x_t^{[i]}$ equal to the t^{th} iterate $x_t^{(i)}$ produced by running GD on the finite-dimensional function $f^{(i)}$.

Proving Eq. (9). Consider any $M > 0$ and any $N \in \mathbb{N}$. Recalling that $\lim_{T\to\infty} h(T) = \infty$, it follows that

$$\exists\, n \in \mathbb{N} \quad \text{s.t.} \quad h(T) > M \quad \forall\, T \geq n. \tag{15}$$

Let $N' = \max\{3, n, N\}$. Then we have the following:

$$\begin{aligned}
f\left(x_{T_{N'}+1}\right) &= \sum_{i=1}^{\infty} \frac{1}{C_i^2} f^{(i)}\left(C_i x_{T_{N'}+1}^{[i]}\right) && \text{(by definition of } f \text{ in (11))}\\
&= \sum_{i=1}^{\infty} \frac{1}{C_i^2} f^{(i)}\left(x_{T_{N'}+1}^{(i)}\right) && \text{(by Lemma 26)}\\
&\geq \frac{1}{C_{N'}^2} f^{(N')}\left(x_{T_{N'}+1}^{(N')}\right) && \text{(each } f^{(i)} \text{ is non-negative)}\\
&\geq \frac{1}{4C_{N'}^2}\frac{\log T_{N'}}{T_{N'}} && \text{(by Eq. (2))}\\
&\geq \frac{1}{4C_{N'}^2}\frac{\log(T_{N'}+1)}{T_{N'}+1} && (\log(x)/x \text{ is decreasing for } x > e)\\
&\geq g(T_{N'}+1)\cdot h(T_{N'}+1) && \text{(by Eq. (14))}\\
&> M \cdot g(T_{N'}+1),
\end{aligned}$$

since $T_{N'} + 1 \geq N'$ by (14), and $N' \geq n$ by definition, then using Eq. (15).

To summarize, this argument shows that, for every $M > 0$ and for every $N \in \mathbb{N}$, there exists $t \geq N$ (namely, $t = T_{N'} + 1$) such that $f(x_t) > M \cdot g(t)$. This proves Eq. (9).

6.2 Proof of Lemma 26

Consider any $x \in \mathcal{X}$. Following (10), it decomposes as $x = (x^{[1]}, x^{[2]}, \ldots)$. The definition of $\mathcal{X}$ implies that $x^{[i]} \in \mathcal{X}_i / C_i$. Recall that $C_i \geq 1$, $\sum_{i \geq 1} 1/C_i \leq 1$ and $\mathcal{X}_i \subseteq B_{T_i}(0, R)$. Thus

$$\|x\|^2 \;=\; \sum_{i \geq 1} \left\| x^{[i]} \right\|^2 \;\leq\; \sum_{i \geq 1} \frac{R^2}{C_i^2} \;\leq\; R^2,$$

which proves P1.

To prove P2 we must show that $\lim_{n \to \infty} \sum_{i=1}^{n} \frac{1}{C_i^2} f^{(i)}(C_i x^{[i]})$ is convergent for all $x \in \ell_2$. Since each $f^{(i)}$ is non-negative, the series is monotonic, so it suffices to show that it is bounded. We have

$$f(x) \;=\; \sum_{i \geq 1} \frac{1}{C_i^2} f^{(i)}(C_i x^{[i]}) \;=\; \sum_{i \,:\, \|x^{[i]}\| > R} \frac{1}{C_i^2} f^{(i)}(C_i x^{[i]}) + \sum_{i \,:\, \|x^{[i]}\| \leq R} \frac{1}{C_i^2} f^{(i)}(C_i x^{[i]}).$$

The first sum is finite since $x \in \ell_2$. On the other hand, for any $y \in \mathcal{X}_i$, we have $f^{(i)}(y) \leq LR$ since $f^{(i)}(0) = 0$, $f^{(i)}$ is L-Lipschitz on $\mathcal{X}_i = B_{T_i}(0, R)$. Thus $0 \leq f^{(i)}(C_i x^{[i]}) \leq LR$ for all i. It follows that

$$\sum_{i \,:\, \|x^{[i]}\| \leq R} \frac{1}{C_i^2} f^{(i)}(C_i x^{[i]}) \;\leq\; LR \sum_{i \geq 1} \frac{1}{C_i^2} \;\leq\; LR.$$

This shows that $f(x)$ is finite, proving P2.

For any $x, y \in \ell_2$ and $\lambda \in [0, 1]$, we have

$$f^{(i)}(\lambda C_i x + (1 - \lambda) C_i y) \;\leq\; \lambda f^{(i)}(C_i x^{[i]}) + (1 - \lambda) f^{(i)}(C_i y^{[i]}) \qquad \forall\, i \geq 1$$

by convexity of $f^{(i)}$. If we sum these inequalities over $i \geq 1$ with coefficients $1/C_i^2$ then the sums converge by P2. Thus $f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y)$, thereby proving P3.

The following claim will be useful for the remaining properties. It is proven in Section 6.3 below.

▷ Claim 27. Let $x = (x^{[1]}, x^{[2]}, \ldots) \in \mathcal{X}$. Then $\partial f(x) = \prod_{i=1}^{\infty} \frac{1}{C_i} \partial f_i(C_i x^{[i]})$.

From this claim, P4 is immediate. For any $x = (x^{[1]}, x^{[2]}, \ldots) \in \mathcal{X}$, we have $C_i x^{[i]} \in \mathcal{X}_i$. Since $f^{(i)}$ is subdifferentiable on $\mathcal{X}_i$ (because it is finite and convex on all of $\mathbb{R}^{T_i}$), we have $\partial f^{(i)}(x) \neq \emptyset \;\; \forall\, x \in \mathcal{X}_i$. So Claim 27 and the axiom of choice imply that $\partial f(x) \neq \emptyset$, which establishes P4.

Next we consider P5. We will use the fact that a function $h(x)$ on a Hilbert space is α-strongly convex iff $h(x) - \alpha \|x\|^2 / 2$ is convex [3, Proposition 10.6]. Then

$$f(x) - \frac{\alpha}{2} \|x\|^2 \;=\; \sum_{i=1}^{\infty} \left[\frac{1}{C_i^2} f^{(i)}(C_i x^{[i]}) - \frac{\alpha}{2} \left\| x^{[i]} \right\|^2 \right] \;=\; \sum_{i=1}^{\infty} \left[\frac{1}{C_i^2} \left(f^{(i)}(C_i x^{[i]}) - \frac{\alpha}{2} \left\| C_i x^{[i]} \right\|^2 \right) \right].$$

This last sum is convex because $f^{(i)} - \frac{\alpha}{2} \|\cdot\|^2$ is convex, since $f^{(i)}$ is α-strongly convex.

Next we prove P6. Consider any $x = (x^{[1]}, x^{[2]}, \ldots) \in \mathcal{X}$ and any $g \in \partial f(x)$. Then Claim 27 implies that $g = (g^{[1]}/C_1, g^{[2]}/C_2, \ldots)$ where $g^{(i)} \in \partial f_i(C_i x^{(i)})$. Hence,

$$\|g\|^2 \;=\; \sum_{i=1}^{\infty} \frac{\left\| g^{(i)} \right\|^2}{C_i^2} \;\leq\; \sum_{i=1}^{\infty} \frac{L^2}{C_i^2} \;\leq\; L^2.$$

Lastly, we will prove P7. The definition of the subgradient oracle σ is straightforward:

$$\sigma(x) \;=\; \sigma\big((x^{[1]}, x^{[2]}, \ldots,)\big) \;=\; \left(\frac{\sigma_1(C_1 x^{(1)})}{C_1}, \frac{\sigma_2(C_2 x^{(2)})}{C_2}, \ldots \right).$$

This definition is valid due to Claim 27. The proof of (13) is by induction. The base case holds by definition of x_1 in (12). So suppose (13) holds for x_t. Then,

$$\begin{aligned}
y_{t+1} &= x_t - \eta_t \sigma(x_t) \qquad \text{(gradient step in Algorithm 1)} \\
&= \Big(\frac{x_t^{(1)}}{C_1}, \frac{x_t^{(2)}}{C_2}, \dots\Big) - \eta_t \sigma\Big(\Big(\frac{x_t^{(1)}}{C_1}, \frac{x_t^{(2)}}{C_2}, \dots\Big)\Big) \qquad \text{(by induction hypothesis)} \\
&= \Big(\frac{x_t^{(1)}}{C_1}, \frac{x_t^{(2)}}{C_2}, \dots\Big) - \eta_t \Big(\frac{\sigma_1(x_t^{(1)})}{C_1}, \frac{\sigma_2(x_t^{(2)})}{C_2}, \dots\Big) \qquad \text{(by definition of } \sigma) \\
&= \Big(\frac{1}{C_1}\Big(x_t^{(1)} - \eta_t \sigma_1(x_t^{(1)})\Big), \frac{1}{C_2}\Big(x_t^{(2)} - \eta_t \sigma_2(x_t^{(2)})\Big), \dots\Big) \\
&= \Big(\frac{1}{C_1} y_{t+1}^{(1)}, \frac{1}{C_2} y_{t+1}^{(2)}, \dots\Big) \qquad \text{(gradient step in Algorithm 1).}
\end{aligned}$$

The next step of Algorithm 1 is the projection: $x_{t+1} \leftarrow \Pi_{\mathcal{X}}(y_{t+1})$. This projection may be performed componentwise by Claim 28 (since $x_t \in \mathcal{X} \subset \ell_2$ by P1 and $\sigma(x_t) \in \ell_2$ by P6, so $y_{t+1} \in \ell_2$). Thus

$$\begin{aligned}
x_{t+1} &= \left(\Pi_{\mathcal{X}_1/C_1}\Big(\frac{y_{t+1}^{(1)}}{C_1}\Big), \Pi_{\mathcal{X}_2/C_2}\Big(\frac{y_{t+1}^{(2)}}{C_2}\Big), \dots\right) \\
&= \left(\frac{\Pi_{\mathcal{X}_1}(y_{t+1}^{(1)})}{C_1}, \frac{\Pi_{\mathcal{X}_2}(y_{t+1}^{(2)})}{C_2}, \dots\right) \qquad \text{(dilation property of projections [2, Prop. 3.2.3])} \\
&= \left(\frac{x_{t+1}^{(1)}}{C_1}, \frac{x_{t+1}^{(2)}}{C_2}, \dots\right) \qquad \text{(by Algorithm 1)}
\end{aligned}$$

This proves (13) for x_{t+1}, completing the induction, and completing the proof of P7.

▷ Claim 28. For $i \geq 1$, let $\mathcal{Y}_i \subseteq \mathbb{R}^{T_i}$ be a closed, convex set containing 0. Let $\mathcal{Y} = \prod_{i=1}^{\infty} \mathcal{Y}_i$. Then we have $\Pi_{\mathcal{Y}}(z) = \big(\Pi_{\mathcal{Y}_1}(z^{[1]}), \Pi_{\mathcal{Y}_2}(z^{[2]}), \dots\big)$ for all $z \in \ell_2$.

Proof. This follows from [3, Proposition 23.31]. ◀

6.3 Proof of Claim 27

Claim 27 follows easily from the following general lemma.

Lemma 29. *Let $h : \ell_2 \to \mathbb{R}$ be defined as $h(y^{[1]}, y^{[2]}, \dots) = \sum_{n=1}^{\infty} h_n(y^{[n]})$ where each $h_n : \mathbb{R}^{T_n} \to \mathbb{R}$ is a convex function. Then*

$$\partial h(y^{[1]}, y^{[2]}, \dots) \subseteq \prod_{i \geq 1} \partial h_i(y^{[i]}) \qquad \forall\, y \in \ell_2. \tag{16}$$

Moreover, if $\sum_{i \geq 1} \|g^{[i]}\| < \infty$ for all $y = (y^{[1]}, y^{[2]}, \dots) \in \mathcal{Y} \subseteq \ell_2$ and all $g^{[i]} \in \partial h_i(y^{[i]})$, then

$$\partial h(y^{[1]}, y^{[2]}, \dots) \supseteq \prod_{i \geq 1} \partial h_i(y^{[i]}) \qquad \forall\, y \in \mathcal{Y}. \tag{17}$$

Proof (of Claim 27). We simply apply Lemma 29 with $h_i = \frac{1}{C_i^2} f^{(i)} \circ C_i I_i$ where I_i is the identity map in $\mathbb{R}^{T_i}$, $\mathcal{Y} = \mathcal{X} = \prod_{i \geq 1} \mathcal{X}_i / C_i$, and $h = f = \sum_{i \geq 1} f^{(i)}$. Clearly h_i is convex. Using earlier conclusions from Lemma 26, we know that h is well-defined on ℓ_2 by P2 and $\mathcal{Y} \subset \ell_2$ by P1. Lastly, consider any $y = (y^{[1]}, y^{[2]}, \dots) \in \mathcal{Y}$ and $g^{[i]} \in \partial h_i(y^{[i]})$. By Claim 33 we have $\partial h_i(y^{[i]}) = \frac{1}{C_i} \partial f^{(i)}(C_i y^{[i]})$. By definition of $\mathcal{X}$ we have $C_i y^{[i]} \in \mathcal{X}_i$. Since $f^{(i)}$ is L-Lipschitz on $\mathcal{X}_i$, it follows that $\|g^{[i]}\| \leq L/C_i$, and so $\sum_{i \geq 1} \|g^{[i]}\| \leq L$. Thus all hypotheses of Lemma 29 are satisfied.

Applying the lemma, for every $x \in \mathcal{X}$, we have

$$\partial f(x) = \prod_{i=1}^{\infty} \partial h_i(x^{[i]}) = \prod_{i=1}^{\infty} \frac{1}{C_i} \partial f_i(C_i x^{[i]}),$$

by Lemma 29 and Claim 33. ◀

The next proof is similar to an argument in Bauschke and Combettes [3, Proposition 16.8], although their setting is simpler since they consider functions with only finitely many components.

Proof (of Lemma 29). First we prove (16). Consider any $g = (g^{[1]}, g^{[2]}, \ldots) \in \partial h(y)$. We must show that $g^{[i]} \in \partial h_i(y^{[i]})$ for all i. For any $z \in \mathbb{R}^{T_i}$, we may define $\widetilde{y} = (y^{[1]}, \ldots, y^{[i-1]}, z, y^{[i+1]}, \ldots)$. Clearly $\widetilde{y} \in \ell_2$. Since g is a subgradient of h at y, we have $h(\widetilde{y}) - h(y) \geq \langle \widetilde{y} - y, g \rangle$. Since y and $\widetilde{y}$ agree except on the i^{th} component, this inequality is equivalent to $h_i(z) - h_i(y^{[i]}) \geq \langle z - y^{[i]}, g^{[i]} \rangle$. Since z is arbitrary, this implies that $g^{[i]} \in \partial h_i(y^{[i]})$ as desired.

Next consider any $y \in \mathcal{Y}$.

$$\begin{aligned}
g \in \prod_{i \geq 1} \partial h_i(y^{[i]}) \quad &\Rightarrow \quad \langle y^{[i]} - \widetilde{y}^{[i]}, g^{[i]} \rangle + h_i(y^{[i]}) \leq h_i(\widetilde{y}^{[i]}) \qquad \forall\, i \in \mathbb{N},\ \forall\, \widetilde{y} \in \ell_2 \\
&\Rightarrow \quad \sum_{i \geq 1} \langle y^{[i]} - \widetilde{y}^{[i]}, g^{[i]} \rangle + \sum_{i \geq 1} h_i(y^{[i]}) \leq \sum_{i \geq 1} h_i(\widetilde{y}^{[i]}) \qquad \forall\, \widetilde{y} \in \ell_2 \\
&\Leftrightarrow \quad \langle y - \widetilde{y}, g \rangle + h(y) \leq h(\widetilde{y}) \qquad \forall\, \widetilde{y} \in \ell_2 \\
&\Leftrightarrow \quad g \in \partial h(y)
\end{aligned}$$

Here the second implication uses that $\sum_{i \geq 1} \langle y^{[i]} - \widetilde{y}^{[i]}, g^{[i]} \rangle$ is absolutely convergent by Cauchy–Schwarz:

$$\sum_{i \geq 1} \langle y^{[i]} - \widetilde{y}^{[i]}, g^{[i]} \rangle \;\leq\; \|y - \widetilde{y}\| \sum_{i \geq 1} \left\| g^{[i]} \right\|.$$

This proves (17) due to the assumption that the last sum is finite. ◀

7 Future work

7.1 Other step sizes

Theorem 1 requires a step size of the form c/t, where c is a constant. Similarly, Theorem 4 requires a step size of the form $c/\sqrt{t}$. It is conceivable that these theorems could respectively be generalized to accommodate step sizes with the asymptotic behaviour $\Theta(1/t)$ and $\Theta(1/\sqrt{t})$. To date we have not been able to prove such a generalization, so we leave it for future work.

A Standard or Elementary Results

▷ Claim 30. For $1 \leq a \leq b$, $\sum_{k=a}^{b} \frac{1}{\sqrt{k}} \leq 2\frac{b-a+1}{\sqrt{b}}$.

Proof.

$$\sum_{k=a}^{b} \frac{1}{\sqrt{k}} \;\leq\; \int_{a-1}^{b} \frac{1}{\sqrt{x}}\, dx \;=\; 2(\sqrt{b} - \sqrt{a-1}) \;=\; 2\frac{b-a+1}{\sqrt{b} + \sqrt{a-1}} \;\leq\; 2\frac{b-a+1}{\sqrt{b}}.$$ ◀

▷ Claim 31. For any $1 \leq j \leq t \leq T$, we have $\frac{t-j}{(T-j+1)\sqrt{t}} \leq \frac{1}{\sqrt{T}}$.

Proof. The function $g(x) = \frac{x-j}{\sqrt{x}}$ has derivative

$$g'(x) = \frac{1}{\sqrt{x}}\Big(1 - \frac{x-j}{2x}\Big) = \frac{1}{\sqrt{x}}\Big(\frac{1}{2} + \frac{j}{2x}\Big).$$

This is positive for all $x > 0$ and $j \geq 0$, and so

$$\frac{t-j}{\sqrt{t}} \;\leq\; \frac{T-j}{\sqrt{T}},$$

for all $0 < t \leq T$. This implies the claim. ◀

▷ Claim 32. Assume $0 \leq k$ and $k+1 \leq m$.

$$\sum_{\ell=k+1}^{m} \frac{1}{\ell^2} \;\leq\; \frac{1}{k} - \frac{1}{m}.$$

Proof. The sum may be upper-bounded by an integral as follows:

$$\sum_{\ell=k+1}^{m} \frac{1}{\ell^2} \;\le\; \int_k^m \frac{1}{x^2}\,dx \;=\; \frac{1}{k} - \frac{1}{m}. \qquad \blacktriangleleft$$

▷ Claim 33 ([7, Theorem VI.4.2.1]). Let $A : \mathbb{R}^n \to \mathbb{R}^m$ be a linear map and let g be a finite convex function on $\mathbb{R}^m$. Then $\partial(g \circ A)(x) = A^{\mathsf{T}} \partial g(Ax)$ for all $x \in \mathbb{R}^m$.

▷ Claim 34. Suppose that g and ϕ are positive functions satisfying $g(x) = o(\phi(x))$. Then we may write $g(x) = o(\phi(x)/h(x))$ for some positive function h satisfying $\lim_{x\to\infty} h(x) = \infty$.

Proof. Let $h(x) = \sqrt{\phi(x)/g(x)}$. Then $\lim_{x\to\infty} h(x) = \infty$ because $g = o(\phi(x))$. We have

$$\lim_{x\to\infty} \frac{g(x)}{\phi(x)/h(x)} \;=\; \lim_{x\to\infty} \sqrt{\frac{g(x)}{\phi(x)}} \;=\; 0,$$

because $g(x) = o(\phi(x))$. ◀

References

1 Viorel Barbu and Teodor Precupanu. *Convexity and optimization in Banach spaces*. Springer, 2012.

2 Heinz H. Bauschke. *Projection Algorithms and Monotone Operators*. PhD thesis, Simon Fraser University, 1996.

3 Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

4 Elad Hazan. Introduction to Online Convex Optimization. *Found. Trends Optim.*, 2(3–4), 2015.

5 Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Mach. Learn.*, 69(2-3):169–192, 2007.

6 Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *J. Mach. Learn. Theory*, 15(1):2489–2512, 2014.

7 Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer, 1996.

8 Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Making the Last Iterate of SGD Information Theoretically Optimal. In *Conference on Learning Theory*, pages 1752–1755, 2019.

9 Simon Lacoste-Julien, Mark W. Schmidt, and Francis R. Bach. A simpler approach to obtaining an $O(1/t)$ convergence rate for the projected stochastic subgradient method. `https://arxiv.org/abs/1212.2002`, 2012.

10 Arkadi S. Nemirovsky and David B. Yudin. *Problem complexity and method efficiency in optimization*. John Wiley & Sons, 1983.

11 Yurii Nesterov and Vladimir Shikhman. Quasi-monotone Subgradient Methods for Nonsmooth Convex Minimization. *J. Optim. Theory Appl.*, 165(3):917–940, 2015.

12 Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, 1992.

13 Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of ICML*, 2012.

14 R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

15 David Ruppert. Efficient estimations from a slowly convergent Robbins-Monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

16 Ohad Shamir. Open Problem: Is Averaging Needed for Strongly Convex Stochastic Gradient Descent? In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 47.1–47.3. 2012.

17 Ohad Shamir and Tong Zhang. Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 71–79. 2013.