On the Iteration Complexity of Oblivious First-Order Optimization Algorithms

Yossi Arjevani YOSSI.ARJEVANI@WEIZMANN.AC.IL
Weizmann Institute of Science, Rehovot 7610001, Israel

Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Weizmann Institute of Science, Rehovot 7610001, Israel

Abstract

We consider a broad class of first-order optimization algorithms which are *oblivious*, in the sense that their step sizes are scheduled regardless of the function under consideration, except for limited side-information such as smoothness or strong convexity parameters. With the knowledge of these two parameters, we show that any such algorithm attains an iteration complexity lower bound of $\Omega(\sqrt{L/\epsilon})$ for L-smooth convex functions, and $\tilde{\Omega}(\sqrt{L/\mu}\ln(1/\epsilon))$ for L-smooth μ-strongly convex functions. These lower bounds are stronger than those in the traditional oracle model, as they hold independently of the dimension. To attain these, we abandon the oracle model in favor of a structure-based approach which builds upon a framework recently proposed in (Arjevani et al., 2015). We further show that without knowing the strong convexity parameter, it is impossible to attain an iteration complexity better than $\tilde{\Omega}\left((L/\mu)\ln(1/\epsilon)\right)$. This result is then used to formalize an observation regarding L-smooth convex functions, namely, that the iteration complexity of algorithms employing time-invariant step sizes must be at least $\Omega(L/\epsilon)$.

1. Introduction

The ever-increasing utility of mathematical optimization in machine learning and other fields has led to a great interest in understanding the computational boundaries of solving optimization problems. Of a particular interest is the class of unconstrained smooth, and possibly strongly convex, optimization problems. Formally, we consider the problem of $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$ where $f : \mathbb{R}^d \to \mathbb{R}$ is convex and L-*smooth*, i.e., $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x}-\mathbf{y}\|$ for some $L > 0$, and possibly μ-*strongly convex*, that is, $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x})\rangle + \frac{\mu}{2}\|\mathbf{y}-\mathbf{x}\|^2$ for some $\mu > 0$. In this work, we address questions regarding how fast can one expect to solve this sort of problems to a prescribed level of accuracy, using methods which are based on first-order information (gradients, or more generally sub-gradients) alone.

The standard approach to quantify the computational hardness of optimization problems is through the *oracle model*. In this approach, one models the interaction of a given optimization algorithm with some instance from a class of functions as a sequence of queries, issued by the algorithm, to an external first-order oracle procedure. Upon receiving a query point $\mathbf{x} \in \mathbb{R}^d$, the oracle reports the corresponding value $f(\mathbf{x})$ and gradient $\nabla f(\mathbf{x})$. In their seminal work, Nemirovsky and Yudin (1983) showed that for any first-order optimization algorithm, there exists an L-smooth and μ-strongly convex function $f : \mathbb{R}^d \to \mathbb{R}$ such that the number of queries required to obtain an ϵ-*optimal* solution $\tilde{\mathbf{x}}$ which satisfies

$$f(\tilde{\mathbf{x}}) < \min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) + \epsilon,$$

is at least[1]

$$\begin{aligned} &\tilde{\Omega}\left(\min\left\{d, \sqrt{\kappa}\right\}\ln(1/\epsilon)\right), && \mu > 0 \qquad (1)\\ &\tilde{\Omega}(\min\{d\ln(1/\epsilon), \sqrt{L/\epsilon}\}), && \mu = 0 \end{aligned}$$

where $\kappa := L/\mu$ is the so-called *condition number*. This lower bound, although based on information considerations alone, is tight. Concretely, it is achieved by a combination of Nesterov's well-known accelerated gradient descent

Proceedings of the 33^{rd} *International Conference on Machine Learning*, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

[1] Following standard conventions, here, tilde notation hides logarithmic factors in the smoothness parameter, the strong convexity parameter and the distance of the initialization point from the minimizer.

(AGD, (Nesterov, 1983)) with an iteration complexity of

$$\tilde{\mathcal{O}}\left(\sqrt{\kappa}\ln(1/\epsilon)\right), \qquad \mu > 0 \qquad (2)$$
$$\mathcal{O}\left(\sqrt{L/\epsilon}\right), \qquad \mu = 0,$$

and the center of gravity method (MCG, (Levin, 1965; Newman, 1965)) whose iteration complexity is $\mathcal{O}(d\ln(1/\epsilon))$.

Although the combination of MCG and AGD appear to achieve optimal iteration complexity, this is not the case when focusing on *computationally efficient* algorithms. In particular, the per-iteration cost of MCG scales poorly with the problem dimension, rendering it impractical for high-dimensional problems. In other words, not taking into account the computational resources needed for processing first-order information limits the ability of the oracle model to give a faithful picture of the complexity of optimization.

To overcome this issue (Arjevani et al., 2015) recently proposed the framework of p-Stationary Canonical Linear Iterative (p-SCLI) in which, instead of modeling the way algorithms acquire information on the function at hand, one assumes certain dynamics which restricts the way new iterates are being generated. This framework includes a large family of computationally efficient first-order algorithms, whose update rule, when applied on quadratic functions, reduce to a recursive application of some fixed linear transformation on the most recent p points (in other words, p indicates the number of previous iterates stored by the algorithm in order to compute a new iterate). The paper showed that the iteration complexity of p-SCLIs over smooth and strongly convex functions is bounded from below by

$$\tilde{\Omega}\left(\sqrt[p]{\kappa}\ln(1/\epsilon)\right). \qquad (3)$$

Crucially, as opposed to the classical lower bounds in (1), the lower bound in (3) holds for any dimension $d > 1$. This implies that even for fixed d, the iteration complexity of p-SCLI algorithms must scale with the condition number. That being said, the lower bound in (3) raises a few major issues which we wish to address in this work:

- Practical first-order algorithms in the literature only attain this bound for $p = 1, 2$ (by standard gradient descent and AGD, respectively), so the lower bound appears intuitively loose. Nevertheless, (Arjevani et al., 2015) showed that this bound is actually tight for all p. The reason for this discrepancy is that the bound for $p > 2$ was shown to be attained by p-SCLI algorithms whose updates require exact knowledge of spectral properties of the Hessian, which is computationally prohibitive to obtain in large-scale problems. In this work, we circumvent this issue by systematically considering the *side-information* available to the algorithm. In particular, we show that under the realistic assumption, that the algorithm may only utilize the strong convexity and smoothness of the objective function, the lower bound in (3) can be substantially improved.
- The lower bound stated above is limited to *stationary* optimization algorithms whose coefficients α_j, β_j are not allowed to change in time (see Section 2.2).
- The formulation suggested in (Arjevani et al., 2015) does not allow generating more than one iterate at a time. This requirement is not met by many popular optimization problems for finite sums minimization.
- Lastly, whereas the proofs in (Arjevani et al., 2015) are elaborate and technically complex, the proofs we provide here are relatively short and simple.

In its simplest form, the framework we consider is concerned with algorithms which generate iterates by applying the following simple update rule repeatedly:

$$\mathbf{x}^{(k+1)} = \sum_{j=1}^{p} \alpha_j \nabla f(\mathbf{x}^{(k+1-j)}) + \beta_j \mathbf{x}^{(k+1-j)}, \qquad (4)$$

where $\alpha_j, \beta_j \in \mathbb{R}$ denote the corresponding coefficients. A clear advantage of this class of algorithms is that, given the corresponding gradients, the computational cost of executing each update rule scales linearly with the dimension of the problem and p.

This basic formulation already subsumes popular first-order optimization algorithms. For example, at each iteration the Gradient Descent (GD) method generates a new iterate by computing a linear combination of the current iterate and the gradient of the current iterate, i.e.,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \nabla f(\mathbf{x}^{(k)}) \qquad (5)$$

for some real scalar α. Another important example is a stationary variant of AGD (Nesterov, 2004) and the heavy-ball method (e.g., (Polyak, 1987)) which generates iterates according to

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \beta_1 \mathbf{x}^{(k)} + \alpha_1 \nabla f(\mathbf{x}^{(k)}) \\ &\quad + \beta_2 \mathbf{x}^{(k-1)} + \alpha_2 \nabla f(\mathbf{x}^{(k-1)}). \end{aligned} \qquad (6)$$

In this paper, we follow a generalized form of (4) which is exhibited by standard optimization algorithms: GD, conjugate gradient descent, sub-gradient descent, AGD, the heavy-ball method, coordinate descent, quasi-Newton methods, ellipsoid method, etc. The main difference being how much effort one is willing to put in computing the coefficients of the optimization process. We call these methods first-order p-Canonical Linear Iterative optimization algorithms (in this paper, abbr. p-CLI). We note that our framework (as a method to prove lower bounds) also applies to stochastic algorithms, as long as the expected update rule (conditioned on the history) follows a generalized form similar to (4).

In the context of machine learning, many algorithms for minimizing finite sums of functions with, possibly, a regularization term (also known as, Regularized Empirical Risk Minimization) also fall into our framework, e.g., Stochastic Average Gradient (SAG, (Schmidt et al., 2013)), Stochastic Variance Reduction Gradient (SVRG, (Johnson & Zhang, 2013)), Stochastic Dual Coordinate Ascent (SDCA, (Shalev-Shwartz & Zhang, 2013)), Stochastic Dual Coordinate Ascent without Duality (SDCA without duality, (Shalev-Shwartz, 2015)) and SAGA (Defazio et al., 2014), to name a few, and as such, are subject to the same lower bounds established through this framework.

In its full generality, the formulation of this framework is too rich to say much. In what follows, we shall focus on *oblivious* p-CLIs, which satisfy the realistic assumption that the coefficients α_j, β_j do not depend on the specific function under consideration. Instead, they can only depend on time and some limited side-information on the function (this term will be made more precise in Definition 1). In particular, we show that the iteration complexity of oblivious p-CLIs over L-smooth and μ-strongly convex functions whose coefficients are allowed to depend on μ and L is

$$\begin{aligned} &\tilde{\Omega}\left(\sqrt{\kappa}\ln(1/\epsilon)\right), && \mu > 0 \\ &\tilde{\Omega}(\sqrt{L/\epsilon}), && \mu = 0. \end{aligned} \tag{7}$$

Note that, in addition to being *dimension-independent* (similarly to (3)), this lower bound holds regardless of p. We further stress that the algorithms discussed earlier which attain the lower bound stated in (3) are not oblivious and require more knowledge of the objective function.

In the paper, we also demonstrate other cases where the side-information available to the algorithm crucially affects its performance, such as knowing vs. not knowing the strong convexity parameter.

Finally, we remark that this approach of modeling the structure of optimization algorithms, as opposed to the more traditional oracle model, can be also found in (Polyak, 1987; Lessard et al., 2014; Flammarion & Bach, 2015; Drori, 2014). However, whereas these works are concerned with upper bounds on the iteration complexity, in this paper we primarily focus on lower bounds.

To summarize, our main contributions are the following:

- In Section 2.1, we propose a novel framework which substantially generalizes the framework introduced in (Arjevani et al., 2015), and includes a large part of modern first-order optimization algorithms.
- In Section 2.2, we identify within this framework the class of oblivious optimization algorithms, whose step sizes are scheduled regardless of the function at hand, and provide an iteration complexity lower bound as given in (7). We improve upon (Arjevani et al., 2015) by establishing lower bounds which hold both for smooth functions and smooth and strongly convex functions, using simpler and shorter proofs. Moreover, in addition to being *dimension-independent*, the lower bounds we derive here are tight. In the context of machine learning optimization problems, the same lower bound is shown to hold on the bias of methods for finite sums with a regularization term, such as: SAG, SAGA, SDCA without duality and SVRG.
- Some oblivious algorithms for L-smooth and μ-strongly convex functions admit a linear convergence rate using step sizes which are scheduled regardless of the strong convexity parameter (e.g., standard GD with a step size of $1/L$. See Section 3 in (Schmidt et al., 2013) and Section 5 in (Defazio et al., 2014)). In Section 4.1, we show that adapting to 'hidden' strong convexity, without explicitly incorporating the strong convexity parameter, results in an inferior iteration complexity of
$$\tilde{\Omega}\left(\kappa\ln(1/\epsilon)\right). \tag{8}$$
This result sheds some light on a major issue regarding scheduling step sizes of optimization algorithms.
- In Section 4.2, we discuss the class of stationary optimization algorithms, which use time-invariant step sizes, over L-smooth functions and show that they admit a tight iteration complexity of
$$\Omega(L/\epsilon). \tag{9}$$
In particular, this bound implies that in terms of dependency on the accuracy parameter ϵ, SAG and SAGA admit an optimal iteration complexity w.r.t. the class of stochastic stationary p-CLIs. Acceleration schemes, such as (Frostig et al., 2015; Lin et al., 2015), are able to break this bound by re-scheduling these algorithms in a non-stationary (though oblivious) way.

2. Framework

2.1. Definitions

In the sequel we present our framework for analyzing first-order optimization algorithms. We begin by providing a precise definition of a class of optimization problems, accompanied by some side-information. We then formally define the framework of p-CLI algorithms and the corresponding iteration complexity.

Definition 1 (Class of Optimization Problems)**.** *A class of optimization problems $\mathcal{C}$ is an ordered pair of $(\mathcal{F}, I)$, where $\mathcal{F}$ is a family of functions which defined over the same domain, and $I : \mathcal{F} \to \mathfrak{I}$ is a mapping which provides for each $f \in \mathcal{F}$ the corresponding side-information element in some set $\mathfrak{I}$. The domain of the functions in $\mathcal{F}$ is denoted by $dom(\mathcal{C})$.*

For example, let us consider quadratic functions of the form $\mathbf{x} \mapsto \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{q}^\top\mathbf{x}$, where $Q \in \mathbb{R}^{d\times d}$ is a positive semidefinite matrix whose spectrum lies in $\Sigma \subseteq \mathbb{R}^+$, and $\mathbf{q} \in \mathbb{R}^d$. Here, each instance may be accompanied with either a complete specification of Σ; lower and upper bounds for Σ; just an upper bound for Σ; a rough approximation of Q^{-1} (e.g., sketching techniques), etc. We will see that the exact nature of side-information strongly affects the iteration complexity, and that this differentiation between the family of functions under consideration and the type of side-information is not mere pedantry, but a crucial necessity.

We now turn to rigorously define first-order p-CLI optimization algorithms. The basic formulation shown in (4) does not allow generating more than one iterate at a time. The framework which we present below relaxes this restriction to allow a greater generality which is crucial for incorporating optimization algorithms for finite sums (see Stochastic p-CLIs in Section 2.2). We further extend (4) to allow non-differentiable functions and constraints into this framework, by generalizing gradients to sub-gradients.

Definition 2. *[First-order p-CLI] An optimization algorithm is called a first-order p-Canonical Linear Iterative (p-CLI) optimization algorithm over a class of optimization problems $\mathcal{C} = (\mathcal{F}, I(\cdot))$, if given an instance $f \in \mathcal{F}$ and an arbitrary set of p initialization points $\mathbf{x}_1^0, \ldots, \mathbf{x}_p^0 \in dom(\mathcal{C})$, it operates by iteratively generating points for which*

$$\mathbf{x}_i^{(k+1)} \in \sum_{j=1}^{p} \left(A_{ij}^{(k)}\partial f + B_{ij}^{(k)}\right)(\mathbf{x}_j^{(k)}), \quad k = 0, 1, \ldots \tag{10}$$

holds, where the coefficients A_{ij}^k, B_{ij}^k are some linear operators which may depend on $I(f)$.

Formally, the expression $A_{ij}^{(k)}\partial f$ in (10) denotes the composition of $A_{ij}^{(k)}$ and the sub-gradient operator. Likewise, the r.h.s. of (10) is to be understand as an evaluation of sum of two operators $A_{ij}^{(k)}\partial f$ and $B_{ij}^{(k)}$ at $\mathbf{x}_j^{(k)}$.

In this level of generality, this framework encompasses very different kinds of optimization algorithms. We shall see that various assumptions regarding the coefficients complexity and side-information yield different lower bound on the iteration complexity.

We note that although this framework concerns algorithms whose update rules are based on a fixed number of points, a large part of the results shown in this paper holds in the case where p grows indefinitely in accordance with the number of iterations.

We now turn to provide a formal definition of *iteration complexity*. We assume that the point returned after k iterations is $\mathbf{x}_p^{(k)}$. This assumption merely serves as a convention and is not necessary for our bounds to hold.

Definition 3 (Iteration Complexity)**.** *The iteration complexity of a given p-CLI w.r.t. a given problem class $\mathcal{C} = (\mathcal{F}, I)$ is defined to be the minimal number of iterations K such that*

$$f(\mathbb{E}\mathbf{x}_p^{(k)}) - \min_{\mathbf{x}\in dom\mathcal{C}} f(\mathbf{x}) < \epsilon, \quad \forall k \geq K$$

uniformly over $\mathcal{F}$, where the expectation is taken over all the randomness introduced into the optimization process.

For simplicity, when stating bounds in this paper, we shall omit the dependency of the iteration complexity on the initialization points. The precise dependency can be found in the corresponding proofs.

2.2. Classification of First-order p-CLIs and Scope of Work

As mentioned before, we cannot say much about the framework in its full generality. In this paper, we restrict our attention to the following three (partially overlapping) classes of p-CLIs:

Stationary p-CLI where the coefficients are allowed to depend exclusively on side-information (see Definition 3). In particular, the coefficients are not allowed to change with time. Seemingly restrictive, this class of p-CLIs subsumes many efficient optimization methods, especially when coupled with stochasticity (see below). Notable stationary p-CLIs are: GD with fixed step size (Nesterov, 2004), stationary AGD (Nesterov, 2004) and the Heavy-Ball method (Polyak, 1987).

Oblivious p-CLI where the coefficients are allowed to depend on side-information, as well as to change in time. Notable algorithms here are GD and AGD with step sizes which are scheduled irrespectively of the function under consideration (Nesterov, 2004) and the Sub-Gradient Descent method (e.g., (Shor, 2012)).

Stochastic p-CLI where (10) holds with respect to $\mathbb{E}\mathbf{x}_j^{(k)}$, that is,

$$\mathbb{E}\mathbf{x}_i^{(k+1)} \in \sum_{j=1}^{p} \left(A_{ij}^{(k)}\partial f + B_{ij}^{(k)}\right)(\mathbb{E}\mathbf{x}_j^{(k)}). \tag{11}$$

Stochasticity is an efficient machinery of tackling optimization problems where forming the gradient is prohibitive, but engineering an efficient unbiased estimator is possible. Such situations occur frequently in the context of machine learning, where one is interested in minimizing finite sums of large number of convex functions,

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := \sum_{i=1}^{m} f_i(\mathbf{x}),$$

in which case, forming a sub-gradient of F at a given point may be too expensive. Notable optimization algorithms for variants of this problem are: SAG, SDCA without duality, SVRG and SAGA, all of which are stationary stochastic p-CLIs. Moreover, as opposed to algorithms which produce only one new point at each iteration (e.g., (4)), these algorithms sometimes update a few points at the same time. To illustrate this, let us express SAG as a stochastic stationary $(m+1)$-CLI. In order to avoid the computationally demanding task of forming the exact gradient of F at each iteration, SAG uses the first m points to store estimates for the gradients of the individual functions

$$\mathbf{y}_i \approx \nabla f_i(\mathbf{x}_{m+1}^{(k)}),\ i = 1\ldots m.$$

At each iteration, SAG sets $\mathbf{y}_i = \nabla f_i(\mathbf{x}_{m+1}^{(k)})$ for some randomly chosen $i \in [m]$, and then updates $\mathbf{x}_{m+1}^{(k)}$ accordingly, by making a gradient step with a fixed step size using the new estimate for $\nabla F(\mathbf{x}_{m+1}^{(k)})$. This implies that the expected update rule of SAG is stationary and satisfies (11).

As opposed to an oblivious schedule of step sizes, many optimization algorithms set the step sizes according to the first-order information which is accumulated during the optimization process. A well-known example for such a non-oblivious schedule is conjugate gradient descent, whose update rule can be expressed as follows:

$$\begin{aligned}\mathbf{x}_1^{(k+1)} &= \mathbf{x}_2^{(k)},\\ \mathbf{x}_2^{(k+1)} &= (\alpha\partial f + (1+\beta)I)\mathbf{x}_2^{(k)} - \beta\mathbf{x}_1^{(k)},\end{aligned} \tag{12}$$

where the step sizes are chosen so as to minimize $f(\mathbf{x}_1^{(k+1)})$ over $\alpha, \beta \in \mathbb{R}$. Other algorithms employ coefficients whose schedule does not depend directly on first-order information. For example, at each iteration coordinate descent updates one coordinate of the current iterate, by completely minimizing the function at hand along some direction. In our formulation, such update rules are expressed using coefficients which are diagonal matrices. In a sense, the most expensive coefficients used in practice are the one employed by Newton method, which in this framework, may be expressed as follows:

$$\mathbf{x}_1^{(k+1)} = (I - \nabla^2(f)^{-1}\nabla f)\mathbf{x}_1^{(k)} \tag{13}$$

The algorithms mentioned above: conjugate gradient descent, coordinate descent and Newton methods; as well as other non-oblivious p-CLI optimization algorithms, such as quasi-Newton methods (e.g., (Nocedal & Wright, 2006)) and the ellipsoid method (e.g., (Atallah, 1998)), will not be further considered in this paper.

3. Lower Bounds on the Iteration Complexity of Oblivious p-CLIs

Having formally defined the framework, we are now in position to state our first main result. Perhaps the most common side-information used by practical algorithms is the strong-convexity and smoothness parameters of the objective function. Oblivious p-CLIs with such side-information tend to have low per-iteration cost and a straightforward implementation. However, this lack of adaptivity to the function being optimized results in an inevitable lower bound on the iteration complexity:

Theorem 1. *Suppose the smoothness parameter L and the strong convexity parameter μ are known, i.e., $I(\cdot) = \{L, \mu\}$. Then the iteration complexity of any oblivious, possibly stochastic, p-CLI optimization algorithm is bounded from below by*

$$\begin{aligned}&\tilde{\Omega}\left(\sqrt{\kappa}\ln(1/\epsilon)\right), && \mu > 0\\ &\Omega(\sqrt{L/\epsilon}), && \mu = 0,\end{aligned} \tag{14}$$

where $\kappa := L/\mu$.

As discussed in the introduction, Theorem 1 significantly improves upon the lower obtained by (Arjevani et al., 2015) in 3 major aspects:

- It holds for both smooth functions, as well as smooth and strongly convex functions.
- In both strongly-convex and non-strongly convex cases, the bounds we derive are tight for $p > 1$ (Note that if the coefficients are scalars and time-invariant, then for smooth and strongly convex functions a better lower bound of $\tilde{\Omega}(\kappa\ln(1/\epsilon))$ holds. See Theorem 8, (Arjevani et al., 2015)).
- It considers a much wider class of algorithms, namely, methods which may use different step size at each iteration and may freely update each of the p points.

We stress again that, in contrast to (1), this lower bound does not scale with the dimension of the problem.

The proof of Theorem 1, including logarithmic factors and constants which appear in the lower bound, is found in (A.1), and can be roughly sketched as follows. First, we consider L-smooth and μ-strongly convex quadratic functions of the form

$$\mathbf{x} \mapsto \frac{\eta}{2}\mathbf{x}^\top \mathbf{x} + \eta \mathbf{1}^\top \mathbf{x}, \quad \eta \in [\mu, L],$$

over $\mathbb{R}^d$, all of which share the same minimizer, $\mathbf{x}^* = -\mathbf{1}$. Next, we observe that each iteration of p-CLI involves application of $A\partial f + B$, which is a linear expression in ∂f whose coefficients are some linear operators, on the current points $\mathbf{x}_j^{(k)}$, $j = 1, \ldots, p$, which are then summed up to form the next iterate. Applying this argument inductively, and setting the initialization points to be zero, we see that the point returned by the algorithm at the k'th iteration can be expressed as follows,

$$\mathbf{x}_p^{(k)} = (s_1(\eta)\eta, \ldots, s_d(\eta)\eta)^\top,$$

where $s_i(\eta)$ are real polynomials of degree $k-1$. Here, the fact that the coefficients are scheduled obliviously, i.e., *do not* depend on the very choice of η, is crucial (when analyzing other types of p-CLIs, one may encounter cases where the coefficients of $s(\eta)$ are not constants, in which case the resulting expression may not be a polynomial). Bearing in mind that our goal is to bound the distance to the minimizer $-\mathbf{1}$ (which, in this case, is equivalent to the iteration complexity up to logarithmic factors), we are thus led to ask how small can $|s(\eta)\eta + 1|$ be. Formally, we aim to bound

$$\max_{\eta \in [\mu, L]} |s(\eta)\eta + 1|$$

from below. To this end, we use the properties of the well-known Chebyshev polynomials, by which we derive the following lower bound:

$$\min_{s(\eta)\in\mathbb{R}[\eta], \partial(s)=k-1} \ \max_{\eta\in[\mu,L]} |s(\eta)\eta + 1| \geq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k.$$

The proof of the smooth non-strongly convex case is also based on a reduction from a minimization problem to a polynomial approximation problem, only this time the resulting approximation problem is slightly different (see Equation (21) in Appendix A.2).

The idea of reducing optimization bounds to polynomial approximation problems is not new, and is also found for instance in (Nemirovsky & Yudin, 1983), where lower bounds under the oracle model are derived. In particular, both approaches, the oracle model and p-CLI, exploit the idea that when applied on some strongly convex quadratic functions $\frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{q}^\top \mathbf{x}$ over $\mathbb{R}^d$, the k'th iterate can be expressed as $s(Q)\mathbf{q}$ for some real polynomial $s(\eta) \in \mathbb{R}[\eta]$ of degree at most $k-1$. Bounding the iteration complexity is then essentially reduced to the question of how well can we approximate Q^{-1} using such polynomials. However, the approach here uses a fundamentally different technique for achieving this, and whereas the oracle model does not impose any restrictions on the coefficients of $s(\eta)$, the framework of p-CLIs allows us to effectively control the way these coefficients are being produced. The excessive freedom in choosing $s(\eta)$ constitutes a major weakness in the oracle model and prevents obtaining iteration complexity bounds significantly larger than the dimension d. To see why, note that by the Cayley-Hamilton theorem, there exists a real polynomial $s(\eta)$ of degree at most $d-1$ such that $s(Q) = -Q^{-1}$. Therefore, the d'th iterate can potentially be $s(Q)\mathbf{q} = -Q^{-1}\mathbf{q}$, the exact minimizer. We avoid this limited applicability of the oracle model by adopting a more structural approach, which allows us to restrict the kind of polynomials which can be produced by practical optimization algorithms. Furthermore, our framework is more flexible in the sense that the coefficients of $s(\eta)$ may be formed by optimization algorithms which do not necessarily fall into the category of first-order algorithms, e.g., coordinate descent.

It is instructive to contrast our approach with another structural approach for deriving lower bounds which was proposed by (Nesterov, 2004). Nesterov (2004) considerably simplifies the technique employed by Nemirovsky and Yudin (1983) at the cost of introducing additional assumption regarding the way new iterates are generated. Specifically, it is assumed that each new iterate lies in the span of all the gradients acquired earlier. Similarly to (Nemirovsky & Yudin, 1983), this approach also does not yield dimension-independent lower bounds. Moreover, such an approach may break in presence of conditioning mechanisms (which essentially, aim to handle poorly-conditioned functions by multiplying the corresponding gradients by some matrix). In our framework, such conditioning is handled through non-scalar coefficients. Thus, as long as the conditioning matrices depend solely on μ, L our lower bounds remain valid.

4. Side-Information in Oblivious Optimization

4.1. No Strong Convexity Parameter, No Acceleration

Below we discuss the effect of not knowing exactly the strong convexity parameter on the iteration complexity of oblivious p-CLIs. In particular, we show that the ability of oblivious p-CLIs to obtain iteration complexity which scales like $\sqrt{\kappa}$ crucially depends on the quality of the strong convexity estimate of the function under consideration. Moreover, we show that stationary p-CLIs are strictly weaker than general oblivious p-CLIs for smooth non-strongly convex functions, in the sense that stationary p-CLIs cannot obtain an iteration complexity of $\mathcal{O}(\sqrt{L/\epsilon})$.

The fact that decreasing the amount of side-information increases the iteration complexity is best demonstrated by a family of quadratic functions which we already discussed before, namely,

$$\mathbf{x} \mapsto \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{q}^\top \mathbf{x},$$

where $Q \in \mathbb{R}^{d\times d}$ is positive semidefinite whose spectrum lies in $\Sigma \subseteq \mathbb{R}^+$ and $\mathbf{q} \in \mathbb{R}^d$. In Theorem 8 in (Arjevani et al., 2015), it is shown that if Q is given in advance, but $\mathbf{q}$ is unknown, then the iteration complexity of stationary p-CLIs which follows (4) is

$$\tilde{\Omega}(\sqrt[p]{\kappa}\ln(1/\epsilon)).$$

It is further shown that this lower bound is tight (see Appendix A in (Arjevani et al., 2015)). In Theorem 1 we show that if both the smoothness and the strong convexity parameters $\{\mu, L\}$ are known then the corresponding lower bound for this kind of algorithms is

$$\tilde{\Omega}(\sqrt{\kappa}\ln(1/\epsilon)).$$

As mentioned earlier, this lower bound is tight and is attained by a stationary version of AGD.

However, what if only the smoothness parameter L is known a-priori? The following theorem shows that in this case the iteration complexity is substantially worse. For reasons which will become clear later, it will be convenient to denote the strong convexity parameter and the condition number of a given function f by $\mu(f)$ and $\kappa(f)$, respectively.

Theorem 2. *Suppose that only L the smoothness parameter is known, i.e. $I(\cdot) = \{L\}$. If the iteration complexity of a given oblivious, possibly stochastic, p-CLI optimization algorithm is*

$$\tilde{O}(\kappa(f)^\alpha \ln(1/\epsilon)), \tag{15}$$

then $\alpha \geq 1$.

Theorem 2 pertains to the important issue of optimal schedules for step sizes. Concretely, it implies that, in the absence of the strong convexity parameter, one is still able to schedule the step sizes according to the smoothness parameter so as to obtain exponential convergence rate, but only to the limited extent of linear dependency on the condition number (as mentioned before, this sub-optimality in terms of dependence on the condition number, can be also found in (Schmidt et al., 2013) and (Defazio et al., 2014)). This bound is tight and is attained by standard gradient descent (GD).

Theorem 2 also emphasizes the superiority of standard GD in cases where the true strong convexity parameter is poorly estimated. Such situations may occur when one underestimate the true strong convexity parameter by following the strong convexity parameter introduced by an explicit regularization term. Specifically, if $\hat{\mu}$ denotes our estimate for the true strong convexity parameter μ (obviously, $\hat{\mu} < \mu$ to ensure convergence), then Theorem 1 already implies that, for a fixed accuracy level, the worst iteration complexity of our algorithm is on the order of $\sqrt{L/\hat{\mu}}$, whereas standard GD with $1/L$ step sizes has iteration complexity on the order of L/μ. Thus, if our estimate is too conservative, i.e., $\hat{\mu} < \mu^2/L$, then the iteration complexity of GD is $\mu/\sqrt{L\hat{\mu}} \geq 1$ times better. Theorem 2 further strengthen this statement, by indicating that if our estimate does not depend on the true strong convexity parameter, then the iteration complexity of GD is even more favorable with a factor of $\mu/\hat{\mu} \geq 1$, compared to our algorithm.

The proof of Theorem 2, which appears in Appendix A.2, is again based on a reduction to an approximation problem via polynomials. In contrast to the proof of Theorem 1 which employs Chebyshev polynomials, here only elementary algebraic manipulations are needed.

Another implication of Theorem 2 is that the coefficients of optimal stationary p-CLIs for smooth and strongly convex functions must have an explicit dependence on the strong convexity parameter. In the next section we shall see that this fact is also responsible for the inability of stationary p-CLIs to obtain a rate of $\mathcal{O}(\sqrt{L/\epsilon})$ for L-smooth convex functions.

4.2. No Acceleration for Stationary Algorithms over Smooth Convex Functions

Below, we prove that, as opposed to oblivious p-CLIs, stationary p-CLIs (namely, p-CLIs with time-invariant coefficients) over L-smooth convex functions can obtain an iteration complexity no better than $\mathcal{O}(L/\epsilon)$. An interesting implication of this is that some current methods for minimizing finite sums of functions, such as SAG and SAGA (which are in fact stationary p-CLIs) cannot be optimal in this setting, and that time-changing coefficients are essential to get optimal rates. This further motivates the use of current acceleration schemes (e.g., (Frostig et al., 2015; Lin et al., 2015)) which turn a given stationary algorithm into an non-stationary oblivious one.

The proof of this result is based on a reduction from the class of p-CLIs over L-smooth convex functions to p-CLIs over L-smooth and μ-strongly convex, where the strong convexity parameter is given explicitly. This reduction allows us to apply the lower bound in Theorem 2 on p-CLIs designed for smooth non-strongly convex functions.

We now turn to describe the reduction in detail. In his seminal paper, Nesterov (1983) presents the AGD algorithm and shows that it obtains a convergence rate of

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \frac{4L \left\|\mathbf{x}^0 - \mathbf{x}^*\right\|^2}{(k+2)^2} \tag{16}$$

for L-smooth convex functions, which admits at least one minimizer (accordingly, throughout the rest of this section we shall assume that the functions under consideration admit at least one minimizer, i.e., $\mathrm{argmin}(f) \neq \emptyset$). In addition, Nesterov proposes a restarting scheme of this algorithm which, assuming the strong convexity parameter is known, allows one to obtain an iteration complexity of $\tilde{\mathcal{O}}(\sqrt{\kappa}\ln(1/\epsilon))$. Scheme 4.2 shown below forms a simple generalization of the scheme discussed in that paper, and allows one to explicitly introduce a strong convexity parameter into the dynamics of (not necessarily oblivious) p-CLIs over L-smooth convex functions.

```
SCHEME 4.2    RESTARTING SCHEME

PARAMETERS    . SMOOTHNESS PARAMETER $L > 0$
              . STRONG CONVEXITY PARAMETER $\mu > 0$
              . CONVERGENCE PARAMETERS $\alpha > 0, C > 0$
GIVEN         A $p$-CLI OVER $L$-SMOOTH FUNCTIONS $\mathcal{P}$ WITH
                $f(\mathbf{x}^k) - f^* \leq \frac{CL\left\|\bar{\mathbf{x}}^0 - \mathbf{x}^*\right\|^2}{k^\alpha}$
                FOR ANY INITIALIZATION VECTOR $\bar{\mathbf{x}}^0$
ITERATE       FOR $t = 1, 2, \ldots$
                RESTART THE STEP SIZE SCHEDULE OF $\mathcal{P}$
                INITIALIZE $\mathcal{P}$ AT $\bar{\mathbf{x}}^0$
                RUN $\mathcal{P}$ FOR $\sqrt[\alpha]{4CL/\mu}$ ITERATIONS
                SET $\bar{\mathbf{x}}^0$ TO BE THE LAST ITERATE OF THIS EXECUTION
END
```

The following lemma provides an upper bound on the iteration complexity of p-CLIs obtained through Scheme 4.2.

Lemma 1. *The convergence rate of a p-CLI algorithm obtained by applying Scheme 4.2, using the corresponding set of parameters L, μ, C, α, is*

$$\tilde{\mathcal{O}}\big(\sqrt[\alpha]{\kappa}\ln(1/\epsilon)\big),$$

where $\kappa = L/\mu$ denotes the condition number.

Proof Suppose $\mathcal{P}$ is a p-CLI as stated in Scheme 4.2 and let f be a L-smooth and μ-strongly convex function. Each external iteration in this scheme involves running $\mathcal{P}$ for $k = \sqrt[\alpha]{4CL/\mu}$ iterations, Thus, for any arbitrary point $\bar{\mathbf{x}}$,

$$f(\mathbf{x}^{(k)}) - f^* \leq \frac{CL\left\|\bar{\mathbf{x}} - \mathbf{x}^*\right\|^2}{(\sqrt[\alpha]{4CL/\mu})^\alpha} = \frac{\left\|\bar{\mathbf{x}} - \mathbf{x}^*\right\|^2}{4/\mu}.$$

Also, f is μ-strongly convex, therefore

$$f(\mathbf{x}^{(k)}) - f^* \leq \frac{2(f(\bar{\mathbf{x}}) - f(\mathbf{x}^*))/\mu}{4/\mu} \leq \frac{f(\bar{\mathbf{x}}) - f(\mathbf{x}^*)}{2}.$$

That is, after each external iteration the sub-optimality in the objective value is halved. Thus, after T external iterations, we get

$$f(\mathbf{x}^{(T\sqrt[\alpha]{4CL/\mu})}) - f^* \leq \frac{f(\bar{\mathbf{x}}^0) - f(\mathbf{x}^*)}{2^T},$$

where $\bar{\mathbf{x}}^0$ denotes some initialization point. Hence, the iteration complexity for obtaining an ϵ-optimal solution is

$$\sqrt[\alpha]{4C\kappa}\log_2\left(\frac{f(\bar{\mathbf{x}}^0) - f(\mathbf{x}^*)}{\epsilon}\right).$$

■

The stage is now set to prove the statement made at the beginning of this section. Let $\mathcal{P}$ be a stationary p-CLI over L-smooth functions with a convergence rate of $\mathcal{O}(L/k^\alpha)$, and let $\mu \in (0, L)$ be the strong convexity parameter of the function to be optimized. We apply Scheme 4.2 to obtain a new p-CLI, which according to Lemma 1, admits an iteration complexity of $\mathcal{O}(\sqrt[\alpha]{\kappa}\ln(1/\epsilon))$. But, since $\mathcal{P}$ is stationary, the resulting p-CLI under Scheme 4.2 is again $\mathcal{P}$ (That is, stationary p-CLIs are invariant w.r.t. Scheme 4.2). Now, $\mathcal{P}$ is a p-CLI over smooth non-strongly convex, and as such, its coefficients do not depend on μ. Therefore, by Theorem 2, we get that $\alpha \leq 1$. Thus, we arrive at the following corollary:

Corollary 1. *If the iteration complexity of a given stationary p-CLI over L-smooth functions is $\mathcal{O}\Big(\sqrt[\alpha]{L/\epsilon}\Big)$, then $\alpha \leq 1$.*

The lower bound above is tight and is attained by standard Gradient Descent.

5. Summary

In this work, we propose the framework of first-order p-CLIs and show that it can be efficiently utilized to derive bounds on the iteration complexity of a wide class of optimization algorithms, namely, oblivious, possibly stochastic, p-CLIs over smooth and strongly-convex functions.

We believe that these results are just the tip of the iceberg, and the generality offered by this framework can be successfully instantiated for many other classes of algorithms. For example, it is straightforward to derive a lower bound of $\Omega(1/\epsilon)$ for 1-CLIs over 1-Lipschitz (possibly non-smooth) convex functions using the following set of functions

$$\left\{\left\|\mathbf{x} - \mathbf{c}\right\| \middle| \mathbf{c} \in \mathbb{R}^d\right\}.$$

How to derive a lower bound for other types of p-CLIs in the non-smooth setting is left to future work.

Acknowledgments

This research is supported in part by an FP7 Marie Curie CIG grant, the Intel ICRI-CI Institute, and Israel Science Foundation grant 425/13. We thank Nati Srebro for several helpful discussions and insights.

References

Arjevani, Yossi, Shalev-Shwartz, Shai, and Shamir, Ohad. On lower and upper bounds for smooth and strongly convex optimization problems. *arXiv preprint arXiv:1503.06833*, 2015.

Atallah, Mikhail J. *Algorithms and theory of computation handbook*. CRC press, 1998.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

Drori, Yoel. *Contributions to the Complexity Analysis of Optimization Algorithms*. PhD thesis, Tel-Aviv University, 2014.

Flammarion, Nicolas and Bach, Francis. From averaging to acceleration, there is only a step-size. *arXiv preprint arXiv:1504.01577*, 2015.

Frostig, Roy, Ge, Rong, Kakade, Sham M, and Sidford, Aaron. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. *arXiv preprint arXiv:1506.07512*, 2015.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Lessard, Laurent, Recht, Benjamin, and Packard, Andrew. Analysis and design of optimization algorithms via integral quadratic constraints. *arXiv preprint arXiv:1408.3595*, 2014.

Levin, A Yu. On an algorithm for the minimization of convex functions. In *Soviet Mathematics Doklady*, volume 160, pp. 1244–1247, 1965.

Lin, Hongzhou, Mairal, Julien, and Harchaoui, Zaid. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pp. 3366–3374, 2015.

Nemirovsky, AS and Yudin, DB. Problem complexity and method efficiency in optimization. 1983. *Willey-Interscience, New York*, 1983.

Nesterov, Yurii. *A method of solving a convex programming problem with convergence rate O (1/k2)*. @, 1983.

Nesterov, Yurii. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

Newman, Donald J. Location of the maximum on unimodal surfaces. *Journal of the ACM (JACM)*, 12(3):395–398, 1965.

Nocedal, Jorge and Wright, Stephen. *Numerical optimization*. Springer Science & Business Media, 2006.

Polyak, Boris T. *Introduction to optimization*. Optimization Software New York, 1987.

Schmidt, Mark, Roux, Nicolas Le, and Bach, Francis. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

Shalev-Shwartz, Shai. Sdca without duality. *arXiv preprint arXiv:1502.06177*, 2015.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

Shor, Naum Zuselevich. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.