Variance Reduction for Non-Convex Stochastic Optimization: General Analysis and New Applications

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

OPTIMIZATION & DECISION INTELLIGENCE

Variance Reduction for Non-Convex Stochastic Optimization: General Analysis and New Applications

Master Thesis

Liang Zhang

Thursday 9th September, 2021

Advisor: Prof. Dr. Niao He

Department of Computer Science, ETH Zürich

Abstract

We introduce a general algorithmic framework for smooth non-convex stochastic optimization based on variance reduction techniques. Our framework allows for flexible selections of batch sizes and stepsize, encapsulating many popular variance reduction algorithms designed for non-convex optimization as well as several new variants. When applying the general framework to non-convex biased stochastic optimization, the algorithms match the optimal complexity achieved by unbiased stochastic methods when the biased gradient estimates satisfy an average bias growth condition. We further apply the general framework to a wide spectrum of structured stochastic optimization problems, including stochastic bilevel optimization, stochastic minimax optimization and stochastic compositional optimization, yielding a rich family of near-optimal single-loop algorithms for each of these problems. When combined with stochastic mirror descent, our general framework also solves a class of non-smooth non-convex stochastic optimization problems with the optimal oracle complexity, providing the first convergence result without using large batch sizes in the non-Euclidean setting.

Contents

Chapter 1

Introduction

Non-convex optimization has attracted more and more attentions with the huge success of deep learning in various domains including image recognition [1], language understanding [2] and drug discovery [3]. One of the most important engines for solving non-convex optimization problems encountered in modern machine learning is the so-called stochastic gradient descent (SGD). For the stochastic optimization of the general form:

$$\min_{x\in\mathbb{R}^d} F(x) \coloneqq \mathbb{E}_\xi[f(x;\xi)],$$

where we consider the case when $F(x)$ is non-convex and smooth, SGD updates

$$x_{t+1} = x_t - \alpha\nabla f(x_t;\xi_t), \qquad t = 1,\cdots,T,$$

with the stepsize α and the stochastic gradient $\nabla f(x_t;\xi_t)$ returned by some first-order oracle at query point x_t. SGD requires at least $\mathcal{O}(\varepsilon^{-4})$ oracle calls [4] to find an ε-stationary point $\tilde{x}$ such that $\|\nabla F(\tilde{x})\| \le \varepsilon$.

Variance reduction has emerged recently as a powerful technique to improve the complexity result of SGD. After SVRG [5] and SAGA [6] achieved theoretically better convergence rate in the convex regime, several variance reduction methods have been proposed for non-convex optimization as well, for example SPIDER [7, 8], SARAH [9] and STORM [10]. These methods have been proved to be effective and can obtain the $\mathcal{O}(\varepsilon^{-3})$ sample complexity to guarantee an ε-stationary point, which is optimal when assuming average smoothness [11]. In this work, we introduce the following general framework for a unified analysis of different variance reduction methods in the non-convex regime.

General Framework:

$$x_{t+1} = x_t - \alpha h_t, \qquad t = 1,...,T,$$

where

$$h_t = \begin{cases} \frac{1}{D}\sum_{i=1}^{D}\nabla f(x_t;\xi_t^i), & \text{if } t \equiv 0 \pmod{Q};\\ (1-\eta)\Big(h_{t-1} - \frac{1}{S}\sum_{i=1}^{S}\nabla f(x_{t-1};\xi_t^i)\Big) + \frac{1}{S}\sum_{i=1}^{S}\nabla f(x_t;\xi_t^i), & \text{otherwise.}\end{cases}$$

In the above framework, Q is the epoch length, D and S are batch sizes, α is the stepsize, $\eta \in [0,1]$ is a momentum parameter, and $\{\xi_t^i\}$ are independent and identically distributed samples. The framework runs in epochs and at the beginning of each epoch, it takes a potentially large batch size D to compute a mini-batch estimator of the true gradient (referred to as checkpoint gradients); and a possibly smaller batch size S at other iterations to compute a recursive gradient with control variate technique.

The previous analysis for SGD and variance reduction methods relys on the construction of unbiased gradient estimators. However, recent years have witnessed a dramatic increase in machine learning applications where unbiased gradient estimators are costly to obtain, e.g., solving Bellman equations in reinforcement learning [12], Wasserstein robust models [13], robust learning over multiple domains [14], meta-learning [15], and hyper-parameter optimization [16], to name a few. Notably, these applications can often be formulated as some structured stochastic optimization such as stochastic bilevel optimization [17], stochastic minimax optimization [18], stochastic compositional optimization [19], conditional stochastic optimization [20], etc.

One common feature of these structured stochastic optimization problems is that the bias in gradient estimators comes from either an estimation or an optimization subproblem. Hence, the bias can often be controlled. In such cases, it is natural to design biased gradient methods based on two different philosophies. The first one is to enforce small bias within targeted accuracy at every iteration, which in general requires a double-loop algorithm [21, 22]; the other is to adaptively reduce the bias of the gradient estimator, which typically results in a simpler single-loop algorithm. However, existing single-loop algorithms heavily rely on a case-by-case construction, whose convergence rates are sometimes even sub-optimal [17, 18, 19].

Based on these observations, one may ask the question that if there exist optimal single-loop algorithms for the aforementioned non-convex structured stochastic optimization problems. To address the issues, we extend the proposed general framework for variance reduction methods to biased stochastic settings where we only have access to some (possibly white-box) stochastic oracle that gives a biased estimate $\bar{\nabla} f(x;\xi)$ of the true gradient $\nabla F(x)$ at every query point x. Our results suggest that the optimal complexity $\mathcal{O}(\varepsilon^{-3})$ is still achievable if the bias can be properly controlled, and thus we can give an affirmative answer to the question before.

In addition to the biased case, another interesting variant of the original non-convex stochastic optimization is the following non-smooth non-convex optimization:

$$\min_{x\in\mathbb{R}^d} \Phi(x) \coloneqq F(x) + r(x) \coloneqq \mathbb{E}_\xi[f(x;\xi)] + r(x),$$

where $F(x)$ is still smooth and non-convex, but the second part $r(x)$ is convex and non-smooth. This optimization problem is also common in machine learning when adding simple regularization, e.g. $r(x) = \|x\|_1$, to the objective function $F(x)$. For example, the objectives of LASSO [23] and 1-norm SVM [24] can be reformulated as the above form. When combining the general framework with mirror descent [25], we naturally obtain a general framework for solving non-smooth non-convex stochastic optimization, and the sample complexity $\mathcal{O}(\varepsilon^{-3})$ is also achieved.

1.1 Related Work

Variance Reduction Methods SPIDER [7] and SARAH [9] were the first to apply variance reduction to non-convex optimization and obtained the optimal $\mathcal{O}(\varepsilon^{-3})$ sample complexity. The original work for SPIDER required ε-dependent stepsize α and normalized gradient descent. A follow up work proposed an improved version named SpiderBoost [8] where constant stepsize is enough. Setting $\eta = 0$, our general framework exactly recovers the updates of SpiderBoost. SARAH was designed for convex optimization at first but found effective for non-convex case as well. It corresponds to setting $\eta = 0$ and $S = 1$ in the framework. Compared to SPIDER/SpiderBoost, SARAH does not need mini-batches for the computation of recursive gradients, but it has to use a smaller

stepsize to control the error. Both SPIDER/SpiderBoost and SARAH are double-loop algorithms with $Q < T$, and they all need very large batch size D, which might cause some issues for practical applications.

STORM [10] introduced the use of momentum parameter η to the updates of SPIDER and SARAH. The benefit of using momentum parameter is that no mini-batch is required at all, and that single-loop is enough. Setting $D = 1$, $S = 1$, $Q = T$ and using time-varying stepsizes α_t and $\eta_t \in (0, 1)$, our framework recovers STORM. When $\eta \neq 0$ and $S = 1$, the update of the recursive gradients becomes

$$\begin{aligned} h_t &= (1-\eta)(h_{t-1} - \nabla f(x_{t-1}; \xi_t)) + \nabla f(x_t; \xi_t) \\ &= (1-\eta)h_{t-1} + \eta \nabla f(x_t; \xi_t) + (1-\eta)(\nabla f(x_t; \xi_t) - \nabla f(x_{t-1}; \xi_t)). \end{aligned}$$

This is similar to adding momentum term to the update before, and thus we refer to η as the momentum parameter. Another improved method for SPIDER/SARAH is a recently proposed one called PAGE [26]. In each iteration of the algorithm, the recursive gradient is computed with some probability, and the mini-batch or checkpoint gradient is computed otherwise. In this way, PAGE becomes a single-loop algorithm and is easier to implement in practice.

As mentioned above, our general framework includes the widely-used variance reduction methods SPIDER [7, 8], SARAH [9] and STORM [10] as special cases under particular selections of batch sizes, stepsize and momentum parameter. However, in practice, tuning these parameters to satisfy the requirement imposed by these specific methods can be difficult. Our analysis for the general framework suggests that a variety of combinations of parameters provide guarantees for achieving optimal complexity $\mathcal{O}(\varepsilon^{-3})$. It justifies more flexible parameter selection in variance reduction methods. Moreover, we also provide the first extension for these variance reduction methods to the biased case. With properly controlled bias, the optimal complexity is also achievable.

Biased Gradient Methods Recently, there exists a series of work on stochastic gradient descent with biased gradients [27, 28, 29]. These papers assume that the bias comes from some black-box oracles or additive noises with non-zero mean. Here our work focuses on the general variance reduction framework with biased (white-box) oracles. In particular, we are interested in structured stochastic optimization where one can utilize the problem structure to construct biased oracle that adaptively reduces the bias and obtain the optimal complexity. Compared to a few work on specific variance reduction techniques for certain applications [30, 31], our framework is much more general.

Non-Convex Structured Stochastic Optimization For stochastic bilevel optimization, when the lower-level problem is strongly convex and smooth, Ghadimi and Wang [21] achieved an $\mathcal{O}(\varepsilon^{-6})$ complexity using a double-loop algorithm. Hong et al. [17] and Chen et al. [32] considered single-loop algorithm and improved the complexity to $\mathcal{O}(\varepsilon^{-5})$ and $\mathcal{O}(\varepsilon^{-4})$, respectively. For non-convex strongly-concave stochastic minimax optimization, Luo et al. [31] and Xu et al. [33] showed that a double-loop algorithm can achieve an $\mathcal{O}(\varepsilon^{-3})$ complexity. Huang et al. [34] achieved the same complexity with a simplified single-loop algorithm using STORM [10]. For stochastic compositional optimization, Chen et al. [35] demonstrated an oracle complexity of order $\mathcal{O}(\varepsilon^{-4})$ and Zhang and Xiao [36] improved it to $\mathcal{O}(\varepsilon^{-3})$ using nested variance reduction.

Non-Smooth Non-Convex Stochastic Optimization For the non-smooth non-convex stochastic optimization problem we consider, Ghadimi et al. [37] proposed ProxSGD and achieved the $\mathcal{O}(\varepsilon^{-4})$ sample complexity with large batch sizes. Based on SVRG

[5], Li and Li [38] improved the complexity to $\mathcal{O}(\varepsilon^{-10/3})$. Wang et al. [8] also extended SpiderBoost to Prox-SpiderBoost for solving non-smooth stochastic optimization and achieved the sample complexity $\mathcal{O}(\varepsilon^{-3})$ to guarantee an ε-stationary point.

1.2 Organization and Main Results

The thesis is organized as follows:

• In Chapter 2, we give a unified analysis for a rich family of variance reduction methods by the general framework we proposed in the unbiased case. We show that the framework achieves the optimal complexity guarantees under a wide range of parameter settings (batch sizes, stepsize, momentum parameter). It encompasses popular variance reduction algorithms, such as SPIDER [7, 8], SARAH [9] and STORM [10], as special cases and also renders new interesting variants. Our result suggests that: (i) if no momentum term is used ($\eta = 0$), one always need to compute the checkpoint gradient with batch size $\mathcal{O}(\varepsilon^{-2})$ for $\mathcal{O}(\varepsilon^{-1})$ times; (ii) using momentum term helps reduce the checkpoint batch size D, and the recursive batch size S and number of epochs T/Q can be set to 1; (iii) there exists a tradeoff between the stepsize α and the batch size S if no momentum term is used, and a tradeoff between α and D when introducing momentum, which could be useful insights to guide the implementations and parameter tuning in practice.

• In Chapter 3, we first extend the framework to general biased case and prove that when the bias of the gradient estimates satisfies an *average bias growth condition*, the general framework achieves the optimal complexity of $\mathcal{O}(\varepsilon^{-3})$ in finding an ε-stationary point. To the best of our knowledge, this is the first general result for convergence of variance reduction methods in the non-convex and biased setting. This condition sheds lights on the construction of biased stochastic oracle for various structured non-convex stochastic optimization. Motivated by the analysis for the general biased case, we then apply the framework to three non-convex structured stochastic optimization problems including stochastic bilevel optimization, stochastic minimax optimization and stochastic compositional optimization, yielding a family of single-loop near-optimal algorithms. In the context of stochastic bilevel optimization, this closes the gap between the existing best-known upper-bound $\mathcal{O}(\varepsilon^{-4})$ [32] and lower-bound $\mathcal{O}(\varepsilon^{-3})$ for non-convex stochastic optimization [11].

• In Chapter 4, we consider a more general optimization problem where the objective can be non-smooth as well. We combine our general framework with stochastic mirror descent and demonstrate that the sample complexity $\mathcal{O}(\varepsilon^{-3})$ is still achievable, which matches the best-known upper-bound. When using the parameter settings to recover STORM [10], we also obtain the first single-loop algorithm to guarantee convergence without using large batch sizes.

• Finally in Chapter 5, we provide some numerical experiments for the comparison of variance reduction methods on three minimization problems, and give a practical example for how to select all parameters in the general framework to recover different methods. The experiment results verify some theoretical findings in Chapter 2, for example constant stepsize is enough for SPIDER [8] and small stepsize is essential for SARAH [9]. We also find that algorithms with momentum parameter $\eta \neq 0$ are often more stable. The reason might be that these methods are single-loop algorithms and do not require mini-batches.

Notations Let $\|\cdot\|$ denote the L_2 norm $\|\cdot\|_2$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is ℓ-Lipschitz continuous if $|f(x) - f(y)| \leq \ell\|x - y\|$ holds for any $x, y \in \mathbb{R}^d$. A function f is L-smooth

on $\mathbb{R}^d$, if it is continuously differentiable on $\mathbb{R}^d$ and it holds that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for any $x, y \in \mathbb{R}^d$. We use $\nabla G(x, y)$ to denote the full gradient of function G and $\nabla_x G(x, y)$ to denote partial gradient of G on x.

Chapter 2

General Analysis of Variance Reduction for Non-Convex Optimization

Overview In this chapter, we propose a general algorithmic framework for variance reduction methods to solve smooth non-convex stochastic optimization with unbiased gradient oracle. The problem setting and assumptions are given in Section 2.1, and the general framework can be found in Section 2.2. In Section 2.3, we provide a unified convergence analysis of the general framework and show its optimal $\mathcal{O}(\varepsilon^{-3})$ oracle complexity. Finally in Section 2.4 we discuss the parameter choices and demonstrate that our general framework recovers popular variance reduction methods including SpiderBoost [8], SARAH [9] and STORM [10]. We also show that the general framework allows other selections of parameters and helps find undiscovered interesting variants of variance reduction methods.

2.1 Problem Setting and Assumptions

We first formally state the problem setting and the assumptions. The problem we consider is the following smooth non-convex stochastic optimization:

$$\min_{x\in\mathbb{R}^d} F(x) := \mathbb{E}_\xi[f(x;\xi)], \tag{2.1}$$

where $F(x):\mathbb{R}^d\to\mathbb{R}$ is a continuously differentiable function satisfying the assumptions below.

Assumption 2.1 (Objective Function) *The possibly non-convex objective function $F(x)$ satisfies that*

- *$F(x)$ is L_F-smooth with parameter $L_F>0$. This means for any $x,y\in\mathbb{R}^d$, it holds that*

$$F(y)\le F(x)+\nabla F(x)^\top(y-x)+\frac{L_F}{2}\|x-y\|^2.$$

- *The optimal value $F^*:=\min_{x\in\mathbb{R}^d}F(x)$ is finite.*

We are interested in the stochastic setting where one only has access to information about $F(x)$ via a first-order stochastic oracle $\mathcal{SO}$. For a query point x, the oracle returns a pair $(f(x;\xi),\nabla f(x;\xi))$ for some well-defined random vector ξ with support $\Omega\subseteq\mathbb{R}^d$ and distribution $\mathbb{P}$. In other words, we can only compute noisy estimates of the function $F(x)$ and its gradient $\nabla F(x)$ with ξ representing all the randomness. The goal is to minimize $F(x)=\mathbb{E}_\xi[f(x;\xi)]$ through access to the stochastic gradient $\nabla f(x;\xi)$. The

modelling above covers most scenarios in machine learning applications, e.g. the finite-sum settings $F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ in supervised learning and the pure stochastic settings in reinforcement learning. In this chapter, we first study the case when the gradient estimate returned by the oracle is unbiased:

$$\mathbb{E}_\xi[\nabla f(x;\xi)] = \nabla F(x), \quad \forall x \in \mathbb{R}^d.$$

In addition to unbiasedness, we make the following assumptions about the stochastic gradient $\nabla f(x;\xi)$.

Assumption 2.2 (Gradient Estimate) *The unbiased gradient estimate $\nabla f(x;\xi)$ returned by the stochastic oracle $\mathcal{SO}$ satisfies that*

$$\mathbb{E}_\xi\|\nabla f(x;\xi) - \nabla F(x)\|^2 \le \sigma^2, \tag{2.2}$$

$$\mathbb{E}_\xi\|\nabla f(x_1;\xi) - \nabla f(x_2;\xi)\| \le \ell_f\|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathbb{R}^d, \tag{2.3}$$

for some constants $\sigma > 0$ and $\ell_f > 0$.

Before discussing the above assumption, we first define the complexity measures. For general non-convex optimization, it is intractable to find the global minimum [25]. As an alternative, we define the notion of ε-stationarity, which is a commmon surrogate for non-convex objectives.

Definition 2.3 *For some accuracy measure $\varepsilon > 0$, a point x is called an ε-stationary point for a differentiable non-convex function $F(x)$ if $\|\nabla F(x)\| \le \varepsilon$.*

For smooth non-convex stochastic optimization with an unbiased gradient oracle satisfying bounded variance condition (2.2), Arjevani et al. [11] proved that any algorithm requires at least $\mathcal{O}(\varepsilon^{-4})$ queries to find an ε-stationary point. The lower bound is tight since it can be achieved by stochastic gradient descent (SGD).

In addition, if the oracle also satisfies the average smoothness property (2.3), Arjevani et al. [11] showed a lower bound of $\mathcal{O}(\varepsilon^{-3})$ queries. The bound is also tight and achieved by various variance reduction methods such as SPIDER [7], SpiderBoost [8], SARAH [9] and STORM [10]. We will see later that the average smoothness condition is crucial for these variance reduction techniques to achieve the $\mathcal{O}(\varepsilon^{-3})$ complexity.

We notice that all these variance reduction methods are very much similar not only in the algorithms, but also in their convergence analyses. That's why we want to propose a general framework and provide a simple and unified analysis.

2.2 General Algorithm

The general algorithmic framework we propose for solving (2.1) is given in Algorithm 1. In Algorithm 1, T is the total iteration number, Q denotes the epoch length, D and S are batch sizes, α is the fixed stepsize and η is the momentum parameter used to compute the recursive gradient. To be specific, let every Q iterations denote an epoch. At the beginning of each epoch, one queries the oracle $\mathcal{SO}$ for D times and computes

$$h_t = \frac{1}{D}\sum_{i=1}^{D} \nabla f(x_t;\xi_t^i).$$

We will refer to this as the checkpoint gradient. Note that the queries to the oracle are independent, i.e. the random variables $\{\xi_t^i\}_{i=1}^D$ are sampled independently from the same

Algorithm 1 General Framework for Variance Reduction Methods

Input: $T, Q, D, S, x_0, \alpha, \eta$.
  for $t = 0, 1, \cdots, T-1$ **do**
    if $t \equiv 0 \pmod{Q}$ **then**
      Query $\mathcal{SO}$ for D times and compute $h_t = \frac{1}{D}\sum_{i=1}^{D} \nabla f(x_t; \xi_t^i)$.
      $x_{t+1} = x_t - \alpha h_t$.
    else
      Query the oracle $\mathcal{SO}$ for $2S$ times.
      $h_t = (1-\eta)\Big(h_{t-1} - \frac{1}{S}\sum_{i=1}^{S} \nabla f(x_{t-1}; \xi_t^i)\Big) + \frac{1}{S}\sum_{i=1}^{S} \nabla f(x_t; \xi_t^i)$.
      $x_{t+1} = x_t - \alpha h_t$.
    end if
  end for
Output: x_τ with τ chosen uniformly at random from $\{0, 1, \cdots, T-1\}$.

distribution $\mathbb{P}$. In other iterations, one queries the oracle $\mathcal{SO}$ for S times at both points x_{t-1} and x_t and computes the recursive gradient

$$h_t = (1-\eta)\left(h_{t-1} - \frac{1}{S}\sum_{i=1}^{S} \nabla f(x_{t-1}; \xi_t^i)\right) + \frac{1}{S}\sum_{i=1}^{S} \nabla f(x_t; \xi_t^i). \tag{2.4}$$

Here S is typically much smaller than D. After computing the gradient estimator h_t of the true gradient $\nabla F(x_t)$, the update for x_{t+1} is just one gradient descent step using the estimator h_t with stepsize α.

When there is no momentum, i.e. $\eta = 1$, h_t reduces to a mini-batch estimator of $\nabla F(x_t)$ and loses the variance recursion property. The framework thus reduces to mini-batch SGD. When $\eta < 1$, h_t has a similar form as the classical variance reduction technique SPIDER and SARAH (when $\eta = 0$), and STORM (when using time-varying $\eta_t \in (0, 1)$).

The total number of calls to the oracle $\mathcal{SO}$ is $\mathcal{O}(T(2S + D/Q))$, as shown in the lemma below.

Lemma 2.4 *The oracle complexity of Algorithm 1 is $\mathcal{O}\Big(T\Big(2S + \frac{D}{Q}\Big)\Big)$, where T stands for the total number of iterations, Q is the epoch length, D is the batch size of the checkpoint gradients and S is the batch size to compute recursive gradients.*

Proof In iteration $t = 0, Q, 2Q, \cdots, \Big\lfloor \frac{T}{Q} \Big\rfloor Q$, we use a batch with size D, thus $\Big(\Big\lfloor \frac{T}{Q} \Big\rfloor + 1\Big)D$ calls of oracle in total. In other iterations, we use a batch with size S and use $2S$ calls per-iteration, leading to $\Big(T - \Big(\Big\lfloor \frac{T}{Q} \Big\rfloor + 1\Big)\Big) \cdot 2S$ calls in total. Adding these two, the oracle complexity of Algorithm 1 is

$$\begin{aligned}
\Big(\Big\lfloor \frac{T}{Q} \Big\rfloor + 1\Big)D + \Big(T - \Big(\Big\lfloor \frac{T}{Q} \Big\rfloor + 1\Big)\Big) \cdot 2S &= 2TS + \Big(\Big\lfloor \frac{T}{Q} \Big\rfloor + 1\Big)(D - 2S) \\
&< 2TS + \Big(\frac{T}{Q} + 1\Big)D \\
&= \mathcal{O}\Big(2TS + T\frac{D}{Q}\Big).
\end{aligned}$$

We use the fact that $\Big\lfloor \frac{T}{Q} \Big\rfloor \le \frac{T}{Q}$, $S \ge 1$ and $Q \le T$ in general. □

Before we move to the convergence analysis of Algorithm 1, we restate a well-known result in Lemma 2.5 for recursive gradients in the variance reduction related literature

and demonstrate the proof in Appendix A.1 for completeness. We define

$$A_t := \mathbb{E}\|h_t - \nabla F(x_t)\|^2$$

be the estimation error of the gradient estimator h_t with the expectation taking over all the randomness up to iteration t. By decomposing the recursive update (2.4) and using the average smoothness assumption (2.3), we have the following results.

Lemma 2.5 *For Algorithm 1, under Assumptions 2.2, supposing that* $0 \leq \eta < 1$, *we have*

$$A_{t+1} \leq (1-\eta)A_t + 2\sigma^2\frac{\eta^2}{S} + 2\ell_f^2\frac{\alpha^2}{S}\mathbb{E}\|h_t\|^2,$$

for $t+1 \not\equiv 0 \pmod{Q}$, *i.e. in iterations where recursive gradients are used.*

Lemma 2.5 suggests that the estimation error at iteration $t+1$ can be traced back to iteration t up to some additional error terms. This is different from what we have in SGD. The equivalent term in SGD is $A_t = \mathbb{E}\|\nabla f(x_t;\xi_t) - \nabla F(x_t)\| \leq \sigma^2$ by (2.2), which is independent of other iterations.

Lemma 2.5 is the key to proving optimal $\mathcal{O}(\varepsilon^{-3})$ complexity for different variance reduction methods. We will give more details in the next section.

2.3 Convergence Analysis

We provide a simple and unified convergence analysis for the general framework in Algorithm 1 with different parameter setups. The setups we find not only recover popular variance reduction methods such as SPIDER and SARAH, but also help discover novel variants.

First of all, since $F(x)$ is L_F-smooth, the gradient update rule gives us

$$\begin{aligned} F(x_{t+1}) &\leq F(x_t) - \alpha\nabla F(x_t)^\top h_t + \frac{L_F}{2}\alpha^2\|h_t\|^2 \\ &\leq F(x_t) - \frac{\alpha}{2}\|\nabla F(x_t)\|^2 - \frac{\alpha}{3}\|h_t\|^2 + \frac{\alpha}{2}\|h_t - \nabla F(x_t)\|^2, \end{aligned} \tag{2.5}$$

where the last inequality uses $2a^\top b = \|a\|^2 + \|b\|^2 - \|a-b\|^2$ and assumes $\alpha \leq \frac{1}{3L_F}$.

Dividing $\alpha/2$ on both sides of (2.5) and summing up from $t=0$ to $T-1$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(x_t)\|^2 \leq \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 + \frac{1}{T}\sum_{t=0}^{T-1}A_t, \tag{2.6}$$

where we uses the fact that $F(x_T) \geq F^* = \min_x F(x)$ and $A_t = \mathbb{E}\|h_t - \nabla F(x_t)\|^2$. Equation (2.6) shows that the average squared gradient norm can be upper bounded with the average estimation error $\frac{1}{T}\sum_{t=0}^{T-1}A_t$ incurred by recursive gradient estimator. We then show that the average estimation error can be bounded by carefully selecting batch sizes D, S, stepsize α, and other parameters η and Q in Algorithm 1 with the help of Lemma 2.5.

We first show that the average estimation error satisfies the following bound, which is a direct consequence of Lemma 2.5 by distinguishing two cases for $\eta = 0$ and $\eta \in (0,1)$. The detailed proof can be found in Appendix A.1.

Lemma 2.6 *Under Assumptions 2.2, for a constant $c_\eta \in (0, 1/\alpha^2)$ we can choose, we have that*

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \le \begin{cases} \frac{\sigma^2}{D} + 2\ell_f^2 \frac{Q\alpha^2}{S} \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2, & \text{if } \eta = 0; \\ \frac{\sigma^2}{c_\eta \alpha D}\frac{1}{\alpha T} + \frac{2c_\eta\sigma^2}{S}\frac{1}{\alpha T}\sum_{t=0}^{T-1}\alpha^3 + \frac{2\ell_f^2}{c_\eta S}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2, & \text{if } \eta = c_\eta\alpha^2. \end{cases}$$

Note that the above bounds hold for any choice of D, S, Q, T, stepsize α and momentum parameter η. The following main result shows that for specific choices of these parameters, the average estimation error will not grow faster than the average square norm of the gradient estimator h_t, up to an additive diminishing term.

Theorem 2.7 *Suppose that Assumption 2.2 holds, the average estimator error satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \le \frac{\sigma^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2, \tag{2.7}$$

for some constant $\rho_A > 0$ we can select, under either of the following setups.

(i) $\eta = 0$, for any α such that $(\ell_f^{-2}\rho_A)^{1/3} \cdot T^{-1/3} \le \alpha \le 1/(3L_F)$, and

$$D = \alpha T, \quad S = \frac{\ell_f}{\sqrt{\rho_A}}\alpha^{3/2}T^{1/2}, \quad Q = \frac{\sqrt{\rho_A}}{2\ell_f}\alpha^{-1/2}T^{1/2},$$

with oracle complexity $4\rho_A^{-1/2}\ell_f \cdot \alpha^{3/2}T^{3/2} = \mathcal{O}(\alpha^{3/2}T^{3/2})$.

(ii) $\eta = 2\ell_f^2\alpha^2/\rho_A$, for $\alpha = \rho_A^{1/3}(8\ell_f^2 \cdot T)^{-1/3} \le 1/(3L_F)$ if T large enough, and

$$D = \rho_A \ell_f^{-2}\alpha^{-1}, \quad S = 1, \quad Q = T,$$

with oracle complexity $\mathcal{O}(T)$.

Proof When $\eta = 0$, by Lemma 2.6,

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \le \frac{\sigma^2}{D} + 2\ell_f^2\frac{Q\alpha^2}{S}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 = \frac{\sigma^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2,$$

with the choice that

$$D = \alpha T, \quad 2\ell_f^2\frac{Q\alpha^2}{S} = \rho_A. \tag{2.8}$$

By Lemma 2.4, the number of total oracle calls required is

$$T(2S + \frac{D}{Q}) \ge T \cdot 2\sqrt{2D \cdot \frac{S}{Q}} = T \cdot 2\sqrt{2\alpha T \cdot \frac{2\ell_f^2\alpha^2}{\rho_A}} = \frac{4\ell_f}{\sqrt{\rho_A}}\alpha^{3/2}T^{3/2},$$

where the first inequality holds since $a + b \ge 2\sqrt{ab}$ for any $a, b > 0$, and the first equality holds by (2.8). The minimum oracle complexity is thus $4\rho_A^{-1/2}\ell_f \cdot \alpha^{3/2}T^{3/2} = \mathcal{O}(\alpha^{3/2}T^{3/2})$, and this complexity is achieved by setting $2S = D/Q$. Solving for $2SQ = D = \alpha T$ and $2\ell_f^2 S^{-1}Q = \rho_A\alpha^{-2}$, we obtain that

$$Q = \frac{\sqrt{\rho_A}}{2\ell_f}\alpha^{-1/2}T^{1/2}, \quad S = \frac{\ell_f}{\sqrt{\rho_A}}\alpha^{3/2}T^{1/2}.$$

The requirement that $\alpha \geq (\ell_f^{-2}\rho_A)^{1/3} \cdot T^{-1/3}$ comes from the fact that $S \geq 1$. The requirement that $\alpha \leq 1/(3L_F)$ makes sure that (2.5) at the beginning of Section 2.3 holds.

When $\eta \neq 0$, by Lemma 2.6, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1} A_t &\leq \frac{1}{c_\eta \alpha D}\cdot\frac{\sigma^2}{\alpha T} + \frac{2c_\eta}{S}\sum_{t=0}^{T-1}\alpha^3\cdot\frac{\sigma^2}{\alpha T} + \frac{2\ell_f^2}{c_\eta S}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 \\ &= \frac{\sigma^2}{\alpha T} + \rho_A\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2,\end{aligned}$$

if the following condition holds:

$$\alpha D = \frac{2}{c_\eta}, \quad \sum_{t=0}^{T-1}\alpha^3 = \frac{S}{4c_\eta}, \quad c_\eta S = \frac{2\ell_f^2}{\rho_A}.$$

The requirements now have no explicit dependence on Q and thus we can select $Q = T$ to avoid multiple computations of the checkpoint gradients and obtain a single-loop algorithm. In addition, the requirements also have better dependence on the batch sizes S and D. Selecting $c_\eta = 2\ell_f^2\rho_A^{-1}$, we can set $S = 1$, i.e. no mini-batch is required apart from one checkpoint gradient at the beginning. Therefore, we immediately obtain that $\alpha = 1/(4c_\eta T)^{-1/3} = \rho_A^{1/3}(8\ell_f^2\cdot T)^{-1/3}$ and $D = 2/(c_\eta\alpha) = \rho_A\ell_f^{-2}\alpha^{-1} = \mathcal{O}(T^{1/3})$. When $S = 1$ and $Q = T$, the oracle complexity is just $\mathcal{O}(T + D) = \mathcal{O}(T)$. □

Based on the above results and equation (2.6), we immediately have the following theorem.

Theorem 2.8 *Suppose Assumption 2.1 and 2.2 hold. Let the choice of parameters be specified as in Theorem 2.7 with $\rho_A \in (0, \frac{2}{3}]$. Then the output of the general framework in Algorithm 1 satisfies that*

$$\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \leq \frac{2[F(x_0) - F^*] + \sigma^2}{\alpha T},$$

where x_τ is a random output uniformly chosen from the iterates $\{x_t\}_{t=0}^{T-1}$. This further implies that to achieve an ε-stationary point, the total oracle complexity is $\mathcal{O}(\varepsilon^{-3})$, as shown in Remark 2.9 below.

Proof By equation (2.6) and (2.7), we immediately obtain that

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(x_t)\|^2 &\leq \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 + \frac{1}{T}\sum_{t=0}^{T-1}A_t \\ &\leq \frac{2[F(x_0) - F^*] + \sigma^2}{\alpha T} - \left(\frac{2}{3} - \rho_A\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 \\ &\leq \frac{2[F(x_0) - F^*] + \sigma^2}{\alpha T},\end{aligned}$$

since $\rho_A \leq 2/3$. By Assumption 2.1, F^* is finite. If we select x_0 such that $F(x_0)$ is also finite, we have that $\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \leq \mathcal{O}(1/(\alpha T))$. To guarantee ε-stationarity, by Definition 2.3, we need $\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \leq \varepsilon^2$. Setting $\alpha T = \mathcal{O}(\varepsilon^{-2})$ would satisfy the requirement and lead to the $\mathcal{O}(\varepsilon^{-3})$ complexity as shown in Remark 2.9. □

Remark 2.9 (Oracle Complexity) *To guarantee ε-stationarity, we need $\alpha T = \mathcal{O}(\varepsilon^{-2})$. As a result, the oracle complexity under parameter setup (i) is $\mathcal{O}(\alpha^{3/2}T^{3/2}) = \mathcal{O}(\varepsilon^{-3})$ and setup (ii) is $\mathcal{O}(T) = \mathcal{O}(\varepsilon^{-3})$ since $\alpha = \mathcal{O}(T^{-1/3})$. Both conditions yield the oracle complexity of $\mathcal{O}(\varepsilon^{-3})$,*

Table 2.1: Summary of parameter selections for Algorithm 1 for finding an ε-stationary point. T stands for iteration complexity, T/Q for number of epoches, D for batch size at checkpoints, S for batch size at other iterations, η for the momentum parameter and α for the stepsize. Note that SPIDER here refers to the follow-up and improved version SpiderBoost.

Parameters	SPIDER	SARAH	STORM	New 1	New 2
T	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-5/2})$	$\mathcal{O}(\varepsilon^{-3})$
T/Q	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-1})$	1	$\mathcal{O}(\varepsilon^{-1})$	1
D	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(1)$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1})$
S	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\varepsilon^{-1/2})$	$\mathcal{O}(1)$
η (or η_t)	0	0	$\mathcal{O}(t^{-2/3})$	0	$\mathcal{O}(\varepsilon^2)$
α (or α_t)	$\mathcal{O}(1)$	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(t^{-1/3})$	$\mathcal{O}(\varepsilon^{1/2})$	$\mathcal{O}(\varepsilon)$
Complexity	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-3})$	$\tilde{\mathcal{O}}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-3})$

which is known to be the optimal oracle complexity [11]. In the proof of Theorem 2.7, we also show that $\mathcal{O}(\alpha^{3/2}T^{3/2})$ is the minimum complexity under setup (i) and $\mathcal{O}(T)$ is also the minimum complexity the general framework allows under setup (ii) since $S \geq 1$ and $Q \leq T$. This suggests the best complexity the framework can achieve is $\mathcal{O}(\varepsilon^{-3})$, also matching the lower bound.

2.4 Parameter Choices and New Variants

Combining Theorem 2.7 and 2.8, we then discuss the different settings of parameters which help achieve the optimal rate and explain how our framework recovers popular variance reduction methods. We also find new variations of variance reduction methods as byproducts of the general analysis.

Under Parameter Setup (i) When $\eta = 0$ and we set $\alpha T = \mathcal{O}(\varepsilon^{-2})$ to guarantee an ε-stationary point, by the parameter selection rule in Theorem 2.7, we obtain that $D = \alpha T = \mathcal{O}(\varepsilon^{-2})$, $T/Q = \mathcal{O}(\alpha^{1/2}T^{1/2}) = \mathcal{O}(\varepsilon^{-1})$ and $S/\alpha = \mathcal{O}(\alpha T) = \mathcal{O}(\varepsilon^{-1})$. This means that one has to use a large checkpoint batch size $D = \mathcal{O}(\varepsilon^{-2})$ and compute the checkpoint gradients for $T/Q = \mathcal{O}(\varepsilon^{-1})$ times, and there exhibits a tradeoff between the per-iteration batch size S and stepsize α. When using small batch size S, a small stepsize α is also required to control the error.

To give specific choice of all parameters in the order of ε, we let $\alpha = T^{-1/k} \geq T^{-1/3}$ for $k \geq 3$ without loss of generality. Then we have $T = \mathcal{O}(\varepsilon^{-2k/(k-1)})$ and $\alpha = \mathcal{O}(\varepsilon^{2/(k-1)})$ since $\alpha T = \mathcal{O}(\varepsilon^{-2})$. If we choose $k = \infty$ such that $\alpha = \mathcal{O}(1)$, $T = \mathcal{O}(\varepsilon^{-2})$ and $S = \mathcal{O}(\varepsilon^{-1})$, this reduces to SpiderBoost [8] where a constant level of stepsize is allowed. If we choose $k = 3$ such that $\alpha = \mathcal{O}(\varepsilon)$, $T = \mathcal{O}(\varepsilon^{-3})$ and $S = \mathcal{O}(1)$, this reduces to SARAH [9] where we can set $S = 1$ to avoid the use of mini-batch for recursive gradients. If we choose other $k > 3$, this leads to a set of new variations which have not been discovered before, providing more flexibility in the selection of parameters. We summarize the different choices of parameters for each methods recovered by our general framework in Table 2.1, as well as a new variation New 1 with the choice that $k = 5$ as an example.

Under Parameter Setup (ii) When $\eta > 0$ and we set $T = \mathcal{O}(\varepsilon^{-3})$ to guarantee an ε-stationary point, by the parameter selection rule in Theorem 2.7, we obtain that $\alpha = \mathcal{O}(T^{-1/3}) = \mathcal{O}(\varepsilon)$ and $D = \mathcal{O}(\alpha^{-1}) = \mathcal{O}(\varepsilon^{-1})$. Introducing the momentum largely reduces the batch sizes as $S = 1$ and D is smaller than before. Furthermore, setting $Q = T$ avoids computation of multiple checkpoint gradients, i.e. we only compute it at iteration $t = 0$. This new single-loop algorithm is called New 2 in Table 2.1.

Note that when one uses time-varying stepsizes α_t and corresponding η_t, the framework also recovers STORM [10] by setting stepsizes $\alpha_t = \frac{k}{(\omega+t)^{1/3}} = \mathcal{O}(t^{-1/3})$ and momentum parameters $\eta_t = c_\eta \alpha_t^2$ for some constants c_η, k and ω that one can choose. The benefit of time-varying stepsizes is that we can choose $D = 1$, which means no mini-batch is needed at all. However, it will introduce additional logarithmic terms in the oracle complexity, leading to only near-optimal rate. We then give a more detailed analysis below.

When using time-varying stepsizes $\{\alpha_t\}_{t=0}^T$ for the update $x_{t+1} = x_t - \alpha_{t+1} h_t$ and $\eta_t = c_\eta \alpha_t^2$ for the computation of recursive gradient h_t, Lemma 2.5 holds by replacing α and η by α_{t+1} and η_{t+1}. With a slight modification of the proof of Lemma 2.6 for $\eta > 0$, we first show that

$$\frac{\eta_{t+1}}{\alpha_{t+1}} A_t + \frac{A_t}{\alpha_t} - \frac{A_t}{\alpha_{t+1}} \leq \frac{A_t}{\alpha_t} - \frac{A_{t+1}}{\alpha_{t+1}} + \frac{2c_\eta^2 \sigma^2}{S} \alpha_{t+1}^3 + \frac{2\ell_f^2}{S} \alpha_{t+1} \mathbb{E}\|h_t\|^2, \tag{2.9}$$

by rearranging terms of the modified Lemma 2.5 and dividing α_{t+1} on both sides. The selection of $\alpha_t = \frac{k}{(\omega+t)^{1/3}}$ guarantees that $\alpha_0 \geq \alpha_t \geq \alpha_{t+1} \geq \alpha_T$ and

$$\frac{\eta_{t+1}}{\alpha_{t+1}} + \frac{1}{\alpha_t} - \frac{1}{\alpha_{t+1}} = c_\eta \alpha_{t+1} + \frac{1}{\alpha_t} - \frac{1}{\alpha_{t+1}} \geq (c_\eta - c_\alpha)\alpha_{t+1},$$

for some constant c_α depending on k and ω [10, Proof of Theorem 1]. Taking summation of (2.9) and dividing both sides by $(c_\eta - c_\alpha)\alpha_T T$, we obtain

$$\frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} A_t \leq \frac{\sigma^2}{(c_\eta - c_\alpha)\alpha_0 D} \frac{1}{\alpha_T T} + \frac{2c_\eta^2 \sigma^2}{(c_\eta - c_\alpha) S} \frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1}^3 + \frac{2\ell_f^2}{(c_\eta - c_\alpha) S} \frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} \mathbb{E}\|h_t\|^2.$$

We choose $c_\eta = c_\alpha + 2\ell_f^2 \rho_A^{-1}$ such that $S = 1$. Furthermore, we can set $D = 1$ since now $\alpha_0 = \mathcal{O}(1)$. However, we have that $\sum_{t=0}^{T-1} \alpha_{t+1}^3 = \mathcal{O}\left(\sum_{t=0}^{T-1} \frac{1}{t}\right) = \mathcal{O}(\log T)$, and thus

$$\frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} A_t \leq \mathcal{O}\left(\frac{1}{\alpha_T T}\right) + \mathcal{O}\left(\frac{\log T}{\alpha_T T}\right) + \rho_A \cdot \frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} \mathbb{E}\|h_t\|^2.$$

Combined with (2.5) by first taking summations and then dividing both sides by $\alpha_T T/2$, we can show that

$$\begin{aligned}
\mathbb{E}_\tau \|\nabla F(x_\tau)\|^2 &\leq \frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} \mathbb{E}\|\nabla F(x_t)\|^2 \\
&\leq \frac{2[F(x_0) - F^*]}{\alpha_T T} - \frac{2}{3\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} \mathbb{E}\|h_t\|^2 + \frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} A_t \\
&\leq \mathcal{O}\left(\frac{\log T}{\alpha_T T}\right) - \left(\frac{2}{3} - \rho_A\right) \frac{1}{\alpha_T T} \sum_{t=0}^{T-1} \alpha_{t+1} \mathbb{E}\|h_t\|^2 \\
&\leq \mathcal{O}\left(\frac{\log T}{\alpha_T T}\right) = \mathcal{O}\left(\frac{\log T}{T^{2/3}}\right),
\end{aligned}$$

since $\alpha_{t+1} \geq \alpha_T = \mathcal{O}(T^{-1/3})$ and we choose $\rho_A \leq 2/3$. The oracle complexity is thus $\mathcal{O}(T) = \tilde{\mathcal{O}}(\varepsilon^{-3})$ with $\tilde{\mathcal{O}}$ hiding additional logarithmic terms in ε^{-1}. Therefore, STORM avoids mini-batch by time-varying stepsizes. However, the complexity is only near-optimal. The parameter choice of STORM is also included in Table 2.1.

Concluding Remarks In this chapter, we propose a general framework in Algorithm 1 for different variance reduction methods to solve smooth non-convex stochastic optimization (2.1) with unbiased oracle satisfying Assumption 2.2. We provide a unified analysis of the convergence rate of Algorithm 1 and show its optimal $\mathcal{O}(\varepsilon^{-3})$ oracle complexity. With the proper parameter selection rules as given in Section 2.4, our general framework encompasses popular variance reduction algorithms such as SPIDER [7, 8], SARAH [9] and STORM [10] as special cases, and also renders other new interesting variants as summarized in Table 2.1. In the next chapter, we will apply the general framework to smooth non-convex stochastic optimization with biased gradients and show that we can still achieve the optimal $\mathcal{O}(\varepsilon^{-3})$ oracle complexity as long as the bias can be properly controlled.

Chapter 3

Variance Reduction for Biased Stochastic Optimization

Overview In this chapter, we apply the general framework proposed in Section 2.2 of the previous chapter to smooth non-convex stochastic optimization with biased gradient estimates. In Section 3.1, we show that our general framework can still achieve $\mathcal{O}(\varepsilon^{-3})$ oracle complexity if the bias satisfies a novel growth condition. In the next sections, we apply the general framework to three well-known structured stochastic optimization problems including stochastic bilevel optimization (Section 3.2), stochastic minimax optimization (Section 3.3) and stochastic compositional optimization (Section 3.4). In all three applications, we can only construct biased gradient oracles. However, we prove that the bias can be properly controlled in a similar way as the growth condition in Section 3.1. Therefore, using the general framework to estimate different quantities of interest, we can still achieve the optimal $\mathcal{O}(\varepsilon^{-3})$ convergence rate, which also demonstrates the effectiveness of our general analysis in the previous chapter.

3.1 General Framework with Biased Gradient Estimates

The problem we consider in this chapter is still the smooth non-convex stochastic optimization defined in (2.1) with the objective function $F(x)$ satisfying Assumption 2.1, i.e. $F(x)$ is L_F-smooth and the optimal value F^* is finite. However, in this chapter we study the case where we only have access to some (possibly white-box) biased gradient oracle. Such problems where we can only construct biased gradient estimator have appeared in many modern machine learning applications, and have received more and more attentions recently. These applications can often be formulated as structured or nested optimization, and we will discuss more about it in the following sections. In this section, we first give a general analysis of the biased stochastic optimization problems.

For a query point x, the biased stochastic oracle $\mathcal{SO}$ returns some biased gradient estimate of the true gradient $\nabla F(x)$. We define the biased gradient estimate as $\bar{\nabla} f(x;\xi)$ to distinguish it from the unbiased case and denote the bias as

$$B(x) := \mathbb{E}_\xi[\bar{\nabla} f(x;\xi)] - \nabla F(x),$$

where $B : \mathbb{R}^d \to \mathbb{R}$ measures the bias at point x. Similar to Assumption 2.2, we also assume the biased gradient estimate satisfies the bounded variance condition (3.1) and the average smoothness condition (3.2) as follows.

Assumption 3.1 (Biased Gradient Estimate) *The biased gradient estimate $\bar{\nabla} f(x;\xi)$ returned by the stochastic oracle $\mathcal{SO}$ satisfies that*

$$\mathbb{E}_\xi \|\bar{\nabla} f(x;\xi) - \mathbb{E}_\xi[\bar{\nabla} f(x;\xi)]\|^2 \le \sigma^2, \tag{3.1}$$

$$\mathbb{E}_\xi \|\bar{\nabla} f(x_1;\xi) - \bar{\nabla} f(x_2;\xi)\| \le \ell_f \|x_1 - x_2\|, \forall x_1, x_2 \in \mathbb{R}^d, \tag{3.2}$$

for some constants $\sigma > 0$ and $\ell_f > 0$.

Everything is the same as Assumption 2.2 except that currently we only have the biased gradient estimate $\bar{\nabla} f(x;\xi)$. The average smoothness assumption (3.2) allows the use of variance reduction methods to achieve $\mathcal{O}(\varepsilon^{-3})$ oracle complexity.

Note that this stochastic oracle might be given by some black-box or possibly white-box that allows us to utilize problem specific structure to construct the estimate $\bar{\nabla} f(x;\xi)$. We shall elaborate this in various applications in the later sections.

Replacing $\nabla f(x;\xi)$ in Algorithm 1 by $\bar{\nabla} f(x;\xi)$, we obtain the general framework for solving biased optimization using variance reduction. We use the recursive gradient estimator h_t constructed via access to the biased oracle to estimate the true gradient $\nabla F(x_t)$, and update $x_{t+1} = x_t - \alpha h_t$ iteratively for some stepsize $\alpha > 0$. The gradient error term $\mathbb{E}\|h_t - \nabla F(x_t)\|^2$ can be decomposed as:

$$\underbrace{\mathbb{E}\|h_t - \nabla F(x_t)\|^2}_{\text{(gradient error)}} \le 2\underbrace{\mathbb{E}\|h_t - \mathbb{E}_{\xi_t}[\bar{\nabla} f(x_t;\xi_t)]\|^2}_{:=A_t\text{(estimation error)}} + 2\underbrace{\mathbb{E}\|B(x_t)\|^2}_{:=B_t\text{(bias)}},$$

by the inequality $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$. Compared to the unbiased case, in addition to the estimation error A_t, we also have a bias term B_t. Thus by (2.5) and equivalent to (2.6), under the assumption that $F(x)$ is L_F-smooth, we have that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla F(x_t)\|^2 \le \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2 + \frac{2}{T}\sum_{t=0}^{T-1} A_t + \frac{2}{T}\sum_{t=0}^{T-1} B_t. \tag{3.3}$$

The above equation shows that the average squared gradient norm in this case can be upper bounded with the average estimation error $\frac{1}{T}\sum_{t=0}^{T-1} A_t$ incurred by the recursive gradient estimator and the average bias $\frac{1}{T}\sum_{t=0}^{T-1} B_t$ incurred by the biased stochastic oracle. Note that the bias depends solely on the biased oracle, whereas the average estimation error is mainly determined by the parameter choices of our general framework.

The estimation error A_t does not depend on the bias. Thus simply replacing $\nabla f(x;\xi)$ by $\bar{\nabla} f(x;\xi)$ and $F(x_t)$ by $\mathbb{E}_\xi[\bar{\nabla} f(x;\xi)]$, it is easy to see that Lemma 2.5, 2.6 and then Theorem 2.7 holds for

$$A_t := \mathbb{E}\|h_t - \mathbb{E}_{\xi_t}[\bar{\nabla} f(x_t;\xi_t)]\|^2$$

in the biased case if Assumption 3.1 holds. Under the parameter choices as specified in Theorem 2.7, we can control the average estimation error term by (2.7), and thus by (3.3), we obtain that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla F(x_t)\|^2 \le \frac{2[F(x_0) - F^*] + 2\sigma^2}{\alpha T} - \left(\frac{2}{3} - 2\rho_A\right)\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2 + \frac{2}{T}\sum_{t=0}^{T-1} B_t, \tag{3.4}$$

for a constant $\rho_A > 0$ we can choose. Equation (3.4) suggests that if the bias term can be properly controlled, we can obtain the same complexity result as the unbiased case. Next we propose a novel growth condition to control the bias.

Average Bias Growth Condition As discussed earlier, we are interested in cases when the bias can be properly controlled. We first introduce a notion of *strong bias growth condition*, namely, for any input $x \in \mathbb{R}^d$,

$$\|B(x)\|^2 \leq \rho_1 \|\nabla F(x)\|^2, \tag{SBG}$$

for some $\rho_1 \in (0, \frac{1}{2})$, which says that the bias shrinks relative to the true gradient. This is analogous to the strong growth condition on the stochastic gradients widely studied for over-parametrized models [39, 40]. Indeed, this kind of condition can be very strong and expensive to obtain. Instead, we will consider a relaxed condition, which we call *average bias growth condition*:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|B(x_t)\|^2 \leq \frac{\rho_0}{\alpha T} + \rho_1 \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla F(x_t)\|^2, \tag{ABG}$$

for some $\rho_0 \geq 0$ and $\rho_1 \in [0, \frac{1}{2})$. Note that (ρ_0, ρ_1)-ABG condition is much weaker than the ρ_1-SBG condition, and only requires the average of the bias to shrink relative to the average true gradient, up to a diminishing term.

We then show that SBG or ABG condition is sufficient for the general algorithm to achieve the same efficiency as unbiased counterparts in Theorem 3.2 below. The proof of Theorem 3.2 directly follows from (3.4) and the bias growth condition. We provide a detailed proof of it in Appendix A.2.1 for completeness.

Theorem 3.2 *Suppose Assumptions 2.1 and 3.1 hold, and either the ρ_1-SBG condition or the (ρ_0, ρ_1)-ABG condition holds for some $\rho_0 \geq 0$ and $\rho_1 \in [0, \frac{1}{2})$. Let the choice of parameters be specified as in Theorem 2.7 with $\rho_A \in (0, \frac{1}{3}]$. Then the output of the general framework in Algorithm 1 satisfies that*

$$\mathbb{E}_\tau \|\nabla F(x_\tau)\|^2 \leq \frac{2[F(x_0) - F^*] + 2\sigma^2 + 2\rho_0}{(1 - 2\rho_1)\alpha T},$$

where x_τ is a random output uniformly chosen from the iterates $\{x_t\}_{t=0}^{T-1}$. This further implies that to achieve an ε-stationary point, the total oracle complexity is $\mathcal{O}(\varepsilon^{-3})$ as shown in Remark 2.9.

The complexity $\mathcal{O}(\varepsilon^{-3})$ is known to be optimal even assuming unbiased oracle as in Chapter 2. To the best of our knowledge, this is the first general analysis for the convergence of stochastic variance reduction methods with biased stochastic oracles in the non-convex setting. It is worth mentioning that our analysis is much simpler comparing to existing analysis with unbiased oracle and accommodates a broad range of parameter choices and algorithms as shown in Section 2.4.

Theorem 3.2 suggests that using the general framework even for functions with biased stochastic oracle, as long as the SBG or ABG condition holds, we can achieve the optimal rate. This shows the effectiveness of our general analysis. In the next sections, we give some specific applications for which we can use the general framework to achieve the optimal complexity. In these applications, the bias term or the gradient error term can be controlled similarly to the bias growth condition. We find that although ABG condition is very general and characterizes a certain class of biased optimization problems that can be solved by our general framework to obtain optimal convergence rate, it is often difficult to directly verify in practice. We add more discussions on the ABG condition here before moving to specific examples.

By (3.4), if the average bias can be bounded by the average square norm of the recursive gradient up to a diminishing term, we can still prove the optimal rate. That is,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|B(x_t)\|^2 \leq \frac{\rho_0}{\alpha T} + \rho_2 \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2. \tag{3.5}$$

If we have condition (3.5) for some constant $\rho_2 \in (0, \frac{1}{3}]$, by (3.4) with the choice that $\rho_A \in (0, \frac{1}{3}]$, we directly obtain that

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\leq \frac{2[F(x_0) - F^*] + 2\sigma^2 + 2\rho_0}{\alpha T} - \left(\frac{2}{3} - 2\rho_A - 2\rho_2\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 \\ &\leq \frac{2[F(x_0) - F^*] + 2\sigma^2 + 2\rho_0}{\alpha T}.\end{aligned}$$

Similar to Theorem 2.8, we can achieve the $\mathcal{O}(\varepsilon^{-3})$ oracle complexity. As we will see in the following sections, it is often easier to bound the average bias or the average gradient error by $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2$, thus not affecting too much. Therefore, we are actually only verifying condition (3.5) in specific applications.

Although condition (3.5) is easier to verify, it is less general and informative than the ABG condition since it would rely on the specific choice of the gradient estimator. We then show that if the recursive gradient estimator is used as in the general framework, we can prove that condition (3.5) implies the ABG condition.

Suppose condition (3.5) holds. Since $\mathbb{E}\|h_t\|^2 \leq 2\mathbb{E}\|\nabla F(x_t)\|^2 + 4A_t + 4B_t$ using the fact that $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ twice, by Theorem 2.7, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 &\leq \frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(x_t)\|^2 + \frac{4}{T}\sum_{t=0}^{T-1}A_t + \frac{4}{T}\sum_{t=0}^{T-1}B_t \\ &\leq \frac{4\sigma^2}{\alpha T} + 4\rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(x_t)\|^2 + \frac{4}{T}\sum_{t=0}^{T-1}B_t.\end{aligned}$$

Choosing $\rho_A < 1/4$, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 \leq \frac{4\sigma^2}{(1-4\rho_A)\alpha T} + \frac{2}{(1-4\rho_A)T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(x_t)\|^2 + \frac{4}{(1-4\rho_A)T}\sum_{t=0}^{T-1}B_t.$$

Plugging this bound back into (3.5), we recover ABG condition by properly selecting ρ_2. We prove that ABG condition also holds if we can show condition (3.5). All discussions above suggest that we can directly verify condition (3.5) in practice.

In the next sections, we apply the general framework in Algorithm 1 to several specific examples including stochastic bilevel optimization, stochastic minimax optimization and stochastic compositional optimization. In all these examples, similarly to condition (3.5), we can prove that the average bias or the average gradient error can be bounded by the average recursive gradient up to a diminishing term, and thus we can still obtain the optimal complexity.

3.2 Stochastic Bilevel Optimization

We first apply the general framework to non-convex strongly-convex stochastic bilevel optimization. Stochastic bilevel optimization solves the following problem:

$$\begin{aligned}&\min_{x\in\mathbb{R}^{d_1}} F(x) := \mathbb{E}_\xi[f(x, y^*(x); \xi)], \quad \text{(upper)} \\ &\text{s.t. } y^*(x) \in \arg\min_{y\in\mathbb{R}^{d_2}} G(x,y) = \mathbb{E}_\zeta[g(x,y;\zeta)]. \quad \text{(lower)}\end{aligned} \tag{3.6}$$

We assume that both upper-level function $F : \mathbb{R}^{d_1} \to \mathbb{R}$ and lower-level function $G : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ are continuously differentiable. The difficulty of bilevel optimization lies in the fact that it is nested. The upper-level function depends on $y^*(x)$, which is an optimal solution to another lower-level optimization problems. In order to construct an upper-level gradient estimator and minimize $F(x)$, we need to solve another problem $\min_y G(x, y)$ at each iteration. When there is no closed-form solution of $y^*(x)$, we can only obtain an approximation and thus create bias. Such kind of optimization problem has regained attention these years since its application in model-agnostic meta-learning [41, 15, 42], hyper-parameter optimization [43, 16], reinforcement learning [17], representation learning [44] and continual learning [45].

We consider the non-convex strongly-convex setting when $G(x, y)$ is μ_g-strongly convex in y for any x. Therefore, the lower-level optimal solution $y^*(x)$ is unique for any given x. This setting corresponds to adding strongly-convex regularization and also widely-used in many aforementioned machine learning applications. Even when lower-level objective is strongly-convex, bilevel optimization is still a challenging problem if no closed-form solution of $y^*(x)$ is known since we can only construct biased gradient estimators.

In the following, we first give a short review of related literature, then we list all the necessary assumptions and explain how to construct the biased gradient oracle, finally we present our algorithm and give its convergence analysis.

Related Work

Ghadimi and Wang [21] were the first to derive finite time convergence rate for non-convex strongly-convex stochastic bilevel optimization. The algorithm they proposed uses a double-loop structure and achieves $\tilde{\mathcal{O}}(\varepsilon^{-6})$ complexity to obtain an ε-stationary point as defined in Definition 2.3. In each iteration t of the outer-loop for the update of x, the algorithm runs multiple gradient descent in the inner-loop to minimize $G(x_t, y)$ and find an accurate approximation of $y^*(x_t)$. Ji et al. [46] improved the rate to $\mathcal{O}(\tilde{\varepsilon}^{-4})$ by using large batch sizes.

Hong et al. [17] showed that double-loop structure is not necessary. In the early stages when x_t is far from optimal, there is no need to solve lower-level optimization to a high accuracy. They thus proposed a single-loop algorithm that updates one step of x and then one step of y alternately. The algorithm has to use two-timescale stepsizes for upper and lower-level updates to control the bias, and achieves $\tilde{\mathcal{O}}(\varepsilon^{-5})$ complexity. Chen et al. [32] and Khanduri [47] added correction terms to either lower-level update or upper-level update, and were able to improve the rate to $\tilde{\mathcal{O}}(\varepsilon^{-4})$. A recent work by Chen et al. [48] found that the optimal solution $y^*(x)$ is actually smooth. With this hidden smoothness property, they were able to prove that a single-loop algorithm with simple SGD step on both levels achieves the $\tilde{\mathcal{O}}(\varepsilon^{-4})$ complexity.

Note that $\mathcal{O}(\varepsilon^{-4})$ is the optimal rate even for classical non-convex minimization [11]. As stochastic bilevel optimization is also a special case of non-convex optimization, we are able to obtain near-optimal algorithms. This suggests that non-convex strongly-convex stochastic bilevel optimization is no-harder than classical non-convex minimization measured in sample complexity.

We also know that the optimal complexity is $\mathcal{O}(\varepsilon^{-3})$ if we assume that the stochastic gradient satisfies average smoothness assumption [11]. Thus we wonder whether this rate is also achievable for bilevel optimization. We are the first to give an affirmative answer. As shown later, we propose a single-loop algorithm with both upper and lower-level gradients estimated using the general framework, and prove its $\tilde{\mathcal{O}}(\varepsilon^{-3})$ sample complexity. The results imply that any variance reduction methods are able to achieve

Table 3.1: Comparison of the sample complexity of different algorithms to achieve an ε-stationary point defined in Definition 2.3 for solving non-convex strongly-convex stochastic bilevel optimization. $\tilde{\mathcal{O}}$ in the table hides additional logarithmic terms in ε^{-1}. AS means that the oracle satisfies average smoothness assumption as in Assumption 3.1, and Non-AS represents the case when average smoothness condition (3.2) does not hold.

Algorithm	Structure	Batch Size	Oracle	Complexity
Ghadimi and Wang [21]	Double-Loop	$\tilde{\mathcal{O}}(1)$	Non-AS	$\tilde{\mathcal{O}}(\varepsilon^{-6})$
Ji et al. [46]	Double-Loop	$\tilde{\mathcal{O}}(\varepsilon^{-2})$	Non-AS	$\tilde{\mathcal{O}}(\varepsilon^{-4})$
Hong et al. [17]	Single-Loop	$\tilde{\mathcal{O}}(1)$	Non-AS	$\tilde{\mathcal{O}}(\varepsilon^{-5})$
Chen et al. [32]	Single-Loop	$\mathcal{O}(1)$	Non-AS	$\tilde{\mathcal{O}}(\varepsilon^{-4})$
Chen et al. [48]	Single-Loop	$\tilde{\mathcal{O}}(1)$	Non-AS	$\tilde{\mathcal{O}}(\varepsilon^{-4})$
Khanduri et al. [47]	Single-Loop	$\tilde{\mathcal{O}}(1)$	AS	$\tilde{\mathcal{O}}(\varepsilon^{-4})$
Khanduri et al. [49]	Single-Loop	$\tilde{\mathcal{O}}(1)$	AS	$\tilde{\mathcal{O}}(\varepsilon^{-3})$
Guo and Yang. [50]	Single-Loop	$\tilde{\mathcal{O}}(1)$	AS	$\tilde{\mathcal{O}}(\varepsilon^{-3})$
Yang et al. [51]	Double-Loop	$\tilde{\mathcal{O}}(\varepsilon^{-2})$	AS	$\tilde{\mathcal{O}}(\varepsilon^{-3})$
This work ($\eta = 0$)	Double-Loop	$\tilde{\mathcal{O}}(\varepsilon^{-2})$	AS	$\tilde{\mathcal{O}}(\varepsilon^{-3})$
This work ($\eta > 0$)	Single-Loop	$\tilde{\mathcal{O}}(1)$	AS	$\tilde{\mathcal{O}}(\varepsilon^{-3})$

the optimal rate for bilevel optimization when applied on both levels. After we submitted our work to a conference, we found that there are several recent papers [49, 50, 51] obtaining the same complexity result. They either used SPIDER [7] or STORM [10] to construct the gradient estimator and thus can be regarded as special case of our results. See Table 3.1 for a comparison of different algorithms we mentioned above.

Assumptions and Basic Properties

Before presenting our algorithm and its convergence analysis, we first list all the necessary assumptions which are common in the related literature.

Assumption 3.3 *The upper-level function $f(x,y;\xi)$ is differentiable and satisfies that*

- *$\nabla_x f(x,y;\xi)$ is Lipschitz continous w.r.t. both x and y for any given ξ with constant $L_{f_x} > 0$, i.e. $\|\nabla_x f(x_1,y_1;\xi) - \nabla_x f(x_2,y_2;\xi)\| \le L_{f_x}(\|x_1 - x_2\| + \|y_1 - y_2\|)$.*
- *$\nabla_y f(x,y;\xi)$ is Lipschitz continous w.r.t. both x and y for any given ξ with constant $L_{f_y} > 0$, i.e. $\|\nabla_y f(x_1,y_1;\xi) - \nabla_y f(x_2,y_2;\xi)\| \le L_{f_y}(\|x_1 - x_2\| + \|y_1 - y_2\|)$.*
- *For any x and y, $\|\nabla_y f(x,y;\xi)\| \le C_{f_y}$ for some constant $C_{f_y} > 0$.*

The lower-level function $g(x,y;\zeta)$ is twice-differentiable and satisfies that

- *For any given x, $g(x,y;\zeta)$ is μ_g-strongly convex and L_g-smooth w.r.t. y, and that $\mu_g I \preceq \nabla^2_{yy} g(x,y;\zeta) \preceq L_g I$.*
- *$\nabla^2_{xy} g(x,y;\zeta)$ is Lipschitz continous w.r.t. both x and y for any given ζ with constant $L_{g_{xy}} > 0$. $\nabla^2_{yy} g(x,y;\zeta)$ is Lipschitz continous w.r.t. both x and y for any given ζ with constant $L_{g_{yy}} > 0$.*
- *For any x and y, $\|\nabla^2_{xy} g(x,y;\zeta)\|^2 \le C_{g_{xy}}$ for some constant $C_{g_{xy}} > 0$.*

The estimates $\nabla_x f(x,y;\xi)$, $\nabla_y f(x,y;\xi)$, $\nabla_y g(x,y;\zeta)$, $\nabla^2_{xy} g(x,y;\zeta)$ and $\nabla^2_{yy} g(x,y;\zeta)$ are unbiased and satisfy bounded gradient conditions.

- *$\mathbb{E}_\xi \|\nabla_x f(x,y;\xi) - \mathbb{E}_\xi[\nabla_x f(x,y;\xi)]\|^2 \le \sigma^2_{f_x}$ for some constant $\sigma_{f_x} > 0$.*
- *$\mathbb{E}_\xi \|\nabla_y f(x,y;\xi) - \mathbb{E}_\xi[\nabla_y f(x,y;\xi)]\|^2 \le \sigma^2_{f_y}$ for some constant $\sigma_{f_y} > 0$.*

- $\mathbb{E}_\zeta \|\nabla_y g(x,y;\zeta) - \nabla_y G(x,y)\|^2 \le \sigma_g^2$ *for some constant* $\sigma_g > 0$.
- $\mathbb{E}_\zeta \|\nabla^2_{xy} g(x,y;\zeta) - \nabla^2_{xy} G(x,y)\|^2 \le \sigma_{g_{xy}}^2$ *for some constant* $\sigma_{g_{xy}} > 0$.

Assumption 3.3 is just a summary of Assumptions 1, 2, and 3 in Ghadimi and Wang [21], but for the stochastic functions $f(x,y;\xi)$ and $g(x,y;\zeta)$. This is slightly stronger than average smoothness but we can still assume all properties hold under expectations and obtain the same results.

Let $F(x,y) := \mathbb{E}_\xi[f(x,y;\xi)]$ for simplicity. With Assumption 3.3, by implicit function theorem, the upper-level gradient is well defined [21, Lemma 2.1]:

$$\nabla F(x) = \nabla_x F(x, y^*(x)) - \nabla^2_{xy} G(x, y^*(x))[\nabla^2_{yy} G(x, y^*(x))]^{-1} \nabla_y F(x, y^*(x)).$$

Since $y^*(x)$ is generally not accessible, a common choice is to approximate it by some y and thus we define a surrogate as

$$\bar{\nabla} F(x,y) = \nabla_x F(x,y) - \nabla^2_{xy} G(x,y)[\nabla^2_{yy} G(x,y)]^{-1} \nabla_y F(x,y).$$

As the surrogate contains the need to inverse the Hessian matrix, we construct the following widely-used biased oracle [21, Algorithm 3] that computes an approximation:

$$\bar{\nabla} f(x,y;\bar{\xi}) = \nabla_x f(x,y;\xi) - \nabla^2_{xy} g(x,y;\zeta) \left[\frac{K}{L_g} \prod_{i=1}^{k} \left(I - \frac{1}{L_g} \nabla^2_{yy} g(x,y;\zeta^{(i)}) \right) \right] \nabla_y f(x,y;\xi),$$

where $\bar{\xi} := \{\xi, \zeta, \zeta^{(1)}, \zeta^{(2)}, \cdots, \zeta^{(K)}\}$ uses a sample ξ and $K+1$ i.i.d. samples ζ and $\{\zeta^{(i)}\}_{i=1}^K$, and k is selected uniformly at random from $\{1, ..., K\}$. For the lower-level, we simply use the unbiased stochastic oracle $\nabla_y g(x,y;\zeta)$.

Under Assumption 3.3, we can show that $F(x)$ is L_f-smooth and both oracles satisfy Assumption 3.1, i.e. the bounded variance assumption with constants σ_f^2 and σ_g^2 respectively and the average smoothness assumption with constants ℓ_f and ℓ_g. As shown in the two lemmas below.

Lemma 3.4 *[21, Lemma 2.2] Under assumptions 3.3, we can show that $F(x)$ is L_F-smooth and $y^*(x)$ is L_y-Lipschitz. Moreover, the error $\|\bar{\nabla} F(x,y) - \nabla F(x)\|$ can be bounded. That is, for all x_1, x_2, $x \in \mathbb{R}^{d_1}$ and $y \in \mathbb{R}^{d_2}$,*

$$\|\nabla F(x_1) - \nabla F(x_2)\| \le L_F \|x_1 - x_2\|, \tag{3.7}$$
$$\|\bar{\nabla} F(x,y) - \nabla F(x)\| \le L \|y - y^*(x)\|, \tag{3.8}$$
$$\|y^*(x_1) - y^*(x_2)\| \le L_y \|x_1 - x_2\|. \tag{3.9}$$

For the constants defined in Assumption 3.3, the above parameters are:

$$L = L_{f_x} + \frac{L_{f_y} C_{g_{xy}}}{\mu_g} + C_{f_y} \left(\frac{L_{g_{xy}}}{\mu_g} + \frac{L_{g_{yy}} C_{g_{xy}}}{\mu_g^2} \right),$$
$$L_F = L_{f_x} + \frac{(L_{f_y} + L) C_{g_{xy}}}{\mu_g} + C_{f_y} \left(\frac{L_{g_{xy}}}{\mu_g} + \frac{L_{g_{yy}} C_{g_{xy}}}{\mu_g^2} \right), \; L_y = \frac{L_g}{\mu_g}.$$

The above lemma guarantees that $F(x)$ is well-defined and the error induced by approximating $y^*(x)$ by some y can be controlled.

Lemma 3.5 *Under Assumption 3.3, we can show that both the upper and lower-level oracle satisfy bounded variance assumption and average smoothness assumption. For the upper-level biased gradient oracle $\bar{\nabla} f(x,y;\xi)$, we have*

$$\mathbb{E}_{\bar{\xi}}\|\bar{\nabla} f(x,y;\bar{\xi}) - \mathbb{E}_{\bar{\xi}}[\bar{\nabla} f(x,y;\bar{\xi})]\| \le \sigma_f^2, \tag{3.10}$$

$$\mathbb{E}_{\bar{\xi}}\|\bar{\nabla} f(x_1,y_1;\bar{\xi}) - \bar{\nabla} f(x_2,y_2;\bar{\xi})\| \le \ell_f^2(\|x_1-x_2\|^2 + \|y_1-y_2\|^2), \tag{3.11}$$

where (3.10) *comes from Lemma 11 in Hong et al. [17] for*

$$\sigma_f^2 = \sigma_{f_x}^2 + \frac{3}{\mu_g^2}\Big[(\sigma_{f_y}^2 + C_{f_y}^2)(\sigma_{g_{xy}}^2 + 2C_{g_{xy}}^2) + \sigma_{f_y}^2 C_{g_{xy}}^2\Big],$$

and (3.11) *follows from Lemma 4.1 in Khanduri et al. [47] for*

$$\ell_f^2 = 2L_{f_x}^2 + \frac{6C_{g_{xy}}^2 L_{f_y}^2 K}{2\mu_g L_g - \mu_g^2} + \frac{6L_{g_{xy}}^2 C_{f_y}^2 K}{2\mu_g L_g - \mu_g^2} + \frac{6C_{g_{xy}}^2 C_{f_y}^2 L_{g_{yy}}^2 K^3}{(L_g-\mu_g)^2(2\mu_g L_g - \mu_g^2)}.$$

For the lower-level unbiased oracle $\nabla_y g(x,y;\zeta)$, we also have

$$\mathbb{E}_{\zeta}\|\nabla_y g(x,y;\zeta) - \nabla_y G(x,y)\|^2 \le \sigma_g^2, \tag{3.12}$$

$$\mathbb{E}_{\zeta}\|\nabla g(x_1,y_1;\zeta) - \nabla g(x_2,y_2;\zeta)\| \le \ell_g^2(\|x_1-x_2\|^2 + \|y_1-y_2\|^2). \tag{3.13}$$

Both (3.12) *and* (3.13) *directly comes from Assumption 3.3.*

The above lemma shows that the upper-level biased oracle satisfies Assumption 3.1 and the lower-level unbiased oracle satisfies Assumption 2.2. Thus we can apply variance reduction methods to obtain better convergence rate.

Algorithm and Convergence Analysis

We propose the following single-loop algorithm

$$y_{t+1} = y_t - \beta h_t^g,$$
$$x_{t+1} = x_t - \alpha h_t^f,$$

with both h_t^g estimating $\nabla_y G(x_t,y_t)$ and h_t^f estimating $\mathbb{E}_{\bar{\xi}_t}[\bar{\nabla} f(x_t,y_{t+1};\bar{\xi}_t)]$ via the general framework in Algorithm 1, and α, β be the corresponding stepsizes. The detailed algorithm is described in Algorithm 2.

Similar to the analysis before, the key is to bound the gradient error $\mathbb{E}\|h_t^f - \nabla F(x_t)\|^2$, and it can be decomposed to the estimation error $A_t^f := \mathbb{E}\|h_t^f - \mathbb{E}_{\bar{\xi}_t}[\bar{\nabla} f(x_t,y_{t+1};\bar{\xi}_t)]\|^2$ and the bias $\|B(x_t,y_{t+1})\|^2 = \|\mathbb{E}_{\bar{\xi}_t}[\bar{\nabla} f(x_t,y_{t+1};\bar{\xi}_t)] - \nabla F(x_t)\|^2$.

We first handle the estimation error A_t^f. In the following analysis, we also denote $A_t^g := \mathbb{E}\|h_t^g - \nabla_y G(x_t,y_t)\|^2$ as the lower-level estimation error and call A_t^f the upper-level estimation error. Since now the necessary average smoothness condition (3.11) to obtain the desired bound of A_t^f depends on y as well, with a slight modification of Lemma 2.5, we have

$$\begin{aligned} A_{t+1}^f &\le (1-\eta)A_t^f + 2\sigma_f^2\frac{\eta^2}{S} + \frac{2}{S}\mathbb{E}\|\bar{\nabla} f(x_{t+1},y_{t+2};\bar{\xi}_{t+1}) - \bar{\nabla} f(x_t,y_{t+1};\bar{\xi}_{t+1})\|^2 \\ &\le (1-\eta)A_t^f + 2\sigma_f^2\frac{\eta^2}{S} + 2\ell_f^2\frac{\alpha^2}{S}\mathbb{E}\|h_t^f\|^2 + 2\ell_f^2\frac{\beta^2}{S}\mathbb{E}\|h_{t+1}^g\|^2. \end{aligned}$$

Algorithm 2 Variance Reduction for Stochastic Bilevel Optimization

Input: $T, Q, D, S, x_0, y_0, \alpha, \beta, \eta$.
  for $t = 0, 1, \cdots, T-1$ **do**
    if $t \equiv 0 \pmod{Q}$ **then**
      Sample $D_t^g = \{\zeta_t^1, \zeta_t^2, \cdots, \zeta_t^D\}$ and compute $h_t^g = \frac{1}{D}\sum_{i=1}^{D} \nabla_y g(x_t, y_t; \zeta_t^i)$.
      $y_{t+1} = y_t - \beta h_t^g$.
      Sample $D_t^f = \{\bar{\xi}_t^1, \bar{\xi}_t^2, \cdots, \bar{\xi}_t^D\}$ and compute $h_t^f = \frac{1}{D}\sum_{i=1}^{D} \bar{\nabla} f(x_t, y_{t+1}; \bar{\xi}_t^i)$.
      $x_{t+1} = x_t - \alpha h_t^f$.
    else
      Sample $S_t^g = \{\zeta_t^1, \zeta_t^2, \cdots, \zeta_t^S\}$.
      $h_t^g = (1-\eta)\Big(h_{t-1}^g - \frac{1}{S}\sum_{i=1}^{S} \nabla_y g(x_{t-1}, y_{t-1}; \zeta_t^i)\Big) + \frac{1}{S}\sum_{i=1}^{S} \nabla_y g(x_t, y_t; \zeta_t^i)$.
      $y_{t+1} = y_t - \beta h_t^g$.
      Sample $S_t^f = \{\bar{\xi}_t^1, \bar{\xi}_t^2, \cdots, \bar{\xi}_t^S\}$.
      $h_t^f = (1-\eta)\Big(h_{t-1}^f - \frac{1}{S}\sum_{i=1}^{S} \bar{\nabla} f(x_{t-1}, y_t; \bar{\xi}_t^i)\Big) + \frac{1}{S}\sum_{i=1}^{S} \bar{\nabla} f(x_t, y_{t+1}; \bar{\xi}_t^i)$.
      $x_{t+1} = x_t - \alpha h_t^f$.
    end if
  end for
Output: x_τ with τ chosen uniformly at random from $\{0, 1, \cdots, T-1\}$.

Note that there is an additional error term in $\mathbb{E}\|h_{t+1}^g\|^2$. Thus we also need to apply variance reduction for the lower-level gradient to cancel this additional term. We then obtain the bound for both upper and lower-level gradient estimation error in Lemma 3.6. This is a direct consequence of Theorem 2.7 and the average smoothness assumption (3.11) and (3.13).

Lemma 3.6 *Supposing Assumption 3.3 holds, with all the settings in Algorithm 2 for parameters D, S, Q, η and α selected according to Theorem 2.7 for some constant $\rho_A > 0$ we define later and the choice that $\beta = \alpha/c_\beta$ for some constant c_β we define later, i.e.,*

(i) $\eta = 0$, $D = \alpha T$, $S = \rho_A^{-1/2}\ell_f \alpha^{3/2} T^{1/2}$, $Q = \rho_A^{1/2}(2\ell_f)^{-1}\alpha^{-1/2}T^{1/2}$ *for any α such that $(\rho_A \ell_f^{-2})^{1/3} T^{-1/3} \le \alpha \le 1/(3L_F)$ with the smooth parameter L_F defined in* (3.7) *and $\beta = \alpha/c_\beta$;*

(ii) $\eta = 2\ell_f^2\alpha^2/\rho_A$, $\alpha = \rho_A^{1/3}(8\ell_f^2 \cdot T)^{-1/3}$, $\beta = \alpha/c_\beta$, $D = \rho_A \ell_f^{-2}\alpha^{-1}$, $S = 1$ *and* $Q = T$,

then we have

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t^f \le \frac{\sigma_f^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t^f\|^2 + \frac{\rho_A}{c_\beta^2}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^g\|^2,$$
$$\frac{1}{T}\sum_{t=0}^{T-1} A_{t+1}^g \le \frac{\sigma_g^2}{\alpha T} + \rho_A \frac{\ell_g^2}{\ell_f^2}\cdot\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t^f\|^2 + \frac{\rho_A}{c_\beta^2}\frac{\ell_g^2}{\ell_f^2}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^g\|^2.$$

Then we analysis the bias $\|B(x_t, y_{t+1})\| = \|\mathbb{E}_{\bar{\xi}_t}[\bar{\nabla} f(x_t, y_{t+1}; \bar{\xi}_t)] - \nabla F(x_t)\|$. With Assumption 3.3, the bias term can be bounded by

$$\begin{aligned}\|B(x_t, y_{t+1})\| &\le \|\mathbb{E}_{\bar{\xi}_t}[\bar{\nabla} f(x_t, y_{t+1}; \bar{\xi}_t)] - \bar{\nabla} F(x_t, y_{t+1})\| + \|\bar{\nabla} F(x_t, y_{t+1}) - \nabla F(x_t)\| \\ &\le \underbrace{C_{g_{xy}} C_{f_y} \mu_g^{-1}(1 - \mu_g/L_g)^K}_{\text{approximation of inverse Hessian}} + \underbrace{L\|y_{t+1} - y^*(x_t)\|}_{\text{approximation of } y^*(x)}, \end{aligned} \tag{3.14}$$

where the first part follows from Lemma 11 in Hong et al. [17], and the second part holds by (3.8) in Lemma 3.4. Choosing $K = \mathcal{O}(\log \varepsilon^{-1})$, the bias which comes from approximating Hessian inverse is less than ε. The rest is to bound the bias introduced by approximating $y^*(x_t)$ by y_{t+1}, i.e. $\|y_{t+1} - y^*(x_t)\|^2$. We bound it by lower-level descent in the lemma below.

Lemma 3.7 *Under Assumption 3.3, for Algorithm 2 with the choice that $\beta = \alpha / c_\beta$ for some constant c_β to be determined, assuming $\beta \leq \frac{1}{2(\mu_g + L_g)}$, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 \leq \frac{2c_\beta}{c_g \alpha T}\|y_1 - y^*(x_0)\|^2 + \frac{4L_y^2 c_\beta^2}{c_g^2} \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t^f\|^2 - \frac{1}{\mu_g T}\sum_{t=0}^{T-1} \mathbb{E}\|h_{t+1}^g\|^2 + \left(\frac{4}{\mu_g} + \frac{2}{c_g^2}\right)\frac{1}{T}\sum_{t=0}^{T-1} A_{t+1}^g,$$

where $c_g = \frac{\mu_g}{\mu_g + L_g}$.

Note that the lemma above is similar to what we have in (3.3), with an additional error term bounded by $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t^f\|^2$. That is why we can achieve the optimal rate by the discussion of condition (3.5) at the end of Section 3.1. With the help of Lemma 3.6 to handle the estimation error and Lemma 3.7 to bound the bias, we can show the following convergence theorem. The detailed proof of Lemma 3.6, 3.7 and Theorem 3.8 can be found in Appendix A.2.2.

Theorem 3.8 *For non-convex strongly-convex stochastic bilevel optimization defined in* (3.6)*, under Assumption 3.3, with the parameters D, S, Q, η and α selected according to Lemma 3.6 for*

$$\rho_A = \min\left\{\frac{1}{12}, \frac{\mu_g \ell_f^2}{96L^2\ell_g^2}, \frac{c_g^2\ell_f^2}{48L^2\ell_g^2}, \frac{c_\beta^2\ell_f^2}{16\ell_g^2}, \frac{c_g^2 c_\beta^2 \ell_f^2}{8\mu_g\ell_g^2}, \frac{c_\beta^2 L^2}{\mu_g}\right\},$$

and the choice of $\beta = \frac{\alpha}{c_\beta}$ for constant $c_\beta = \frac{c_g}{4\sqrt{6}LL_y}$ and $c_g = \frac{\mu_g}{\mu_g + L_g}$, supposing $\alpha \leq \frac{c_\beta}{2(\mu_g + L_g)}$, with $K = \mathcal{O}(\log \varepsilon^{-1})$, we can show that the output of Algorithm 2 satisfies

$$\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \leq \left[2[F(x_0) - F^*] + \frac{8c_\beta L^2\|y_1 - y^*(x_0)\|^2}{c_g} + 2\sigma_f^2 + \left(\frac{16}{\mu_g} + \frac{8}{c_g^2}\right)L^2\sigma_g^2 + 2\right]/(\alpha T).$$

To make sure that x_τ is an ε-stationary point, similar to the analysis in Remark 2.9, we can show that the sample complexity is $\tilde{\mathcal{O}}(\varepsilon^{-3})$.

Note that constructing $\bar{\nabla} f(x, y; \bar{\xi})$ uses $K + 2 = \mathcal{O}(\log \varepsilon^{-1})$ samples. That's why the batch size for algorithms in Table 3.1 is at least $\tilde{\mathcal{O}}(1)$. Hence, the sample complexity is also $\tilde{\mathcal{O}}(\varepsilon^{-3})$. Compared with existing single-loop algorithms in Hong et al. [17] and Chen et al. [32], we improve their complexity from $\tilde{\mathcal{O}}(\varepsilon^{-5})$ and $\tilde{\mathcal{O}}(\varepsilon^{-4})$ to $\tilde{\mathcal{O}}(\varepsilon^{-3})$ under the average smoothness condition, closing the gap between the upper-bounds and the lower-bound for non-convex stochastic minimization problems. The contemporaneous works [49, 50, 51] obtained the same rate as we do. However, they all focused on a specific variance reduction technique, e.g. STORM. In comparison, we focus on a general variance reduction framework that includes STORM as a special case as shown in Section 2.4. Our framework allows more versatility in parameter choices during implementation, and our analysis provides guidelines for design of new algorithms on other biased stochastic optimization problem rather than just on bilevel optimization.

3.3 Stochastic Minimax Optimization

In this section, we consider the stochastic minimax optimization:

$$\min_{x\in\mathbb{R}^{d_1}} F(x) := \max_{y\in\mathcal{Y}\subseteq\mathbb{R}^{d_2}} G(x,y), \quad \text{where } G(x,y) := \mathbb{E}_\xi[g(x,y;\xi)], \tag{3.15}$$

where $G : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ is continuously differentiable. Minimax optimization has many interesting applications in machine learning including generative adversarial networks (GANs) [52], robust optimization [53, 54], adversarial machine learning [13, 55], reinforcement learning [56, 57] and so on. There are plenty of works establishing convergence theory for solving (3.15) under strongly-convex strongly-concave setting [58, 59] or general convex-concave setting [60, 61, 62]. However, non-convex minimax problems are more popular in practice, especially when neural networks are involved as in GANs. Therefore, in this work, we focus on the smooth non-convex stochastic setting with non-convex strongly-concave case shown in Section 3.3.1 and non-convex P-Ł case given in Section 3.3.2. Minimax optimization can be regarded as a special case of bilevel optimization when lower-level objective in (3.6) is replaced with $-\mathbb{E}_\xi[f(x,y;\xi)]$. Thus we will see that the algorithm and its analysis in this section are very much similar to what we have in Section 3.2 for bilevel optimization.

3.3.1 Non-Convex Strongly-Concave Case

We first consider the non-convex strongly-concave setting where $G(x,y)$ is μ-strongly concave in y. See Table 3.2 for a brief comparison of sample complexity of the related algorithms to obtain an ε-stationary point. Without average smoothness assumption, Rafique et al. [63] and Jin et al. [64] achieved the optimal $\mathcal{O}(\varepsilon^{-4})$ rate through a double-loop algorithm. Lin et al. [18] proposed a single-loop algorithm and achieved the optimal rate with large batch sizes. The single-loop algorithm for solving (3.15) is also called gradient descent ascent (GDA), i.e. one descent step for x and one ascent step for y. Chen et al. [48] extended their work for bilevel optimization to minimax case and proved that single-loop alternating GDA without large batch sizes is already optimal. However, assumptions in their paper are much stronger.

When average smoothness assumption holds, Luo et al. [31] and Xu et al. [33] obtained the optimal $\mathcal{O}(\varepsilon^{-3})$ sample complexity when applying SARAH on both updates of x and y. For each update of x, the algorithms run multi-steps for y, and thus are actually triple-loop algorithms. Huang et al. [34] used STORM and obtained a single-loop algorithm achieving the optimal complexity.

Before presenting our results, we first make the following standard assumptions. Suppose that $\mathcal{Y}$ is convex and compact, G is L-smooth with respect to (x,y) and $G(x,y)$ is μ-strongly concave in y for any given x, as listed below.

Assumption 3.9 *The objective G and domain $\mathcal{Y}$ satisfy*

- *G is L-smooth in (x,y) and $G(x,\cdot)$ is μ-strongly concave for any given x.*
- *$\mathcal{Y}$ is convex and compact.*

The condition number is defined as $\kappa := L/\mu$. When Assumption 3.9 holds, we have the following lemma to guarantee that $F(x)$ is L_F-smooth and $y^*(x) := \arg\max_{y\in\mathcal{Y}} G(x,y)$ is κ-Lipschitz continuous, which is simlar to Lemma 3.4 for bilevel optimization. The lemma also shows how to compute $\nabla F(x)$, as given by Danskin's Theorem [65] when $\mathcal{Y}$ is convex and compact.

Table 3.2: Comparison of the sample complexity of different algorithms to achieve an ε-stationary point defined in Definition 2.3 for solving non-convex strongly-concave stochastic minimax optimization. $\tilde{\mathcal{O}}$ in the table hides additional logarithmic terms in ε^{-1}. AS means that the oracle satisfies average smoothness assumption as in Assumption 3.1, and Non-AS represents the case when average smoothness condition (3.2) does not hold.

Algorithm	Structure	Batch Size	Oracle	Complexity
Rafique et al. [63]	Double-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-4})$
Jin et al. [64]	Double-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-4})$
Lin et al. [18]	Single-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-5})$
Lin et al. [18]	Single-Loop	$\mathcal{O}(\varepsilon^{-2})$	Non-AS	$\mathcal{O}(\varepsilon^{-4})$
Chen et al. [48]	Single-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-4})$
Luo et al. [31]	Triple-Loop	$\mathcal{O}(\varepsilon^{-2})$	AS	$\mathcal{O}(\varepsilon^{-3})$
Xu et al. [33]	Triple-Loop	$\mathcal{O}(\varepsilon^{-2})$	AS	$\mathcal{O}(\varepsilon^{-3})$
Huang et al. [34]	Single-Loop	$\mathcal{O}(1)$	AS	$\tilde{\mathcal{O}}(\varepsilon^{-3})$
This work ($\eta = 0$)	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	AS	$\mathcal{O}(\varepsilon^{-3})$
This work ($\eta > 0$)	Single-Loop	$\mathcal{O}(1)$	AS	$\mathcal{O}(\varepsilon^{-3})$

Lemma 3.10 *[18, Lemma 4.3] Supposing Assumption 3.9 holds, then for function $F(x)$ defined in* (3.15) *and $y^*(x) := \arg\max_{y\in\mathcal{Y}} G(x,y)$, we have*

- *$F(x)$ is L_F-smooth with $L_F = 2\kappa L$ and $\nabla F(x) = \nabla_z G(z, y^*(x))|_{z=x}$.*
- *$y^*(x)$ is κ-Lipschitz continuous.*

We have access to the unbiased gradient oracle $\nabla_x g(x,y;\xi)$ and $\nabla_y g(x,y;\xi)$ satisfying the following classical assumptions, i.e. bounded variance assumption and average smoothness assumption.

Assumption 3.11 *Let $\nabla g(x,y;\xi)$ denote the full gradient of g. We assume that*

- $\mathbb{E}_\xi\|\nabla g(x,y;\xi) - \nabla G(x,y)\|^2 \le \sigma^2$.
- $\mathbb{E}_\xi\|\nabla g(x_1,y_1;\xi) - \nabla g(x_2,y_2;\xi)\|^2 \le \ell^2(\|x_1-x_2\|^2 + \|y_1-y_2\|^2)$.

Similar as what we do for bilevel optimization, we estimate both $\nabla_y G(x_t,y_t)$ and $\nabla_x G(x_t,y_{t+1})$ with Algorithm 1 and obtain a single-loop algorithm that is similar to alternating GDA. We denote the estimators as h_t^y and h_t^x respectively. The algorithm updates one ascent step of y to find an approximation y_{t+1} for $y^*(x_t)$, and then updates one descent step of x using y_{t+1}. The detailed algorithm is in Algorithm 3, and the projection step of y is to ensure that $y_{t+1} \in \mathcal{Y}$.

We first analyse the estimation error $A_t^y = \mathbb{E}\|h_t^y - \nabla_y G(x_t,y_t)\|^2$ and $A_t^x = \mathbb{E}\|h_t^x - \nabla_x G(x_t,y_{t+1})\|^2$. Similar to the bilevel optimization, we have the bound in Lemma 3.12. This is simpler than Lemma 3.6 since we have the same Lipschitz parameter for both $\nabla_x g(x,y;\xi)$ and $\nabla_y g(x,y;\xi)$. The proof directly follows from Lemma 3.6 for bilevel optimization.

Lemma 3.12 *Supposing Assumption 3.11 holds, with all the settings in Algorithm 3 for parameters D, S, Q, η and α selected according to Theorem 2.7 for some constant $\rho_A > 0$ we define later and the choice that $\beta = \alpha/c_\beta$ for some constant c_β we define later, i.e.,*

(i) *$\eta = 0$, $D = \alpha T$, $S = \rho_A^{-1/2}\ell\alpha^{3/2}T^{1/2}$, $Q = \rho_A^{1/2}(2\ell)^{-1}\alpha^{-1/2}T^{1/2}$ for any α such that $(\rho_A\ell^{-2})^{1/3}T^{-1/3} \le \alpha \le 1/(3L_F)$ with the smooth parameter L_F defined in Lemma 3.10 and $\beta = \alpha/c_\beta$;*

(ii) *$\eta = 2\ell^2\alpha^2/\rho_A$, $\alpha = \rho_A^{1/3}(8\ell^2\cdot T)^{-1/3}$, $\beta = \alpha/c_\beta$, $D = \rho_A\ell^{-2}\alpha^{-1}$, $S = 1$ and $Q = T$,*

Algorithm 3 Variance Reduction for Stochastic Minimax Optimization

Input: $T, Q, D, S, x_0, y_0, \alpha, \beta, \eta$.
  for $t = 0, 1, \cdots, T-1$ **do**
    if $t \equiv 0 \pmod{Q}$ **then**
      Sample $D_t^y = \{\xi_t^1, \xi_t^2, \cdots, \xi_t^D\}$ and compute $h_t^y = \frac{1}{D}\sum_{i=1}^D \nabla_y g(x_t, y_t; \xi_t^i)$.
      $y_{t+1} = \Pi_{\mathcal{Y}}(y_t + \beta h_t^y)$.
      Sample $D_t^x = \{\tilde{\xi}_t^1, \tilde{\xi}_t^2, \cdots, \tilde{\xi}_t^D\}$ and compute $h_t^x = \frac{1}{D}\sum_{i=1}^D \nabla_x g(x_t, y_{t+1}; \tilde{\xi}_t^i)$.
      $x_{t+1} = x_t - \alpha h_t^x$.
    else
      Sample $S_t^y = \{\xi_t^1, \xi_t^2, \cdots, \xi_t^S\}$.
      $h_t^y = (1-\eta)\Big(h_{t-1}^y - \frac{1}{S}\sum_{i=1}^S \nabla_y g(x_{t-1}, y_{t-1}; \xi_t^i)\Big) + \frac{1}{S}\sum_{i=1}^S \nabla_y g(x_t, y_t; \xi_t^i)$.
      $y_{t+1} = \Pi_{\mathcal{Y}}(y_t + \beta h_t^y)$.
      Sample $S_t^x = \{\tilde{\xi}_t^1, \tilde{\xi}_t^2, \cdots, \tilde{\xi}_t^S\}$.
      $h_t^x = (1-\eta)\Big(h_{t-1}^x - \frac{1}{S}\sum_{i=1}^S \nabla_x g(x_{t-1}, y_t; \tilde{\xi}_t^i)\Big) + \frac{1}{S}\sum_{i=1}^S \nabla_x g(x_t, y_{t+1}; \tilde{\xi}_t^i)$.
      $x_{t+1} = x_t - \alpha h_t^x$.
    end if
  end for
Output: x_τ with τ chosen uniformly at random from $\{0, 1, \cdots, T-1\}$.

then we have

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t^x \le \frac{\sigma^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t^x\|^2 + \frac{\rho_A}{c_\beta^2} \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_{t+1}^y\|^2,$$
$$\frac{1}{T}\sum_{t=0}^{T-1} A_{t+1}^y \le \frac{\sigma^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t^x\|^2 + \frac{\rho_A}{c_\beta^2} \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_{t+1}^y\|^2.$$

Next we analyse the bias term. Although both $\nabla_y g(x_t, y_t; \xi_t)$ and $\nabla_x g(x_t, y_{t+1}; \tilde{\xi}_t)$ are unbiased, we can only construct biased oracle for $\nabla F(x_t)$ since $y^*(x_t)$ is unknown. The corresponding bias $B(x_t, y_{t+1}) \coloneqq \nabla_x G(x_t, y_{t+1}) - \nabla F(x_t)$ can be bounded by

$$\|B(x_t, y_{t+1})\| = \|\nabla_x G(x_t, y_{t+1}) - \nabla_x G(x_t, y^*(x_t))\| \le L\|y_{t+1} - y^*(x_t)\|, \tag{3.16}$$

since G is L-smooth. With the similar analysis for bilevel optimization, we obtain the following lemma to bound $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|y_{t+1} - y^*(x_t)\|^2$ which is equivalent to the bias.

Lemma 3.13 *Under Assumption 3.9 and 3.11, for Algorithm 3 with the choice that $\beta = \alpha / c_\beta$ for some constant c_β to be determined, assuming $\beta \le \frac{1}{4\kappa\mu}$, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 \le \frac{4\kappa c_\beta}{\alpha T}\|y_1 - y^*(x_0)\|^2 + 16\kappa^4 c_\beta^2 \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t^x\|^2$$
$$- \frac{1}{\mu T}\sum_{t=0}^{T-1} \mathbb{E}\|h_{t+1}^y\|^2 + \left(\frac{4\kappa}{\mu} + 8\kappa^2\right)\frac{1}{T}\sum_{t=0}^{T-1} A_{t+1}^y,$$

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

Now we have Lemma 3.12 for the estimation error and Lemma 3.13 for the bias, thus we can obtain the following convergence theorem. The proofs for both Lemma 3.13 and Theorem 3.14 can be found in Appendix A.2.3.

Theorem 3.14 *For non-convex strongly-concave stochastic minimax optimization defined in* (3.15)*, under Assumption 3.9 and 3.11, with the parameters D, S, Q, T, η and α selected according to Lemma 3.12 for*

$$\rho_A = \min\left\{\frac{1}{12}, \frac{1}{48\kappa^2 L}, \frac{1}{96\kappa^2 L^2}, \frac{c_\beta^2}{16\kappa}, \frac{c_\beta^2}{32\kappa L}, \frac{\kappa L c_\beta^2}{2}\right\},$$

supposing that $\alpha \leq \frac{c_\beta}{4\kappa\mu}$ and $\beta = \frac{\alpha}{c_\beta}$ for a constant $c_\beta = \frac{1}{8\sqrt{3\kappa^2 L}}$, then the output of Algorithm 3 satisfies

$$\mathbb{E}_\tau \|\nabla F(x_\tau)\|^2 \leq \left[2[F(x_0) - F^*] + 8\kappa L^2 c_\beta \|y_1 - y^*(x_0)\|^2 + (2 + 8\kappa^2 L + 16\kappa^2 L^2)\sigma^2\right]/(\alpha T).$$

To make sure that x_τ is an ε-stationary point, similar to the analysis in Remark 2.9, we can show that the sample complexity is $\mathcal{O}(\varepsilon^{-3})$, which is optimal w.r.t. ε.

In this section, we provide a single-loop algorithmic framework for non-convex strongly-concave stochastic minimax optimization. It achieves a sample complexity of $\mathcal{O}(\varepsilon^{-3})$ as shown in Theorem 3.14, matching the best known result achieved by two multi-loop algorithms by Luo et al. [31] and Xu et al. [33]. However, Algorithm 3 is much simpler and more general. If one relaxes the fixed stepsize requirement as mentioned in Section 2.4, our algorithm also recovers Huang et al. [34] as a special case. Such relaxation comes with a cost and the sample complexity has an additional $\log(\varepsilon^{-1})$ term, which is $\tilde{\mathcal{O}}(\varepsilon^{-3})$.

3.3.2 Non-Convex P-Ł Case

We are also interested in the setting when $G(x, y)$ is possibly non-convex in y but satisfies the well-studied Polyak-Łojasiewicz (P-Ł) condition. We provide a definition of P-Ł condition here for reference.

Definition 3.15 (Polyak-Łojasiewicz Condition) *A differentiable function $f(x)$ with the minimum value $f^* := \min_x f(x)$ satisfies μ-Polyak-Łojasiewicz condition if for any x, it holds that*

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu(f(x) - f^*).$$

The definition of P-Ł condition does not require convexity, and that a function can be non-convex and still satisfy P-Ł condition [66]. For functions satisfying P-Ł condition, Karimi et al. [66] proved that gradient descent (GD) converges to the global minimum at a linear rate under deterministic settings and SGD also converges globally at a sublinear rate under stochastic settings. Both convergence rates match the results for strongly-convex functions. Karimi et al. [66] also showed that P-Ł condition is weaker than other conditions (e.g. quadratic growth (QG) condition defined later in Lemma 3.18) that have been explored to show linear convergence rates without strong convexity.

Actually, many optimization problems that appear in practical applications have been verified to satisfy the P-Ł condition, e.g. training over-parameterized deep networks [67] and learning linear quadratic regulator (LQR) models [68]. For the minimax optimization we discuss in this section, P-Ł condition also holds for objectives in generative adversarial imitation learning with LQR dynamics [69, 70]. This motivates us to study the convergence rate for non-convex P-Ł minimax optimization.

Nouiehed et al. [71] obtained a double-loop algorithm for solving (3.15) under P-Ł setting. The algorithm requires $\mathcal{O}(\log \varepsilon^{-1})$ ascent steps in each iteration and thus achieves $\tilde{\mathcal{O}}(\varepsilon^{-4})$

sample complexity to find an ε-stationary point. Recently, Xie et al. [72] studied non-convex P-Ł minimax optimization in the context of federated learning. Their work implies that a single-loop GDA can obtain $\mathcal{O}(\varepsilon^{-4})$ complexity when using large batch sizes, and it can also achieve $\tilde{\mathcal{O}}(\varepsilon^{-3})$ rate when using STORM [10] to estimate both gradients under average smoothness assumptions. In a different line of research, Yang et al. [73] studied the case when $G(x,y)$ satisfies P-Ł condition for both x and y (two-sided P-Ł), and proved a linear rate for alternating GDA to find a global optimum.

We consider the one-sided P-Ł case in this section and prove that Algorithm 3 still requires $\mathcal{O}(\varepsilon^{-3})$ samples to obtain an ε-stationary solution defined in Definition 2.3 for the problem (3.15). We first formally state all the assumptions. For the stochastic oracle, we use the same gradient estimates satisfying Assumption 3.11 as in Section 3.3.1 for strongly-concave case. In this section, we consider the P-Ł case as given in the assumption below.

Assumption 3.16 *The objective function $G(x,y)$ defined in* (3.15) *satisfies that*

- *G is L-smooth in (x,y).*
- *For any $x \in \mathbb{R}^{d_1}$, $-G(x,\cdot)$ satisfies P-Ł condition, that is*

$$\frac{1}{2}\|\nabla_y G(x,y)\|^2 \geq \mu(\max_y G(x,y) - G(x,y)), \qquad \forall y \in \mathbb{R}^{d_2}.$$

The above condition directly follows from Definition 3.15 noticing that $\min_y -G(x,y) = -\max_y G(x,y)$. We still denote the condition number as $\kappa := L/\mu$. The Danskin type lemma in the P-Ł case to guarantee that $F(x)$ is well-defined and show how to compute $\nabla F(x)$ is given below.

Lemma 3.17 *[71, Lemma A.5] In the minimax problem, when $-G(x,\cdot)$ satisfies P-Ł condition for any x with constant μ and $G(x,y)$ is L-smooth, then the function $F(x) := \max_{y\in\mathcal{Y}} G(x,y)$ is L_F-smooth with $L_F := 2\kappa L$ and $\nabla F(x) = \nabla_x G(x, y^*(x))$ for any $y^*(x) \in \arg\max_{y\in\mathcal{Y}} G(x,y)$.*

Following the above lemma, the compactness requirement on the domain $\mathcal{Y}$ can be removed. Thus in this section, we let $\mathcal{Y} = \mathbb{R}^{d_2}$ and the projection step in Algorithm 3 can be omitted. Then we analysis its convergence rate for P-Ł case. The bound for estimation error remains the same as in Lemma 3.12. We proceed with the bias term. We first give a helpful lemma which shows the relation between P-Ł condition and another well-known QG condition.

Lemma 3.18 *[66, Theorem 2] For some function $f(x)$, if it is L-smooth and satisfies P-Ł condition with constant μ, then it also satisfies quadratic growth (QG) condition with μ, i.e.,*

$$f(x) - f^* \geq \frac{\mu}{2}\|x_p - x\|^2,$$

where $f^* = \min_x f(x)$ is the optimal value and x_p is the projection of x onto the optimal set $\arg\min_x f(x)$.

Since $y^*(x)$ may not be unique anymore when $G(x,y)$ is possibly non-convex in y, we can not control the bias by directly bounding the error $\|y_{t+1} - y^*(x_t)\|$ like in Section 3.3.1 for the strongly-concave case. However, with the help of Lemma 3.18, if we choose $y^*(x_t)$ to be the projection of y_{t+1} onto the optimal set $\arg\max_y G(x_t,y)$, the bias term

satisfies

$$\begin{aligned}\|B(x_t, y_{t+1})\|^2 &= \|\nabla_x G(x_t, y_{t+1}) - \nabla F(x_t)\|^2 \\ &\overset{(a)}{=} \|\nabla_x G(x_t, y_{t+1}) - \nabla_x G(x_t, y^*(x_t))\|^2 \\ &\overset{(b)}{\leq} L^2 \|y_{t+1} - y^*(x_t)\|^2 \\ &\overset{(c)}{\leq} \frac{2L^2}{\mu}\Big(G(x_t, y^*(x_t)) - G(x_t, y_{t+1})\Big) \\ &\overset{(d)}{=} 2\kappa L(F(x_t) - G(x_t, y_{t+1})),\end{aligned} \tag{3.17}$$

where (a) comes from Lemma 3.17, (b) holds by smoothness of G, (c) follows from Lemma 3.18 since $-G(x_t, \cdot)$ satisfies P-Ł condition and (d) follows from definition of $F(x)$ in (3.15). Although the optimal point may not be unique, the optimal function value is unique. Then we can bound the bias in Lemma 3.19 below.

Lemma 3.19 *Under Assumption 3.16 and 3.11, for Algorithm 3 with the choice that $\beta = \alpha/c_\beta$ for some constant c_β to be determined, assuming $\beta < 1/\mu$, we have*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\Big[F(x_t) - G(x_t, y_{t+1})\Big] \leq{}& \frac{F(x_0) - G(x_0, y_1)}{\beta\mu T} + \left(1 - \frac{1}{\beta\mu}\right)\frac{F(x_0) - F(x_T)}{T} \\ &+ \frac{c_\beta}{2\mu}(3 + \alpha L)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + \frac{c_\beta}{2\mu}\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_t^x \\ &- \frac{1}{2\mu}(1 - \beta L)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^y\|^2 + \frac{1}{2\mu}\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^y.\end{aligned}$$

With Lemma 3.19 to control the bias and Lemma 3.12 to control the variance, we can derive the following theorem for the output of Algorithm 3. We provide a detailed proof of both Lemma 3.19 and Theorem 3.20 in Appendix A.2.3.

Theorem 3.20 *For non-convex P-Ł stochastic minimax optimization defined in* (3.15)*, under Assumption 3.16 and 3.11, with the parameters D, S, Q, T, η and α selected according to Lemma 3.12 for $\rho_A = c_\beta^2/4$, supposing that $\alpha < \min\Big\{\frac{3}{L}, \frac{c_\beta}{\mu}, \frac{c_\beta}{4L}\Big\}$ and $\beta = \alpha/c_\beta$ for a constant $c_\beta = 1/(36\kappa^2)$, then the output of Algorithm 3 satisfies*

$$\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \leq \frac{2[F(x_0) - F^*] + [F(x_0) - G(x_0, y_1)]/9 + (37/18 + 2\kappa^2)\sigma^2}{\alpha T},$$

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

Algorithm 3 also achieves the $\mathcal{O}(\varepsilon^{-3})$ sample complexity for non-convex P-Ł stochastic minimax optimization, matching the lower-bound for smooth non-convex optimization with average smoothness assumption [11], and we are the first to explicitly close this gap under non-convex P-Ł settings.

3.4 Stochastic Compositional Optimization

Finally we discuss the application to stochastic compositional optimization of the form:

$$\min_{x\in\mathbb{R}^{d_2}} F(x) := F_1(F_2(x)) = \mathbb{E}_\xi[f_1(F_2(x); \xi)] \quad \text{for} \quad F_2(x) := \mathbb{E}_\zeta[f_2(x; \zeta)], \tag{3.18}$$

Table 3.3: Comparison of the sample complexity of different algorithms to achieve an ε-stationary point defined in Definition 2.3 for solving stochastic compositional optimization. AS means that the oracle satisfies average smoothness assumption as in Assumption 3.1, and Non-AS represents the case when average smoothness condition (3.2) does not hold.

Algorithm	Structure	Batch Size	Oracle	Complexity
Wang et al. [19]	Single-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-8})$
Wang et al. [76]	Single-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-4.5})$
Ghadimi et al. [77]	Single-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-4})$
Chen et al. [35]	Single-Loop	$\mathcal{O}(1)$	Non-AS	$\mathcal{O}(\varepsilon^{-4})$
Zhang and Xiao [30]	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	AS	$\mathcal{O}(\varepsilon^{-3})$
Zhang and Xiao [36]	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	AS	$\mathcal{O}(\varepsilon^{-3})$
Hu et al. [78]	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	AS	$\mathcal{O}(\varepsilon^{-3})$
This work ($\eta = 0$)	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	AS	$\mathcal{O}(\varepsilon^{-3})$
This work ($\eta > 0$)	Single-Loop	$\mathcal{O}(1)$	AS	$\mathcal{O}(\varepsilon^{-3})$

where $f_1 : \mathbb{R}^{d_1} \to \mathbb{R}$ and $f_2 : \mathbb{R}^{d_2} \to \mathbb{R}^{d_1}$ are continuously differentiable functions. We have easily accessible unbiased estimates $f_2(x;\zeta)$, $\nabla f_2(x;\zeta)$ and $\nabla f_1(y;\xi)$. However, $\nabla f_2(x;\zeta^1)^\top \nabla f_1(f_2(x;\zeta^2);\xi)$ is only a biased estimate of the true gradient $\nabla F(x) = \nabla F_2(x)^\top \nabla F_1(F_2(x))$, where ζ^1 and ζ^2 are drawn independently. The difficulty comes from the fact that we have no access to $F_2(x)$ and need to estimate it through $\nabla f_2(x;\zeta)$.

Stochastic compositional optimization (3.18) has many applications in risk management [74], policy evaluation in reinforcement learning [75] and model-agnostic meta-learning [41]. Wang et al. [19] gave the first non-asymptotic analysis of stochastic compositional optimization and proposed a two-timescale stochastic compositional gradient descent (SCGD) algorithm. They further provided an accelerated version of SCGD and improved the rate to $\mathcal{O}(\varepsilon^{-4.5})$ [76]. Ghadimi et al. [77] developed a single-timescale approach and Chen et al. [35] added some corrections term to SCGD. They both achieved the optimal $\mathcal{O}(\varepsilon^{-4})$ complexity. When assuming average smoothness and using variance reduction, Zhang and Xiao [30] achieved $\mathcal{O}(\varepsilon^{-3})$ sample complexity for a special case when $F_1(x)$ is deterministic. They improved their work to general multi-level stochastic case in Zhang and Xiao [36] and proved the same sample complexity. Hu et al. [78] used SARAH [9] to estimate all the stochastic parts and also obtained $\mathcal{O}(\varepsilon^{-3})$ complexity. Table 3.3 provides a summary of all related algorithms.

The stochastic compositional optimization in (3.18) can be reformulated as a special case of bilevel optimization [48]:

$$\min_{x\in\mathbb{R}^{d_2}} F(x) := \mathbb{E}_\xi[f_1(y^*(x);\xi)] \quad \text{for} \quad y^*(x) = \arg\min_{y\in\mathbb{R}^{d_1}} \|y - f_2(x;\zeta)\|^2.$$

Motivated by Algorithm 2 for solving bilevel optimization (3.6), we propose Algorithm 4 for stochastic compositional optimization. The algorithm uses the general framework in Algorithm 1 to estimate $F_2(x_t)$ using y_t through access to $f_2(x_t;\zeta_t)$, and then constructs the biased gradient estimator h_t for $\nabla F(x_t)$ through queries for $\nabla f_2(x_t;\tilde{\zeta}_t)^\top \nabla f_1(y_t;\xi_t)$. Note that we use independent samples ζ_t and $\tilde{\zeta}_t$ to ensure that

$$\mathbb{E}_{\tilde{\zeta}_t,\xi_t}[\nabla f_2(x_t;\tilde{\zeta}_t)^\top \nabla f_1(y_t;\xi_t)] = \nabla F_2(x_t)^\top \nabla F_1(y_t),$$

since y_t is independent from $\tilde{\zeta}_t$. Therefore, h_t is actually estimating $\nabla F_2(x_t)^\top \nabla F_1(y_t)$, creating some bias depending on $\|y_t - F_2(x_t)\|$. The update step is still $x_{t+1} = x_t - \alpha h_t$.

We make the following assumptions that are common in the related literature. Note that it is slightly stronger than the average smoothness condition in Assumption 3.1.

Algorithm 4 Variance Reduction for Stochastic Compositional Optimization

Input: $T, Q, D, S, x_0, \alpha, \eta$.
  for $t = 0, 1, \cdots, T-1$ **do**
    if $t \equiv 0 \pmod{Q}$ **then**
      Sample $D_t^y = \{\zeta_t^1, \zeta_t^2, \cdots, \zeta_t^D\}$ and compute $y_t = \frac{1}{D}\sum_{i=1}^D f_2(x_t; \zeta_t^i)$.
      Sample $D_t^h = \{(\tilde{\zeta}_t^1, \xi_t^1), (\tilde{\zeta}_t^2, \xi_t^2), \cdots, (\tilde{\zeta}_t^D, \xi_t^D)\}$.
      $h_t = \frac{1}{D}\sum_{i=1}^D \nabla f_2(x_t; \tilde{\zeta}_t^i)^\top \nabla f_1(y_t; \xi_t^i)$.
    else
      Sample $S_t^y = \{\zeta_t^1, \zeta_t^2, \cdots, \zeta_t^S\}$.
      $y_t = (1-\eta)\Big(y_{t-1} - \frac{1}{S}\sum_{i=1}^S f_2(x_{t-1}; \zeta_t^i)\Big) + \frac{1}{S}\sum_{i=1}^S f_2(x_t; \zeta_t^i)$.
      Sample $S_t^h = \{(\tilde{\zeta}_t^1, \xi_t^1), (\tilde{\zeta}_t^2, \xi_t^2), \cdots, (\tilde{\zeta}_t^S, \xi_t^S)\}$.
      $h_t = (1-\eta)\Big(h_{t-1} - \frac{1}{S}\sum_{i=1}^S \nabla f_2(x_{t-1}; \tilde{\zeta}_t^i)^\top \nabla f_1(y_{t-1}; \xi_t)\Big) + \frac{1}{S}\sum_{i=1}^S \nabla f_2(x_t; \tilde{\zeta}_t^i)^\top \nabla f_1(y_t; \xi_t)$.
    end if
    Update $x_{t+1} = x_t - \alpha h_t$.
  end for
Output: x_τ with τ chosen uniformly at random from $\{0, 1, \cdots, T-1\}$.

Assumption 3.21 *For any ξ and ζ, the function $f_1(y;\xi)$ and $f_2(x;\zeta)$ satisfy that*

- *$f_1(y;\xi)$ is M_{f_1}-Lipschitz continuous and L_{f_1}-smooth.*
- *$f_2(x;\zeta)$ is M_{f_2}-Lipschitz continuous and the Jacobian $\nabla f_2(x;\zeta)$ is L_{f_2}-Lipschitz continuous.*
- *The estimate $\nabla f_1(y;\xi)$ has bounded variance, i.e. $\mathbb{E}_\xi\|\nabla f_1(y;\xi) - \nabla F_1(y)\|^2 \le \sigma_{f_1}^2$.*
- *The bounded variance condition for $f_2(x;\zeta)$ also holds, i.e. $\mathbb{E}_\zeta\|f_2(x;\zeta) - F_2(x)\|^2 \le \sigma_{f_2}^2$ and $\mathbb{E}_\zeta\|\nabla f_2(x;\zeta) - \nabla F_2(x)\|^2 \le \sigma_{f_2'}^2$.*

With Assumption 3.21, it is easy to verify that $F(x)$ is L_F-smooth [30, Appendix A] with $L_F = M_{f_2}^2 L_{f_1} + M_{f_1} L_{f_2}$. Moreover, the gradient error satisfies

$$\begin{aligned}
\mathbb{E}\|h_t - \nabla F(x_t)\|^2 &= \mathbb{E}\|h_t - \nabla F_2(x_t)^\top \nabla F_1(y_t) + \nabla F_2(x_t)^\top \nabla F_1(y_t) - \nabla F_2(x_t)^\top \nabla F_1(F_2(x_t))\|^2 \\
&\le 2\mathbb{E}\|h_t - \nabla F_2(x_t)^\top \nabla F_1(y_t)\|^2 + 2\mathbb{E}\|\nabla F_2(x_t)^\top [\nabla F_1(y_t) - \nabla F_1(F_2(x_t))]\|^2 \\
&\le 2\underbrace{\mathbb{E}\|h_t - \nabla F_2(x_t)^\top \nabla F_1(y_t)\|^2}_{A_t} + 2M_{f_2}^2 L_{f_1}^2 \underbrace{\mathbb{E}\|y_t - F_2(x_t)\|^2}_{B_t}. \qquad (3.19)
\end{aligned}$$

The last inequality holds since $F_2(x)$ is M_{f_2}-Lipschitz continuous and $F_1(y)$ is L_{f_1}-smooth by Assumption 3.21. The first term A_t is the estimation error of h_t and the second bias term B_t is the estimation error of y_t. We then show that both terms can satisfy equation (2.7) with parameters selected according to Theorem 2.7.

Lemma 3.22 *Supposing Assumption 3.21 holds, with all the settings in Algorithm 4 for parameters D, S, Q, η and α selected according to Theorem 2.7 for some constant $\rho_A > 0$ we define later, i.e.,*

(i) $\eta = 0$, $D = \alpha T$, $S = \rho_A^{-1/2}\ell_f \alpha^{3/2} T^{1/2}$ and $Q = \rho_A^{1/2}(2\ell_f)^{-1}\alpha^{-1/2}T^{1/2}$ for any α such that $(\rho_A \ell_f^{-2})^{1/3} T^{-1/3} \le \alpha \le 1/(3L_F)$ with the smooth parameter $L_F = M_{f_2}^2 L_{f_1} + M_{f_1} L_{f_2}$;

(ii) $\eta = 2\ell_f^2\alpha^2/\rho_A$, $\alpha = \rho_A^{1/3}(8\ell_f^2 \cdot T)^{-1/3}$, $D = \rho_A \ell_f^{-2}\alpha^{-1}$, $S = 1$ and $Q = T$,

then we have

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \le \begin{cases} \sigma^2/(\alpha T) + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2, & \text{if (i)}\ ; \\ (\sigma^2 + 12M_{f_2}^2 L_{f_1}^2 \sigma_{f_2}^2)/(\alpha T) + 2\rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2, & \text{if (ii)}\ , \end{cases}$$

$$\frac{1}{T}\sum_{t=0}^{T-1} B_t \le \frac{\sigma_{f_2}^2}{\alpha T} + \rho_A \frac{M_{f_2}^2}{\ell_f^2} \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2,$$

where $\sigma^2 := 2M_{f_2}^2\sigma_{f_1}^2 + 2M_{f_1}^2\sigma_{f_2'}^2$ *and* $\ell_f^2 := 4M_{f_2}^4 L_{f_1}^2 + 2M_{f_1}^2 L_{f_2}^2$ *with all constants defined in Assumption 3.21.*

Similar to the analysis before, with Lemma 3.22 to bound both the estimation error and the bias, we can obtain the following convergence theorem. The proofs of Lemma 3.22 and Theorem 3.23 are given in Appendix A.2.4.

Theorem 3.23 *For smooth non-convex stochastic compositional optimization defined in* (3.18)*, under Assumption 3.21, with the parameters* D, S, Q, η *and* α *selected according to Lemma 3.22 for* $\rho_A = 1/12$*, we have that the output of Algorithm 4 satisfies*

$$\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \le \frac{2[F(x_0) - F^*] + 2\sigma^2 + 26M_{f_2}^2 L_{f_1}^2 \sigma_{f_2}^2}{\alpha T},$$

where $\sigma^2 := 2M_{f_2}^2\sigma_{f_1}^2 + 2M_{f_1}^2\sigma_{f_2'}^2$*. To make sure that* x_τ *is an* ε*-stationary point, similar to the analysis in Remark 2.9, one can show that the sample complexity is* $\mathcal{O}(\varepsilon^{-3})$*, which is optimal.*

In comparison, Wang et al. [76] and Ghadimi et al. [77] considered moving average estimators of $F_2(x)$ and $\nabla F_2(x)$ in the form of $h_t = (1-\eta)h_{t-1} + z_t$, where z_t are unbiased estimates returned by the first-order stochastic oracle. Without variance reduction, they achieved $\mathcal{O}(\varepsilon^{-4})$ sample complexity to find an ε-stationary point defined in Definition 2.3. When assuming average smoothness condition (3.2) holds and applying variance reduction, our result matches the optimal complexity obtained by Zhang and Xiao [36] and Hu et al. [78], which can be viewed as a special case of our Algorithm 4 with $\eta = 0$. Moreover, when setting $\eta > 0$, our algorithm is the first single-loop algorithm to achieve $\mathcal{O}(\varepsilon^{-3})$ sample complexity without using large batch sizes. This allows simpler implementation and reduces the cost for practical applications.

Concluding Remarks In this chapter, we extend the general analysis in Chapter 2 from unbiased smooth non-convex stochastic optimization to the biased case. As long as the biased stochastic oracle satisfies the average smoothness condition (3.2) in Assumption 3.1, we can still apply variance reduction methods, and then Lemma 2.6 and Theorem 2.7 hold. In Section 3.1, we first consider the general biased optimization and prove that if the bias term satisfies the SBG or ABG condition, Algorithm 1 still achieves the optimal $\mathcal{O}(\varepsilon^{-3})$ complexity. We further apply the general framework to three concrete examples, i.e. stochastic bilevel optimization in Section 3.2, stochastic minimax optimization in Section 3.3 and stochastic compositional optimization in Section 3.4. We explain why we can only construct biased gradient oracles and provide general frameworks for solving these structured optimization problems using variance reduction. By properly analyzing the specific bias terms and showing that the gradient error can be bounded similar to the ABG condition, we prove that the sample complexity of our proposed algorithms is $\mathcal{O}(\varepsilon^{-3})$. In the previous chapter and this chapter, we consider the smooth non-convex stochastic optimization and prove the optimal complexity for our general algorithms. In the next chapter, we will apply the general framework for variance reduction in Algorithm 1 to non-smooth non-convex stochastic optimization. When combined with stochastic mirror descent, Algorithm 1 also solves this class of non-convex stochastic optimization with $\mathcal{O}(\varepsilon^{-3})$ sample complexity.

Chapter 4

Variance Reduction for Stochastic Mirror Descent

Overview We have seen the effectiveness of our general framework and corresponding analysis for variance reduction methods in previous chapters. When applied to smooth non-convex stochastic optimization, we are able to show the $\mathcal{O}(\varepsilon^{-3})$ sample complexity. In this chapter, we consider a more general problem where the objective function can be non-smooth as well. Although we assume that the non-smooth part is relatively simple, such optimization problem has many practical applications. We combine the general framework in Algorithm 1 with stochastic mirror descent and prove that the sample complexity is still $\mathcal{O}(\varepsilon^{-3})$ to achieve an ε-stationary point for non-smooth non-convex stochastic optimization.

4.1 Problem Setting

In this chapter, we study the following non-smooth non-convex stochastic optimization problem:

$$\min_{x\in\mathcal{X}\subseteq\mathbb{R}^d} \Phi(x) := F(x) + r(x) = \mathbb{E}_\xi[f(x;\xi)] + r(x), \tag{4.1}$$

where $F(x) := \mathbb{E}_\xi[f(x;\xi)]$ is smooth but non-convex, and $r(x)$ is non-smooth but convex. When $r(x) = 0$, the problem reduces to smooth non-convex stochastic optimization that we have discussed in the previous chapters. When $r(x) \neq 0$ and is relatively simple, for example L_1-regularizer $r(x) = \lambda\|x\|_1$ and indicator function

$$r(x) = \begin{cases} 0, & \text{if } x \in \mathbb{C}, \\ +\infty, & \text{otherwise,} \end{cases}$$

for some convex domain $\mathbb{C}$, the problem becomes more general and covers a wide spectrum of optimization problems arising in machine learning applications such as LASSO [23], sparse logistic regression [79] and 1-norm SVM [24]. Actually, when adding non-smooth convex regularization (e.g. L_1) to smooth non-convex objectives or solving smooth non-convex optimization constrained on a convex domain (e.g. probability simplex), the problems can be formulated as (4.1).

For smooth optimization problems, we can directly compute the gradients and perform (stochastic) gradient descent. When it comes to the non-smooth case, lots of algorithms based on mirror descent [25] have been proposed for solving (4.1), and nearly all of them can be generalized as the following iterative updates:

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \left\{ r(x) + h_t^\top x + \frac{1}{\alpha} V_\omega(x, x_t) \right\}, \tag{4.2}$$

where α is the step-size, h_t is some gradient estimator of $\nabla F(x_t)$ and Bregman distance $V_\omega(x, y)$ is defined in Definition 4.1.

Definition 4.1 (Bregman Distance) *For some continuously differentiable and μ-strongly convex function $\omega(x) : \mathcal{X} \to \mathbb{R}$, we define the corresponding Bregman distance as*

$$V_\omega(x, y) = \omega(x) - \omega(y) - \nabla\omega(y)^\top (x - y).$$

The strongly-convex function $\omega(x)$ is often called the distance generating function.

Remark 4.2 (Examples of Bregman Distance) *If we choose the generating function $\omega(x) = \frac{1}{2}\|x\|^2$, then its Bregman distance is $V_\omega(x, y) = \frac{1}{2}\|x - y\|^2$, and thus the update step (4.2) reduces to (projected) subgradient descent. In the simplest case when $\mathcal{X} = \mathbb{R}^d$ and $r(x) = 0$, it is just (stochastic) gradient descent $x_{t+1} = x_t - \alpha h_t$. We can also choose other distance generating functions. For example, if $\omega(x) = \sum_{i=1}^d x_i \ln(x_i)$, the Bregman distance $V_\omega(x, y) = \sum_{i=1}^d x_i \ln(x_i/y_i)$ recovers the well-known Kullback-Leibler divergence which measures the difference between two distributions.*

The stochastic mirror descent step (4.2) involves the computation of another simple optimization problem at each iteration. In this chapter, we assume that the step is easily solvable since $r(x)$ is relatively simple. The update step is also referred to as proximal operator in the related literature, and that is why the related algorithms are often called proximal algorithms.

Ghadimi et al. [37] gave the first non-asymptotic convergence analysis for solving (4.1). When using h_t as the true gradient $\nabla F(x_t)$ or its mini-batch estimator in the update (4.2), they proposed ProxGD in the deterministic setting and ProxSGD in the stochastic setting, respectively. Reddi et al. [80] proposed ProxSVRG/SAGA for solving (4.1) in the finite-sum case based on the variance reduction methods SVRG [5] and SAGA [6]. The algorithms compute h_t using SVRG/SAGA to estimate $\nabla F(x_t)$ and consider the special case when $\omega(x) = \frac{1}{2}\|x\|^2$. Li and Li [38] improved the analysis and parameter choices of ProxSVRG, and the improved version is named ProxSVRG+. Li et al. [81] further extended ProxSVRG+ to general generating function $\omega(x)$ and proposed SVRAMD. The best sample complexity so far for solving (4.1) is achieved by Prox-SpiderBoost [8], which is the proximal version of SpiderBoost. This motivates us to combine our general framework with stochastic mirror descent and provide a general analysis. Not surprisingly, our algorithm matches the complexity of Prox-SpiderBoost and includes it as a special case when $\omega(x) = \frac{1}{2}\|x\|^2$ and $\eta = 0$.

We measure the performances of all algorithms by both iteration complexity and sample complexity to find an ε-stationary point of (4.1). We are also interested in the iteration complexity since we need to solve the update (4.2) at each iteration, which might be costly in some scenarios. We first define the concept of stationarity for non-smooth objectives considered in this chapter.

The gradient mapping corresponding to the estimator h_t is defined as

$$\mathcal{G}_\alpha(x_t, h_t) := \frac{1}{\alpha}(x_t - x_{t+1}), \tag{4.3}$$

for the update (4.2). $G_\alpha(x_t, h_t)$ will reduce to h_t when $r(x) = 0$, $\omega(x) = \frac{1}{2}\|x\|^2$ and $\mathcal{X} = \mathbb{R}^d$ by Remark 4.2. This is called the gradient mapping since the update is equivalent to $x_{t+1} = x_t - \alpha\mathcal{G}_\alpha(x_t, h_t)$ by the definition above. Similar to the smooth case, we aim to find a point such that the true gradient is small. Naturally, the mapping of the true gradient in the non-smooth setting is

$$\mathcal{G}_\alpha(x_t, \nabla F(x_t)) := \frac{1}{\alpha}(x_t - x_t^+), \tag{4.4}$$

Table 4.1: Comparison of the sample complexity of different algorithms to achieve an ε-stationary point defined in Definition 4.3 for solving non-smooth non-convex stochastic optimization in (4.1). For double-loop algorithms, Batch (D) stands for the batch size used for checkpoint gradients in the out-loop and Batch (S) stands for the batch size used in the inner-loop. For single-loop algorithms, Batch (D) stands for the batch size used for checkpoint gradients at the beginning and Batch (S) stands for the batch size used in other iterations. Iteration means the total number of iterations required and Sample means the total number of samples required. $\tilde{\mathcal{O}}$ hides additional logarithmic terms in ε^{-1}.

Algorithm	Structure	Batch (D)	Batch (S)	Iteration	Sample
ProxSGD [37]	Single-Loop	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-4})$
ProxSVRG+ [38]	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-4/3})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-10/3})$
SVRAMD [81]	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-4/3})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-10/3})$
Prox-SpiderBoost [8]	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-3})$
This work (SPIDER)	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-3})$
This work (SARAH)	Double-Loop	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(1)$	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-3})$
This work (New 2)	Single-Loop	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(1)$	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-3})$
This work (STORM)	Single-Loop	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}(\varepsilon^{-3})$	$\tilde{\mathcal{O}}(\varepsilon^{-3})$

for x_t^+ defined as follows to avoid confusions:

$$x_t^+ = \arg\min_{x\in\mathcal{X}} \left\{ r(x) + \nabla F(x_t)^\top x + \frac{1}{\alpha} V_\omega(x, x_t) \right\}, \tag{4.5}$$

i.e. replacing h_t by the true gradient $\nabla F(x_t)$ in the update (4.2). For simplicity of the notation, we will use $\mathcal{G}_\alpha(x_t)$ instead of $\mathcal{G}_\alpha(x_t, \nabla F(x_t))$ in the following analysis. $\mathcal{G}_\alpha(x_t)$ reduces to $\nabla F(x_t)$ by the similar arguments above.

Definition 4.3 (ε-Stationary Point) *For some accuracy measure $\varepsilon > 0$, a point x is called an ε-stationary point of the non-smooth non-convex function $\Phi(x)$ defined in* (4.1) *if $\|\mathcal{G}_\alpha(x)\| \le \varepsilon$ with the gradient mapping $\mathcal{G}_\alpha(x)$ defined in* (4.4).

Table 4.1 provides a comparison of all aforementioned algorithms to achieve an ε-stationary point for solving (4.1). Under average smoothness assumption, our algorithm improves upon ProxSGD and ProxSVRG+, and achieves the best $\mathcal{O}(\varepsilon^{-3})$ sample complexity. When using the parameter selection rules to recover New 2 or STORM [10] as mentioned in Section 2.4, our algorithm is the first to guarantee convergence without large batch sizes. However, the improvement comes at the cost of increased iteration complexity.

4.2 Algorithm and Convergence Analysis

When using the general variance reduction framework in Algorithm 1 to estimate the gradient, we obtain the variance reduced stochastic mirror descent in Algorithm 5. Before analyzing the convergence rate of Algorithm 5, we first formally state all the necessary assumptions.

Assumption 4.4 (Objective Function) *The function $\Phi(x) = F(x) + r(x)$ defined in* (4.1) *satisfies that*

- *The possibly non-convex function $F(x)$ is L_F-smooth.*
- *The possibly non-smooth function $r(x)$ is convex, that is, $\forall x, y \in \mathcal{X}$, we have that*

$$r(y) \ge r(x) + g_x^\top (y - x),$$

Algorithm 5 Variance Reduced Stochastic Mirror Descent

Input: $T, Q, D, S, x_0, \alpha, \eta, \omega(x)$.
  for $t = 0, 1, \cdots, T-1$ **do**
    if $t \equiv 0 \pmod{Q}$ **then**
      Query $\mathcal{SO}$ for D times and compute $h_t = \frac{1}{D}\sum_{i=1}^{D} \nabla f(x_t; \xi_t^i)$.
    else
      Query the oracle $\mathcal{SO}$ for $2S$ times.
      $h_t = (1-\eta)(h_{t-1} - \frac{1}{S}\sum_{i=1}^{S} \nabla f(x_{t-1}; \xi_t^i)) + \frac{1}{S}\sum_{i=1}^{S} \nabla f(x_t; \xi_t^i)$.
    end if
    $x_{t+1} = \arg\min_{x \in \mathcal{X}} \{r(x) + h_t^\top x + \frac{1}{\alpha} V_\omega(x, x_t)\}$.
  end for
Output: x_τ with τ chosen uniformly at random from $\{0, 1, \cdots, T-1\}$.

for any subgradient $g_x \in \partial\, r(x)$.

In addition, we assume that the optimal value $\Phi^* := \min_{x \in \mathcal{X}} \Phi(x)$ is finite.

In this chapter, we assume an unbiased gradient oracle $\mathcal{SO}$ for the function $F(x)$. Given some query point $x \in \mathcal{X}$, the oracle returns an estimate $\nabla f(x; \xi)$ for the true gradient $\nabla F(x)$ such that $\mathbb{E}_\xi[\nabla f(x; \xi)] = \nabla F(x)$. To apply variance reduction methods, we assume that the oracle also satisfies Assumption 2.2, and we restate it here.

Assumption 4.5 (Gradient Estimate) *The unbiased gradient estimate $\nabla f(x; \xi)$ returned by the stochastic oracle $\mathcal{SO}$ satisfies that*

$$\mathbb{E}_\xi \|\nabla f(x; \xi) - \nabla F(x)\|^2 \le \sigma^2, \tag{4.6}$$

$$\mathbb{E}_\xi \|\nabla f(x_1; \xi) - \nabla f(x_2; \xi)\| \le \ell_f \|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{X}, \tag{4.7}$$

for some constants $\sigma > 0$ and $\ell_f > 0$.

We first give two technical lemmas in Ghadimi et al. [37] to analysis the behaviors of the gradient mapping $\mathcal{G}_\alpha(x_t, h_t)$ and the relation between $\mathcal{G}_\alpha(x_t, h_t)$ and $\mathcal{G}_\alpha(x_t)$. The detailed proofs can be found in Appendix A.3, and the proofs follow from the optimality condition of the update step (4.2).

Lemma 4.6 *[37, Lemma 1] For the mirror descent update* (4.2) *and corresponding gradient mapping $\mathcal{G}_\alpha(x_t, h_t)$ defined in* (4.3)*, if the generating function $\omega(x)$ is μ-strongly convex, we have*

$$h_t^\top \mathcal{G}_\alpha(x_t, h_t) \ge \frac{1}{\alpha}\Big(r(x_{t+1}) - r(x_t)\Big) + \mu \|\mathcal{G}_\alpha(x_t, h_t)\|^2, \tag{4.8}$$

for any stepsize $\alpha > 0$ and $x_t \in \mathcal{X}$.

Lemma 4.7 *[37, Lemma 2] If we define $\mathcal{G}_\alpha(x_t, h_t)$ in* (4.3) *and define $\mathcal{G}_\alpha(x_t)$ in* (4.4)*, assuming that the generating function $\omega(x)$ is μ-strongly convex, then it holds that*

$$\|\mathcal{G}_\alpha(x_t, h_t) - \mathcal{G}_\alpha(x_t)\|^2 \le \frac{1}{\mu^2}\|h_t - \nabla F(x_t)\|^2, \tag{4.9}$$

for any stepsize $\alpha > 0$ and $x_t \in \mathcal{X}$.

We then analyze the estimation error term $A_t = \mathbb{E}\|h_t - \nabla F(x_t)\|^2$. By Lemma 2.5 in Chapter 2, under Assumption 4.5, the estimation error satisfies

$$\begin{aligned} A_{t+1} &\le (1-\eta) A_t + 2\sigma^2 \frac{\eta^2}{S} + \frac{2\ell_f^2}{S} \mathbb{E}\|x_{t+1} - x_t\|^2 \\ &\le (1-\eta) A_t + 2\sigma^2 \frac{\eta^2}{S} + 2\ell_f^2 \frac{\alpha^2}{S} \mathbb{E}\|\mathcal{G}_\alpha(x_t, h_t)\|^2, \end{aligned}$$

by the definition of $\mathcal{G}_\alpha(x_t, h_t)$ in (4.3). Then by Lemma 2.6 and Theorem 2.7, we obtain that the average estimation error

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \leq \frac{\sigma^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\mathcal{G}_\alpha(x_t, h_t)\|^2,$$

with parameter choices specified in Theorem 2.7. Therefore, with the help of Lemma 4.6 and 4.7, we can show the following convergence theorem for Algorithm 5. The complete proof of Theorem 4.8 is provided in Appendix A.3.

Theorem 4.8 *Suppose Assumption 4.4 and 4.5 hold. Let the selections of parameters be specified as in Theorem 2.7 with the choice that* $\rho_A = \frac{\mu^2}{3}$ *and one additional requirement that* $\alpha \leq \frac{\mu}{3L_F}$. *Assuming that the generating function* $\omega(x)$ *is* μ*-strongly convex, then the output* x_τ *of Algorithm 5 satisfies*

$$\mathbb{E}_\tau\|\mathcal{G}_\alpha(x_\tau)\|^2 \leq \frac{16[\Phi(x_0) - \Phi^*]/\mu + 10\,\sigma^2/\mu^2}{\alpha T}.$$

To guarantee an ε*-stationary point as defined in Definition 4.3, by Remark 2.9, the sample complexity is* $\mathcal{O}(\varepsilon^{-3})$.

The sample complexity for Algorithm 5 to achieve an ε-stationary point defined in Definition 4.3 is $\mathcal{O}(\varepsilon^{-3})$, improving on the $\mathcal{O}(\varepsilon^{-4})$ complexity of ProxSGD [37] without variance reduction and the $\mathcal{O}(\varepsilon^{-10/3})$ complexity of ProxSVRG+ [38]. Prox-SpiderBoost [8] also achieved $\mathcal{O}(\varepsilon^{-3})$ sample complexity, but it is a special case of Algorithm 5. The iteration complexity $\mathcal{O}(T)$, i.e. the number of times to compute the update (4.2), varies with different parameter selections of our algorithm. When setting $\eta = 0$ and $\alpha = \mathcal{O}(1)$ to recover Prox-SpiderBoost, we are able to achieve $\mathcal{O}(\varepsilon^{-2})$ iteration complexity, also matching the best known results. More interestingly, when setting $\eta \neq 0$ to recover New 2 or STORM [10], our algorithm is the first to guarantee convergence without using large batch sizes and also achieves $\mathcal{O}(\varepsilon^{-3})$ sample complexity. However, the iteration complexity when $\eta \neq 0$ is also $\mathcal{O}(\varepsilon^{-3})$, which is worse than other algorithms.

Concluding Remarks In this chapter, we consider the non-smooth non-convex stochastic optimization problem defined in (4.1). When combining the general framework in Algorithm 1 with stochastic mirror descent, we propose Algorithm 5 and show that the sample complexity is still $\mathcal{O}(\varepsilon^{-3})$ to guarantee an ε-stationary point as defined in Definition 4.3. The convergence analysis for Algorithm 5 is greatly simplified with results in Chapter 2. This further shows the effectiveness and convenience of our general analysis for variance reduction methods.

Chapter 5

Experiments

In this chapter, we present some numerical experiments to compare different variance reduction methods designed for non-convex stochastic optimization. Although these variance reduction methods are very powerful in theoretical analysis, there has been a lack of empirical evidence to verify their effectiveness in practical applications. For example, Defazio and Bottou [82] studied the behaviors of SVRG [5] for its application in convolutional neural networks. They found that SVRG fails to reduce the variance and fails to improve upon the performance of SGD, especially for large models. The work mainly studied SVRG and its variants such as SAGA [6] and SCSG [83]. Since such variance reduction methods are mainly designed for convex and strongly-convex objectives, it remains unknown whether the same holds true for non-convex variance reduction methods such as SPIDER [7] and STORM [10] measured by ε-stationarity.

Cutkosky and Orabona [10] thought the failure of SVRG and SPIDER in non-convex optimization applications stems from the use of non-adaptive stepsizes and large batch sizes, and that's why they proposed STORM. They compared the performance of STORM to the widely-used AdaGrad [84] and Adam [85] on an image recognition task using a ResNet [1] model, and showed that STORM is marginally better. PAGE [26] also included empirical comparison on the image recognition task for different deep learning architectures. The experiments show that PAGE is marginally better than SGD. These pioneer works for the application of variance reduction in practical non-convex problems might suggest that single-loop algorithms or algorithms without the use of large batch sizes should be our choice.

As a benefit of our general framework, all variance reduction methods were implemented through Algorithm 1 and recovered according to different parameter choices discussed in Section 2.4. We mainly compared the performances of SpiderBoost [8], SARAH [9], STORM [10] and New 2 on three minimization problems. The parameter selections for individual method follow from Table 2.1. We provide the configuration used in our experiments in Table 5.1 to give an example on how to select them in practice.

We mainly tuned the stepsize α and the momentum parameter η (for New 2 and STORM) in specific applications, and other parameters were fixed to the values in Table 5.1. For STORM, we let $\alpha_t = \frac{k}{(\omega+t)^{1/3}}$ and $\eta_t = c\alpha_t^2$ according to the original paper [10], and tuned k, ω and c. Table 5.1 is just one example, and other configurations are also possible. All parameters are selected such that each algorithm is clearly separated from the others by their characterizations listed in Table 2.1. For example, SPIDER requires large batch sizes for both D and S with $D > S$; SARAH allows small batch size S but need larger Q. Note that in theoretical analysis, we let $S = 1$ or $D = 1$ to avoid mini-batches. In our experiments, we instead use small batch size $S = 32$, which is a common choice in

Table 5.1: Parameter settings for Algorithm 1 in our experiments to recover the variance reduction methods according to Table 2.1. SPIDER refers to its improved version SpiderBoost. Since the general framework recovers SGD by setting $\eta = 1$ and $D = S$, we also include it here. PAGE [26] is the loopless version of SPIDER/SARAH. For single-loop algorithms SGD, New 2 and STORM, Q is set to be T. The momentum parameter η for New 2 and STORM, as well as the stepsize α for all algorithms are tuned for different problems.

Parameters	SGD	SPIDER	SARAH	PAGE	New 2	STORM
η	1	0	0	0	-	-
D	32	1024	1024	1024	256	32
S	32	256	32	32	32	32
Q	-	32	256	-	-	-

practical applications. The settings of New 1 is between SPIDER and SARAH, and thus generating similar performance, so we leave it out.

Toy Example We first consider a toy example:

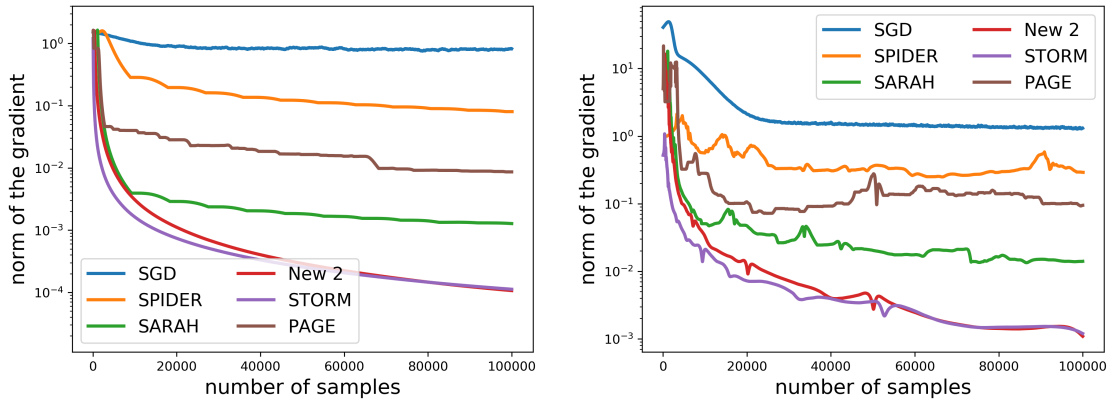
$$\min_{x\in\mathbb{R}^d} f(x) = \ln\left(\frac{1}{2}\|Ax-b\|^2+1\right), \tag{5.1}$$

where $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$. The objective above is non-convex and smooth with one global minimum when $\|Ax-b\| = 0$ and several other stationary points when $\|Ax-b\| \to \infty$. It highly relates to linear regression when regarding A as the feature matrix and b as labels. Here we implemented a pure stochastic setting by adding random noise to the true gradient, and an average of all the samples in the batch is returned if involving mini-batches. A and b were generated in the following two ways.

- We let $n = 200$ and $d = 100$. Each entry of A and b was sampled independently from the standard Gaussian distribution $\mathcal{N}(0,1)$. At each query point, a random vector with every entry sampled independently from $\mathcal{N}(0,10)$ was added to the true gradient, leading to an unbiased gradient estimate.
- We used the well-known Boston house prices dataset [86] for the feature matrix A and label b. The number of instances is 506. For each instance, there are 13 attributes representing basic information of the house, i.e. $n = 506$ and $d = 13$. The goal is to predict the price. Similar to the above case, we added a random vector with every entry sampled independently from $\mathcal{N}(0,1)$ to the true gradient.

For both cases, we swept α over the same logarithmically spaced grid for all algorithms except STORM. The best choice of SGD is $\alpha = 0.001$ for randomly generated data and $\alpha = 10^{-5}$ for Boston house prices dataset. The best choice of other algorithms is $\alpha = 0.01$ for both types of data. This is aligned with our theoretical analysis, since the stepsize for SGD should be $\mathcal{O}(1/\sqrt{T})$, which is of order $\mathcal{O}(\varepsilon^2)$ to guarantee an ε-stationary point and is smaller comparing to variance reduction methods as listed in Table 2.1. For New2, $\eta = 10^{-4}$ generates the best performance, and it is of order $\mathcal{O}(\alpha^2)$. For STORM, the best setting is $k = \omega = 0.1$ and $c = 0.01$. We measure the performance of each algorithm by Euclidean norm of the true gradient and plot it against the number of samples used in Figure 5.1. The performance of every method is averaged over 10 different runs.

Figure 5.1 suggests that variance reduction methods with momentum parameter $\eta \neq 0$, i.e. New 2 and STORM, are much better than others. They converge faster to some stationary point with the smallest gradient norm. SARAH and PAGE are better than SPIDER, and all variance reduction methods are better than SGD. It is worth mentioning



(a) Simulated Data

(b) Boston House Prices

Figure 5.1: The performances of different algorithms to solve (5.1). Logarithmic scale is adopted for the norm of the true gradients. (a) is the case when A and b are randomly sampled, while in (b) the real-world Boston house prices dataset are used.

that SGD actually converges to the global optimum and variance reduction methods converge to other stationary points. This coincides with the intuition that the noise in SGD helps to escape local optima.

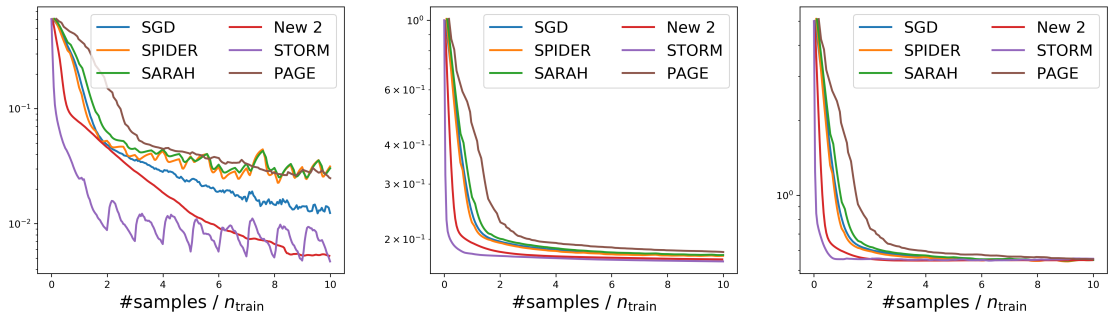
Robust Linear Regression Then we extend the toy example (5.1) to one practical variant, i.e. robust linear regression considered in SpiderBoost [8]. Given a dataset $\{a_i, b_i\}_{i=1}^n$ with vectors $a_i \in \mathbb{R}^d$ representing features and $b_i \in \mathbb{R}$ representing the target labels, the goal is to find some vector $x \in \mathbb{R}^d$ such that $a_i^T x$ is close to b_i for each instance i. We want to minimize the following objective function:

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} \ln\left(\frac{1}{2}(a_i^\top x - b_i)^2 + 1\right). \tag{5.2}$$

The objective above is more robust to outliers comparing to the general linear regression. Equation (5.2) is of finite-sum form and is slightly different from (5.1). The gradient estimate in this case is computed at one randomly sampled instance i, and is unbiased when taking expectations. When involving mini-batches, an average of all instances in the batch is simply returned.

We tested all algorithms on the California housing dataset [87] with information of the houses as feature vectors and prices as targets. The number of instances is $n = 20640$ and the number of attributes for each instance is $d = 8$. The dataset was splitted to a training set with size 15480 and a test set with size 5160. We optimized the objective on the training set and then tested the performance on the test set. The same as before, we swept α over logarithmically spaced grid and tuned η for New 2, and k, ω, c for STORM. For SPIDER, the best stepsize is $\alpha = 0.1$. For other algorithms, the best stepsize is $\alpha = 0.01$. This verifies the analysis in Chapter 2 that SPIDER allows larger stepsize. We let $\eta = 0.001$ for New 2 and $k = \omega = 0.1$ and $c = 1$ for STORM.

We measured the performances of all algorithms by three metrics plotted in Figure 5.2. The training loss and training gradient norm show the optimization ability of each methods, and the test mean squared error (MSE) $\frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i \in \mathcal{D}_{\text{test}}} (a_i^\top x - b_i)^2$ if we denote the test set as $\mathcal{D}_{\text{test}}$ measures the generalization ability. The performance of every method is averaged over 10 different runs. Figure 5.2 shows that New 2 and STORM have the best performances, and SGD is better than the remaining methods. SPIDER converges



(a) Training Gradient Norm (b) Training Loss (c) Test Mean Squared Error

Figure 5.2: Behaviors of different algorithms for solving (5.2) on California housing dataset. All metrics are plotted against number of samples with $n_{\text{train}} = 15480$ as the training size, and logarithmic scale is used. (a) shows the Euclidean norm of the true gradient during training, (b) plots the training objective values and (c) compares the mean squared error on the test dataset.

slightly faster than SGD due to the larger stepsize, but the final training loss and test MSE are slightly worse. We also find that all methods converge to similar levels of training loss and test MSE, although the convergence speed differs a lot. There is some small oscillation for the training gradient norm of STORM, and it is periodic between each pass of the training set. This happens because we reset the stepsize to be $k/\omega^{1/3}$, i.e. $t = 0$, at the beginning of each pass, and we found this actually generated better performance.

Multi-Class Classification The last example is a multi-class classification task on the commonly-used MNIST dataset [88]. The dataset contains 60000 images of hand-written digits, each with size 28×28. The task is to classify every image to the correct number it represents. We used 50000 data for training and the rest 10000 for testing, and each image was flattened to a feature vector of size 784.

We first compared the performances of each algorithm using multi-class logistic regression as follows.

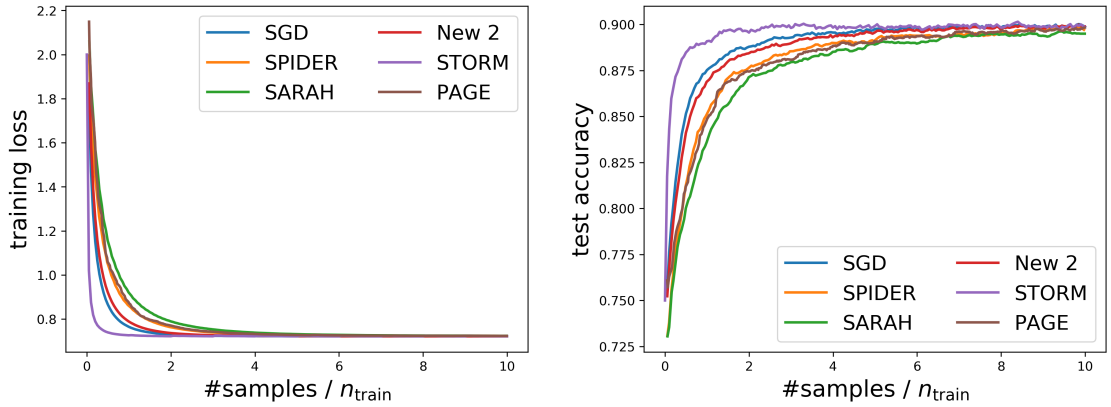
$$\min_{x \in \mathbb{R}^{784 \times 10}} f(x) = \frac{1}{n} \sum_{i=1}^{n} \ell(x; a_i, b_i) + \lambda \sum_{i=1}^{784} \sum_{j=0}^{9} \frac{x_{ij}^2}{1 + x_{ij}^2},$$

where $\{a_i, b_i\}_{i=1}^n$ is the dataset with $a_i \in \mathbb{R}^{784}$ representing the flattened image and $b_i \in \{0, 1, \cdots, 9\}$ representing the digit, and $\ell(x; a_i, b_i)$ is the cross-entropy loss given by

$$\ell(x; a_i, b_i) = -\ln\left(\frac{\exp(a_i^\top x_{b_i})}{\sum_{j=0}^{9} \exp(a_i^\top x_j)}\right).$$

Besides the original cross-entropy loss, we added some non-convex regularization with $\lambda = 0.1$ to make the objective non-convex [8].

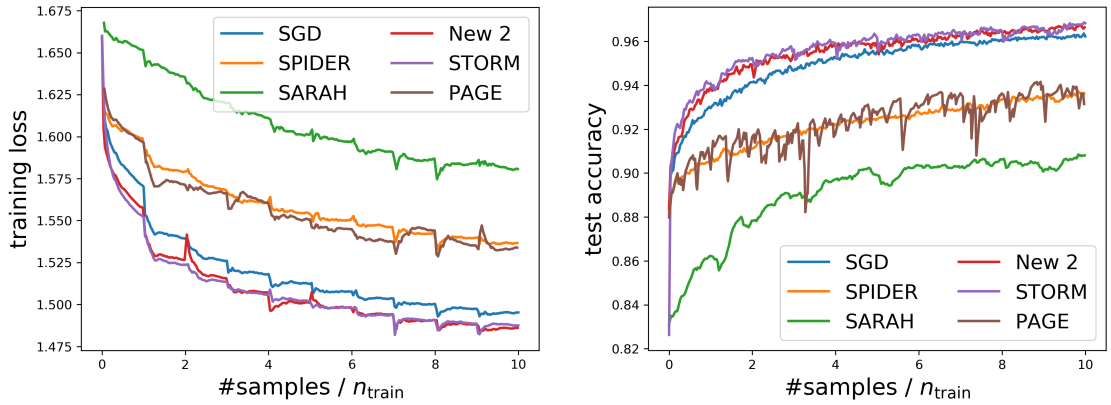
We tuned all parameters in the same way as before. The best choice is $\alpha = 0.001$ for SGD, New 2 and PAGE, $\alpha = 0.005$ for SPIDER and $\alpha = 5 \times 10^{-4}$ for SARAH. We see again that the stepsize for SPIDER is larger, and we also notice that the stepsize for SARAH is smaller, as suggested by our theoretical analysis. Other parameters are $\eta = 0.1$ for New 2, and $k = 0.01$, $\omega = 10$ and $c = 10^4$ for STORM. Different from previous cases, η is large for both New 2 and STORM. Note that $c = 10^4$ is valid when iteration T is large, and this leads to the same level of η as New 2 in most iterations. We measured the performances of all methods by training loss and test accuracy as a convention. The results are shown in Figure 5.3.



(a) Training Loss (b) Test Accuracy

Figure 5.3: The performances of different algorithms for the multi-class classification task on the MNIST dataset using logistic regression. Both metrics are plotted against number of samples used during training with $n_{\text{train}} = 50000$ as the training size.

Figure 5.3 shows that STORM performs the best. It converges the fastest and the final test accuracy is also the highest. New 2 and SGD perform similarly, and are both better than the remaining three methods. However, the final loss and accuracy of all the algorithms do not differ too much. The only concern is the convergence speed. The reason might be that classification task on MNIST is not very challenging, and that the objective function is not too non-convex. Thus variance reduction methods can not take full advantages compared with SGD.



(a) Training Loss (b) Test Accuracy

Figure 5.4: The performances of different algorithms for the multi-class classification task on the MNIST dataset using a three-layer neural network. Both metrics are plotted against number of samples used during training with $n_{\text{train}} = 50000$ as the training size.

We then tested all the algorithms on a three-layer fully-connected neural network. The objective is

$$\min_x f(x) = \frac{1}{n}\sum_{i=1}^{n} \ell(x; a_i, b_i),$$

where x denotes the parameters of the network and $\ell(x; a_i, b_i)$ represents a forward pass given data sample a_i and b_i. The network consisted of two hidden layers with 128 and 64 units respectively. ReLU activation function was used for hidden layers and Softmax was used to compute the outputs. The objective is non-convex, and we implemented it with

PyTorch [89].

The best choice is $\alpha = 0.001$ for SGD, New 2 and PAGE, $\alpha = 10^{-4}$ for SPIDER and $\alpha = 5 \times 10^{-5}$ for SARAH. This again verifies that SARAH requires smaller stepsizes. We chose $\eta = 0.1$ for New 2 and $k = 0.01$, $\omega = 10$ and $c = 10^4$ for STORM. The performance results are given in Figure 5.4. In Figure 5.4, New 2 and STORM still perform better than SGD, and SGD is better than SPIDER, SARAH and PAGE. This again suggests that single-loop algorithms without using large batch sizes are better than others.

Concluding Remarks In this chapter, we provide experiments on three non-convex minimization problems to compare different variance reduction methods. Our experiments suggest that New 2 and STORM perform better than other algorithms, thus introducing the momentum parameter $\eta \neq 0$ indeed helps to improve the practical performances. Note that the new variant New 2 performs similar to the state-of-the-art algorithm STORM, showing the benefits of our general framework. To our best knowledge, this is the first result to empirically compare different variance reduction methods for non-convex optimization. We hope this will provide some useful hints for the application of variance reduction methods in practice.

Chapter 6

Conclusion

The thesis provides a general framework for the analysis of a class of variance reduction methods designed for non-convex stochastic optimization. In the unbiased setting, we observe many potential parameter choices that can achieve the optimal oracle complexity for non-convex stochastic optimization. In the biased setting, we identify the condition on the bias to achieve the optimal complexity. The condition serves as a guideline and a common recipe for white-box biased stochastic oracle constructions of structured non-convex stochastic optimization. Based on that, we obtain a family of single-loop near-optimal algorithms for stochastic bilevel optimization, stochastic minimax optimization and stochastic compositional optimization. We further study the case when the objective function can be non-smooth as well. Our results suggest that these variance reduction methods can be directly extend to non-smooth case via stochastic mirror descent, and that the optimal complexity results also hold.

As for future studies, we provide three potential directions.

- It remains interesting to apply the general framework to more applications where the oracle is biased or the objective function is non-smooth. It is also interesting to find other variants of the original non-convex stochastic optimization problems where our general framework is still helpful.
- Our analysis focus on the dependence on ε and ignores most constants e.g. smoothness parameter L_F. In stochastic bilevel optimization and stochastic minimax optimization, the dependence on the condition number κ is also important. It would be interesting to see whether our framework can be combined with accelerating techniques to obtain better dependence on κ.
- Since there has been very few empirical evidence to show the improved complexity of variance reduction methods compared to SGD, it would be interesting to conduct more experiments for different large-scale dataset and different models, especially in deep learning. The experiments on structured optimization problems such as bilevel optimization and minimax optimization will also be valuable, as they can be direct applied to several recent machine learning problems, e.g. meta-learning, hyper-parameter optimization and adversarial learning.

Appendix A

Deferred Proofs

A.1 Proofs of Results in Chapter 2

We give proofs of Lemma 2.5 and Lemma 2.6 in this section. The proof of Lemma 2.6 is divided into two parts, i.e., Lemma A.1 for the case when $\eta = 0$, and Lemma A.2 for the case when $\eta \neq 0$.

We first give proof of Lemma 2.5 for completeness.

Proof (Lemma 2.5) Subtracting $\nabla F(x_t)$ from both sides of the recursive update (2.4), we decompose the error as the following.

$$\begin{aligned}
h_t - \nabla F(x_t) &= (1-\eta)\left(h_{t-1} - \frac{1}{S}\sum_{i=1}^{S}\nabla f(x_{t-1};\xi_t^i)\right) + \frac{1}{S}\sum_{i=1}^{S}\nabla f(x_t;\xi_t^i) - \nabla F(x_t) \\
&= (1-\eta)(h_{t-1} - \nabla F(x_{t-1})) + \frac{1}{S}\sum_{i=1}^{S}\nabla f(x_t;\xi_t^i) - \nabla F(x_t) \\
&\quad + (1-\eta)\left(\nabla F(x_{t-1}) - \frac{1}{S}\sum_{i=1}^{S}\nabla f(x_{t-1};\xi_t^i)\right) \\
&= (1-\eta)(h_{t-1} - \nabla F(x_{t-1})) + \eta\left(\frac{1}{S}\sum_{i=1}^{S}\nabla f(x_t;\xi_t^i) - \nabla F(x_t)\right) \\
&\quad + (1-\eta)\left(\frac{1}{S}\sum_{i=1}^{S}\delta(\xi_t^i) - \mathbb{E}_{\xi_t}[\delta(\xi_t)]\right), \qquad \text{(A.1)}
\end{aligned}$$

where we denote $\delta(\xi_t) \coloneqq \nabla f(x_t;\xi_t) - \nabla f(x_{t-1};\xi_t)$. For notation simplicity, we let

$$\begin{aligned}
\Delta_1 &= \frac{1}{S}\sum_{i=1}^{S}\nabla f(x_t;\xi_t^i) - \nabla F(x_t), \\
\Delta_2 &= \frac{1}{S}\sum_{i=1}^{S}\delta(\xi_t^i) - \mathbb{E}_{\xi_t}[\delta(\xi_t)].
\end{aligned}$$

Note that when taking expectation w.r.t. $S_t \coloneqq \{\xi_t^1, \cdots, \xi_t^S\}$, the last two terms of (A.1) become 0 and thus

$$\begin{aligned}
A_t &= \mathbb{E}\left\|(1-\eta)(h_{t-1} - \nabla F(x_{t-1})) + \eta\Delta_1 + (1-\eta)\Delta_2\right\|^2 \\
&\overset{(a)}{=} (1-\eta)^2 A_{t-1} + \mathbb{E}\|\eta\Delta_1 + (1-\eta)\Delta_2\|^2 \\
&\overset{(b)}{\leq} (1-\eta)A_{t-1} + 2\eta^2\mathbb{E}\|\Delta_1\|^2 + 2\mathbb{E}\|\Delta_2\|^2, \qquad \text{(A.2)}
\end{aligned}$$

where (a) uses the fact that the cross terms

$$\mathbb{E}_{S_t}[\Delta_1^\top(h_{t-1}-\nabla F(x_{t-1}))]=\mathbb{E}_{S_t}[\Delta_2^\top(h_{t-1}-\nabla F(x_{t-1})]=0,$$

since h_{t-1} and x_{t-1} do not depend on S_t, and (b) uses the fact that $\|a+b\|^2\le 2\|a\|^2+2\|b\|^2$ and $1-\eta\in[0,1]$.

For $\mathbb{E}\|\Delta_1\|^2$, using the fact that each sample is independent from each other and bounded variance condition (2.2) in Assumption 2.2, we have that

$$\begin{aligned}\mathbb{E}\|\Delta_1\|^2&=\mathbb{E}\left\|\frac{1}{S}\sum_{i=1}^{S}\nabla f(x_t;\xi_t^i)-\nabla F(x_t)\right\|^2\\&=\frac{1}{S^2}\sum_{i=1}^{S}\mathbb{E}\|\nabla f(x_t;\xi_t^i)-\nabla F(x_t)\|^2\le\frac{\sigma^2}{S}.\end{aligned}$$

For $\mathbb{E}\|\Delta_2\|^2$, we also have that

$$\begin{aligned}\mathbb{E}\|\Delta_2\|^2&=\mathbb{E}\left\|\frac{1}{S}\sum_{i=1}^{S}\delta(\xi_t^i)-\mathbb{E}_{\xi_t}[\delta(\xi_t)]\right\|^2=\frac{1}{S}\mathbb{E}\|\delta(\xi_t)-\mathbb{E}_{\xi_t}[\delta(\xi_t)]\|^2\\&\overset{(a)}{\le}\frac{1}{S}\mathbb{E}\|\nabla f(x_t;\xi_t)-\nabla f(x_{t-1};\xi_t)\|^2\\&\overset{(b)}{\le}\frac{\ell_f^2}{S}\mathbb{E}\|x_t-x_{t-1}\|^2=\frac{\ell_f^2}{S}\alpha^2\mathbb{E}\|h_{t-1}\|^2,\end{aligned}$$

where (a) follows from the fact that

$$\mathbb{E}_{\xi_t}\|\delta(\xi_t)-\mathbb{E}_{\xi_t}[\delta(\xi_t)]\|^2=\mathbb{E}_{\xi_t}\|\delta(\xi_t)\|^2-\|\mathbb{E}_{\xi_t}[\delta(\xi_t)]\|^2\le\mathbb{E}_{\xi_t}\|\delta(\xi_t)\|^2,$$

and (b) uses average smoothness condition (2.3) in Assumption 2.2. The proof is complete if we plug the two bounds for $\mathbb{E}\|\Delta_1\|^2$ and $\mathbb{E}\|\Delta_2\|^2$ into (A.2) and replace t with $t+1$.□

With the help of the above lemma, we analyze the average estimation error $\frac{1}{T}\sum_{t=0}^{T-1}A_t$. We first look at the case when $\eta=0$.

Lemma A.1 *For Algorithm 1 with $\eta=0$, supposing Assumption 2.2 holds, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}A_t\le\frac{\sigma^2}{D}+2\ell_f^2\frac{Q\alpha^2}{S}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2.$$

Proof When $\eta=0$, Lemma 2.5 becomes

$$A_t\le A_{t-1}+2\ell_f^2\frac{\alpha^2}{S}\mathbb{E}\|h_{t-1}\|^2,$$

for $t\in\{1,2,\cdots,Q-1\}\cup\{Q+1,Q+2,\cdots,2Q-1\}\cup\cdots$, i.e. iterations that $t\not\equiv 0\pmod Q$. Define $t=k_tQ+b_t:=\tilde{k}_t+b_t$ with $k_t=\left\lfloor\frac{t}{Q}\right\rfloor$ and $b_t=t-Q\left\lfloor\frac{t}{Q}\right\rfloor<Q$, such that $\tilde{k}_t\le t<\tilde{k}_t+Q$. We can trace the error A_t back to $A_{\tilde{k}_t}$. First, it is easy to see

$$A_t\le A_{t-1}+2\ell_f^2\frac{\alpha^2}{S}\mathbb{E}\|h_{t-1}\|^2\le A_{t-2}+2\ell_f^2\frac{\alpha^2}{S}\sum_{r=t-2}^{t-1}\mathbb{E}\|h_r\|^2.$$

Then by induction, we have

$$A_t \le A_{\tilde{k}_t} + 2\ell_f^2 \frac{\alpha^2}{S} \sum_{r=\tilde{k}_t}^{t-1} \mathbb{E}\|h_r\|^2 \le \frac{\sigma^2}{D} + 2\ell_f^2 \frac{\alpha^2}{S} \sum_{r=\tilde{k}_t}^{t-1} \mathbb{E}\|h_r\|^2. \tag{A.3}$$

The last inequality holds since for iteration $\tilde{k}_t = k_t Q$, by bounded variance condition (2.2) in Assumption 2.2,

$$A_{\tilde{k}_t} = \mathbb{E}\left\| \frac{1}{D} \sum_{i=1}^{D} \nabla f(x_{\tilde{k}_t}; \xi_{\tilde{k}_t}^i) - \nabla F(x_{\tilde{k}_t}) \right\|^2 \le \frac{\sigma^2}{D}.$$

By convention, we let $\sum_{r=\tilde{k}_t}^{\tilde{k}_t-1} \mathbb{E}\|h_r\|^2 = 0$. Thus we have (A.3) for any $t = 0, 1, \cdots, T-1$. Summing up from $t = 0$ to $t = T-1$, we obtain

$$\begin{aligned} \sum_{t=0}^{T-1} A_t &\le \sum_{t=0}^{T-1} \frac{\sigma^2}{D} + 2\ell_f^2 \frac{\alpha^2}{S} \sum_{t=0}^{T-1} \sum_{r=\tilde{k}_t}^{t-1} \mathbb{E}\|h_r\|^2 \\ &\le T\frac{\sigma^2}{D} + 2\ell_f^2 \frac{Q\alpha^2}{S} \sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2. \end{aligned}$$

To see the last inequality, we denote $H_t = \sum_{r=\tilde{k}_t}^{t-1} \mathbb{E}\|h_r\|^2$. For each t, H_t sums from $\tilde{k}_t$ to $t-1$, involving at most $t - \tilde{k}_t = b_t < Q$ entries. Equivalently, for each t, the entry $\mathbb{E}\|h_t\|^2$ only appears in $H_{t+1}, H_{t+2}, \cdots, H_{\tilde{k}_t+Q-1}$. To explain why, let $t' > t$ be some iteration after t. If $t' \le \tilde{k}_t + Q - 1$, we have $\tilde{k}_{t'} = \tilde{k}_t \le t$, and thus $\mathbb{E}\|h_t\|^2$ appears in $H_{t'}$. If $t' \ge \tilde{k}_t + Q$, then $\tilde{k}_{t'} = \tilde{k}_t + Q > t$ since $b_t < Q$, and thus $\mathbb{E}\|h_t\|^2$ does not appear in $H_{t'}$. Therefore, in the double sum $\sum_{t=0}^{T-1} H_t$, each $\mathbb{E}\|h_t\|^2$ only appears $\tilde{k}_t + Q - 1 - t = Q - b_t - 1 < Q$ times. As a result, we have

$$\sum_{t=0}^{T-1} \sum_{r=\tilde{k}_t}^{t-1} \mathbb{E}\|h_r\|^2 \le \sum_{t=0}^{T-2} Q \cdot \mathbb{E}\|h_t\|^2 \le \sum_{t=0}^{T-1} Q \cdot \mathbb{E}\|h_t\|^2, \tag{A.4}$$

and the proof is complete by dividing T on both sides. □

Lemma A.1 implies that, when $\eta = 0$, we need large D and small Q to control the error. It means that we need to compute multiple checkpoint gradients, each with a large batch size. This can be avoided by introducing the momentum parameter $\eta > 0$.

Lemma A.2 *For Algorithm 1 with $\eta \neq 0$, supposing Assumption 2.2 holds, we have*

$$\frac{1}{T} \sum_{t=0}^{T-1} A_t \le \frac{\sigma^2}{c_\eta \alpha D} \frac{1}{\alpha T} + \frac{2c_\eta \sigma^2}{S} \frac{1}{\alpha T} \sum_{t=0}^{T-1} \alpha^3 + \frac{2\ell_f^2}{c_\eta S} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2,$$

with the choice that $\eta = c_\eta \alpha^2$ for some constant $c_\eta \in (0, 1/\alpha^2)$.

Proof For any $t = 0, 1, \cdots, Q-2$, Lemma 2.5 holds true. Rearranging terms and dividing both sides by α, we have

$$\frac{\eta}{\alpha} A_t \le \frac{A_t}{\alpha} - \frac{A_{t+1}}{\alpha} + \frac{2\sigma^2}{S} \frac{\eta^2}{\alpha} + \frac{2\ell_f^2 \alpha}{S} \mathbb{E}\|h_t\|^2.$$

Under the choice that $\eta = c_\eta \alpha^2$, we have for any $t = 0, 1, \cdots, Q-2$,

$$c_\eta \alpha A_t \le \frac{A_t}{\alpha} - \frac{A_{t+1}}{\alpha} + \frac{2c_\eta^2 \sigma^2}{S} \alpha^3 + \frac{2\ell_f^2 \alpha}{S} \mathbb{E}\|h_t\|^2.$$

Summing up from $t = 0$ to $t = Q - 2$, since $A_{Q-1} \geq 0$, we obtain

$$c_\eta \alpha \sum_{t=0}^{Q-2} A_t \leq \frac{A_0}{\alpha} + \frac{2c_\eta^2 \sigma^2}{S} \sum_{t=0}^{Q-2} \alpha^3 + \frac{2\ell_f^2 \alpha}{S} \sum_{t=0}^{Q-2} \mathbb{E}\|h_t\|^2$$
$$\leq \frac{\sigma^2}{\alpha D} + \frac{2c_\eta^2 \sigma^2}{S} \sum_{t=0}^{Q-2} \alpha^3 + \frac{2\ell_f^2 \alpha}{S} \sum_{t=0}^{Q-2} \mathbb{E}\|h_t\|^2. \tag{A.5}$$

The last inequality holds since in iteration $t = 0$, we compute a checkpoint gradient with batch size D. Different from Lemma A.1, we already have the bound for summation of A_t and there is no double-sum involved. Thus we can just set $Q = T$.

For the case when $t = Q - 1 = T - 1$, to make sure that Lemma 2.5 holds, for simplicity of the analysis, we can run one additional recursive gradient update to compute h_T and thus A_T is well-defined. This will not affect the sample complexity and output of the Algorithm. Therefore, the summation in (A.5) can be extended to $t = T - 1$ and the proof is complete by dividing both sides by $c_\eta \alpha T$. □

From Lemma A.2, when $\eta \neq 0$, we can set $Q = T$ and the bound has a better dependence on D. This means that we only need to compute the checkpoint gradient for one time with a much smaller batch size at the beginning of the algorithm.

Combining Lemma A.1 and A.2, the proof of Lemma 2.6 is complete.

A.2 Proofs of Results in Chapter 3

This section contains the proof of Theorem 3.2 for the convergence rate of Algorithm 1 with general biased oracle under ABG condition, and proofs of three convergence theorems for applications of the general framework to different examples.

A.2.1 General Framework with Biased Gradient Estimates

We give the proof of Theorem 3.2 below.

Proof (Theorem 3.2) For completeness, we repeat here the analysis to obtain (3.3). Since $F(x)$ is L_F-smooth by Assumption 2.1, we have

$$F(x_{t+1}) \leq F(x_t) + \nabla F(x_t)^\top (x_{t+1} - x_t) + \frac{L_F}{2}\|x_{t+1} - x_t\|^2$$
$$\overset{(a)}{=} F(x_t) - \alpha \nabla F(x_t)^\top h_t + \frac{L_F}{2}\alpha^2 \|h_t\|^2$$
$$\overset{(b)}{=} F(x_t) - \alpha \left(\frac{1}{2}\|\nabla F(x_t)\|^2 + \frac{1}{2}\|h_t\|^2 - \frac{1}{2}\|h_t - \nabla F(x_t)\|^2 \right) + \frac{L_F}{2}\alpha^2 \|h_t\|^2$$
$$\overset{(c)}{\leq} F(x_t) - \frac{\alpha}{2}\|\nabla F(x_t)\|^2 - \frac{\alpha}{3}\|h_t\|^2 + \frac{\alpha}{2}\|h_t - \nabla F(x_t)\|^2,$$

where (a) uses the update rule, (b) follows from the fact that $2a^\top b = \|a\|^2 + \|b\|^2 - \|a - b\|^2$ and (c) uses the choice that $\alpha \leq \frac{1}{3L_F}$.

After taking full expectations, rearranging terms and dividing both sides by $\alpha/2$, we get

$$\mathbb{E}\|\nabla F(x_t)\|^2 \leq \frac{2}{\alpha}\mathbb{E}[F(x_t) - F(x_{t+1})] - \frac{2}{3}\mathbb{E}\|h_t\|^2 + \mathbb{E}\|h_t - \nabla F(x_t)\|^2.$$

We proceed by bounding the last term using the inequality that $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$.

$$\begin{aligned}\mathbb{E}\|h_t - \nabla F(x_t)\|^2 &= \mathbb{E}\|h_t - \mathbb{E}_{\xi_t}[\bar{\nabla} f(x_t;\xi_t)] + \mathbb{E}_{\xi_t}[\bar{\nabla} f(x_t;\xi_t)] - \nabla F(x_t)\|^2 \\ &\le 2\mathbb{E}\|h_t - \mathbb{E}_{\xi_t}[\bar{\nabla} f(x_t;\xi_t)]\|^2 + 2\mathbb{E}\|\mathbb{E}_{\xi_t}[\bar{\nabla} f(x_t;\xi_t)] - \nabla F(x_t)\|^2 \\ &= 2A_t + 2B_t.\end{aligned}$$

Therefore, we have

$$\mathbb{E}\|\nabla F(x_t)\|^2 \le \frac{2}{\alpha}\mathbb{E}[F(x_t) - F(x_{t+1})] - \frac{2}{3}\mathbb{E}\|h_t\|^2 + 2A_t + 2B_t.$$

By the fact that $\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(x_t)\|^2$, summing up from $t=0$ to $t = T-1$ and dividing both sides by T, we have

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{2}{\alpha T}\sum_{t=0}^{T-1}\mathbb{E}[F(x_t) - F(x_{t+1})] - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t + \frac{2}{T}\sum_{t=0}^{T-1}B_t \\ &\le \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t + \frac{2}{T}\sum_{t=0}^{T-1}B_t,\end{aligned}$$

since $F(x_T) \ge F^* := \min_x F(x)$. Utilizing the bound in Theorem 2.7 for the estimation error A_t and the (ρ_0,ρ_1)-ABG condition (note that ρ_1-SBG condition implies $(0,\rho_1)$-ABG condition) for the bias B_t, we have for $\rho_A \in (0,1/3]$,

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{2[F(x_0) - F^*] + 2\sigma^2 + 2\rho_0}{\alpha T} - \left(\frac{2}{3} - 2\rho_A\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 + 2\rho_1\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \\ &\le \frac{2[F(x_0) - F^*] + 2\sigma^2 + 2\rho_0}{\alpha T} + 2\rho_1\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2.\end{aligned}$$

As a result, if $\rho_1 \in [0,1/2)$, it holds that

$$\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \le \frac{2[F(x_0) - F^*] + 2\sigma^2 + 2\rho_0}{(1-2\rho_1)\alpha T},$$

and the proof is complete. By Assumption 2.1, F^* is finite, and then $\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 = \mathcal{O}(1/(\alpha T))$. The oracle complexity is thus $\mathcal{O}(\varepsilon^{-3})$ by Remark 2.9 when setting $\alpha T = \mathcal{O}(\varepsilon^{-2})$ to guarantee an ε-stationary point. □

A.2.2 Stochastic Bilevel Optimization

Here we present the proof of Theorem 3.8, i.e. the convergence rate of Algorithm 2 to solve stochastic bilevel optimization (3.6). In order to prove Theorem 3.8, we also give the proofs of Lemma 3.6 and 3.7 to control the estimation error and the bias.

We first prove Lemma 3.6.

Proof (Lemma 3.6) The bound for $\frac{1}{T}\sum_{t=0}^{T-1}A_t^f$ immediately follows from similar arguments as Lemma 2.6 and Theorem 2.7 by the setting that $\beta^2 = \alpha^2/c_\beta^2$. The bound for $\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^g$ can be obtained in a similar way. By (3.13), the lower-level estimation error satisfies

$$\begin{aligned}A_{t+1}^g &\le (1-\eta)A_t^g + 2\sigma_g^2\frac{\eta^2}{S} + \frac{2}{S}\mathbb{E}\|\nabla g(x_{t+1},y_{t+1};\zeta_{t+1}) - \nabla g(x_t,y_t;\zeta_{t+1})\|^2 \\ &\le (1-\eta)A_t^g + 2\sigma_g^2\frac{\eta^2}{S} + 2\ell_g^2\frac{\alpha^2}{S}\mathbb{E}\|h_t^f\|^2 + 2\ell_g^2\frac{\beta^2}{S}\mathbb{E}\|h_t^g\|^2.\end{aligned}$$

Note that we use the same set of parameters when estimating both the upper-level and the lower-level gradients. Then with similar arguments, we have

$$\frac{1}{T}\sum_{t=0}^{T-1} A_{t+1}^g \leq \frac{\sigma_g^2}{\alpha T} + \rho_A \frac{\ell_g^2}{\ell_f^2}\cdot\frac{1}{T}\sum_{t=0}^{T-2}\mathbb{E}\|h_{t+1}^f\|^2 + \frac{\rho_A}{c_\beta^2}\frac{\ell_g^2}{\ell_f^2}\cdot\frac{1}{T}\sum_{t=0}^{T-2}\mathbb{E}\|h_{t+1}^g\|^2.$$

The reason why we can only sum up from 0 to $T-2$ comes from (A.4) and (A.5). The proof is complete since $\sum_{t=0}^{T-2}\mathbb{E}\|h_{t+1}^f\|^2 = \sum_{t=1}^{T-1}\mathbb{E}\|h_t^f\|^2 \leq \sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2$ and that $\sum_{t=0}^{T-2}\mathbb{E}\|h_{t+1}^g\|^2 \leq \sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^g\|^2$. The selection of ρ_A and c_β is explained in the proof of Theorem 3.8. □

Then we give the proof of Lemma 3.7.

Proof (Lemma 3.7) To make sure that y_{T+1} is well-defined and to simplify the analysis, we run one additional step $y_{T+1} = y_T - \beta h_T^g$ after obtaining x_T. For $t = 1, \cdots, T$, by the update $y_{t+1} = y_t - \beta h_t^g$,

$$\begin{aligned}\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 &= \mathbb{E}\|y_t - y^*(x_t)\|^2 + \beta^2\mathbb{E}\|h_t^g\|^2 - 2\beta\mathbb{E}\big[\big(h_t^g\big)^\top\big(y_t - y^*(x_t)\big)\big] \\ &= \mathbb{E}\|y_t - y^*(x_t)\|^2 + \beta^2\mathbb{E}\|h_t^g\|^2 - 2\beta\mathbb{E}\big[\nabla_y G(x_t, y_t)^\top\big(y_t - y^*(x_t)\big)\big] \\ &\quad + 2\beta\mathbb{E}\big[\big(\nabla_y G(x_t, y_t) - h_t^g\big)^\top\big(y_t - y^*(x_t)\big)\big]. \end{aligned} \tag{A.6}$$

Since $y^*(x_t)$ is the lower-level optimal solution given x_t, we have $\nabla_y G(x_t, y^*(x_t)) = 0$, then

$$\begin{aligned}\mathbb{E}\big[\nabla_y G(x_t, y_t)^\top\big(y_t - y^*(x_t)\big)\big] &= \mathbb{E}\big[\big(\nabla_y G(x_t, y_t) - \nabla_y G(x_t, y^*(x_t))\big)^\top\big(y_t - y^*(x_t)\big)\big] \\ &\overset{(a)}{\geq} \frac{\mu_g}{\mu_g + L_g}\mathbb{E}\|y_t - y^*(x_t)\|^2 + \frac{1}{\mu_g + L_g}\mathbb{E}\|\nabla_y G(x_t, y_t) - \nabla_y G(x_t, y^*(x_t))\|^2 \\ &= \frac{\mu_g}{\mu_g + L_g}\mathbb{E}\|y_t - y^*(x_t)\|^2 + \frac{1}{\mu_g + L_g}\mathbb{E}\|\nabla_y G(x_t, y_t)\|^2 \\ &= \frac{\mu_g}{\mu_g + L_g}\mathbb{E}\|y_t - y^*(x_t)\|^2 + \frac{1}{\mu_g + L_g}\mathbb{E}\|h_t^g - \big(h_t^g - \nabla_y G(x_t, y_t)\big)\|^2 \\ &\overset{(b)}{\geq} \frac{\mu_g}{\mu_g + L_g}\mathbb{E}\|y_t - y^*(x_t)\|^2 + \frac{1}{2(\mu_g + L_g)}\mathbb{E}\|h_t^g\|^2 - \frac{1}{\mu_g + L_g}\mathbb{E}\|h_t^g - \nabla_y G(x_t, y_t)\|^2, \end{aligned} \tag{A.7}$$

where (b) follows from the inequality $\|a-b\|^2 \geq \frac{1}{2}\|a\|^2 - \|b\|^2$ and (a) uses a well-known result [90] that for any μ-strongly convex and L-smooth function $f(x)$, the following is true:

$$\big(\nabla f(x) - \nabla f(y)\big)^\top(x - y) \geq \frac{\mu}{\mu + L}\|x - y\|^2 + \frac{1}{\mu + L}\|\nabla f(x) - \nabla f(y)\|^2.$$

Plugging (A.7) into (A.6), we get

$$\begin{aligned}\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 \leq \left(1 - \frac{2\mu_g\beta}{\mu_g + L_g}\right)\mathbb{E}\|y_t - y^*(x_t)\|^2 - \beta\left(\frac{1}{\mu_g + L_g} - \beta\right)\mathbb{E}\|h_t^g\|^2 \\ + \frac{2\beta}{\mu_g + L_g}A_t^g + 2\beta\mathbb{E}\big[\big(\nabla_y G(x_t, y_t) - h_t^g\big)^\top\big(y_t - y^*(x_t)\big)\big]. \end{aligned} \tag{A.8}$$

By the inequality $2a^\top b \leq \|a\|^2 + \|b\|^2$, for some $\gamma_1 > 0$, we have

$$2\mathbb{E}\big[\big(\nabla_y G(x_t, y_t) - h_t^g\big)^\top\big(y_t - y^*(x_t)\big)\big] \leq \frac{1}{\gamma_1}\mathbb{E}\|h_t^g - \nabla_y G(x_t, y_t)\|^2 + \gamma_1\mathbb{E}\|y_t - y^*(x_t)\|^2. \tag{A.9}$$

We define $c_g = \frac{\mu_g}{\mu_g + L_g}$ such that $\frac{1}{\mu_g + L_g} = \frac{c_g}{\mu_g}$ and choose $\gamma_1 = c_g$. Combining (A.8) and (A.9), we obtain

$$\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 \leq (1 - c_g\beta)\mathbb{E}\|y_t - y^*(x_t)\|^2 - \beta\left(\frac{c_g}{\mu_g} - \beta\right)\mathbb{E}\|h_t^g\|^2 + \left(\frac{2c_g}{\mu_g} + \frac{1}{c_g}\right)\beta A_t^g.$$

To handle $\mathbb{E}\|y_t - y^*(x_t)\|^2$ and trace it back to $\mathbb{E}\|y_t - y^*(x_{t-1})\|^2$, we have the bound

$$\begin{aligned}
\mathbb{E}\|y_t - y^*(x_t)\|^2 &= \mathbb{E}\|y_t - y^*(x_{t-1}) + y^*(x_{t-1}) - y^*(x_t)\|^2 \\
&\overset{(a)}{\leq} (1 + \gamma_2)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \left(1 + \frac{1}{\gamma_2}\right)\mathbb{E}\|y^*(x_{t-1}) - y^*(x_t)\|^2 \\
&\overset{(b)}{\leq} (1 + \gamma_2)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \left(1 + \frac{1}{\gamma_2}\right)L_y^2\mathbb{E}\|x_{t-1} - x_t\|^2 \\
&\overset{(c)}{=} (1 + \gamma_2)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \left(1 + \frac{1}{\gamma_2}\right)L_y^2\alpha^2\mathbb{E}\|h_{t-1}^f\|^2,
\end{aligned}$$

where (a) uses $2a^\top b \leq \|a\|^2 + \|b\|^2$ again for some $\gamma_2 > 0$, (b) follows from property (3.9) in Lemma 3.4 and (c) holds because of the update rule.

We choose $\gamma_2 = \frac{c_g\beta}{2(1 - c_g\beta)}$ such that $(1 - c_g\beta)(1 + \gamma_2) = 1 - \frac{c_g}{2}\beta$ and $(1 - c_g\beta)(1 + \frac{1}{\gamma_2}) < 1 + \frac{1}{\gamma_2} = \frac{2}{c_g\beta} - 1 < \frac{2}{c_g\beta}$, therefore

$$\begin{aligned}
\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 \leq \left(1 - \frac{c_g}{2}\beta\right)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \frac{2L_y^2\alpha^2}{c_g\beta}\mathbb{E}\|h_{t-1}^f\|^2 \\
- \beta\left(\frac{c_g}{\mu_g} - \beta\right)\mathbb{E}\|h_t^g\|^2 + \left(\frac{2c_g}{\mu_g} + \frac{1}{c_g}\right)\beta A_t^g.
\end{aligned}$$

After rearranging terms and dividing both sides by $\frac{c_g}{2}\beta$, we obtain

$$\begin{aligned}
\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 &\leq \frac{2}{c_g\beta}\left(\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 - \mathbb{E}\|y_{t+1} - y^*(x_t)\|^2\right) + \frac{4L_y^2}{c_g^2} \cdot \frac{\alpha^2}{\beta^2}\mathbb{E}\|h_{t-1}^f\|^2 \\
&\quad - \frac{2}{\mu_g}\left(1 - \frac{\mu_g}{c_g}\beta\right)\mathbb{E}\|h_t^g\|^2 + \left(\frac{4}{\mu_g} + \frac{2}{c_g^2}\right)A_t^g \\
&\leq \frac{2}{c_g\beta}\left(\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 - \mathbb{E}\|y_{t+1} - y^*(x_t)\|^2\right) + \frac{4L_y^2c_\beta^2}{c_g^2}\mathbb{E}\|h_{t-1}^f\|^2 \\
&\quad - \frac{1}{\mu_g}\mathbb{E}\|h_t^g\|^2 + \left(\frac{4}{\mu_g} + \frac{2}{c_g^2}\right)A_t^g,
\end{aligned}$$

since $\frac{\alpha}{\beta} = c_\beta$ and that $\beta \leq \frac{c_g}{2\mu_g}$. Summing up from 1 to T and dividing both sides by T, we get

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 \leq \frac{2}{c_g\beta T}\|y_1 - y^*(x_0)\|^2 + \frac{4L_y^2c_\beta^2}{c_g^2} \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 \\
- \frac{1}{\mu_g T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^g\|^2 + \left(\frac{4}{\mu_g} + \frac{2}{c_g^2}\right)\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^g,
\end{aligned}$$

and the proof is complete since $\beta = \alpha/c_\beta$ for constant c_β we define in the proof of Theorem 3.8. □

Now we can prove Theorem 3.8.

Proof (Theorem 3.8) By (3.3) or the proof of Theorem 3.2, we get

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{2[F(x_0)-F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f - \nabla F(x_t)\|^2 \\ &\le \frac{2[F(x_0)-F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^f + \frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|B(x_t,y_{t+1})\|^2,\end{aligned}$$

where $B(x_t, y_{t+1})$ defined in (3.14) satisfies

$$\mathbb{E}\|B(x_t,y_{t+1})\|^2 \le \frac{2C_{g_{xy}}^2 C_{f_y}^2}{\mu_g^2}(1-\frac{\mu_g}{L_g})^{2K} + 2L^2\mathbb{E}\|y_{t+1}-y^*(x_t)\|^2,$$

by the inequality $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$. The first term, i.e. the bias induced by approximating inverse Hessian, is less than $1/(\alpha T)$ for $K \ge \frac{L_g}{2\mu_g}\log\left(\frac{2C_{g_{xy}}^2 C_{f_y}^2}{\mu_g^2}\alpha T\right)$ since $\mu_g \le L_g$. The second term, the error from approximating lower-level optimal solution $y^*(x_t)$ by y_{t+1}, comes from (3.8) in Lemma 3.4.

Putting everything together, by Lemma 3.7, we obtain

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{2[F(x_0)-F^*]+2}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^f + 4L^2\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|y_{t+1}-y^*(x_t)\|^2 \\ &\le \frac{2[F(x_0)-F^*]+2}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^f + \frac{16L^2L_y^2c_\beta^2}{c_g^2}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 \\ &\quad + \frac{8c_\beta L^2}{c_g\alpha T}\|y_1-y^*(x_0)\|^2 - \frac{4L^2}{\mu_g T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^g\|^2 + \left(\frac{16}{\mu_g}+\frac{8}{c_g^2}\right)L^2\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^g \\ &\le \frac{2[F(x_0)-F^*]+2+8c_\beta L^2\|y_1-y^*(x_0)\|^2/c_g}{\alpha T} \\ &\quad - \left(\frac{2}{3}-\frac{16L^2L_y^2c_\beta^2}{c_g^2}\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^f \\ &\quad - \frac{4L^2}{\mu_g}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^g\|^2 + \left(\frac{16}{\mu_g}+\frac{8}{c_g^2}\right)L^2\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^g,\end{aligned}$$

where we observe some symmetry between the upper-level and lower-level descent. Using Lemma 3.6 to bound both A_t^f and A_{t+1}^g, we have

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{2[F(x_0)-F^*]+2+8c_\beta L^2\|y_1-y^*(x_0)\|^2/c_g+2\sigma_f^2+(16/\mu_g+8/c_g^2)L^2\sigma_g^2}{\alpha T} \\ &\quad - \left(\frac{2}{3}-\frac{16L^2L_y^2c_\beta^2}{c_g^2}-2\rho_A-\left(\frac{16}{\mu_g}+\frac{8}{c_g^2}\right)L^2\frac{\ell_g^2}{\ell_f^2}\rho_A\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^f\|^2 \\ &\quad - \left(\frac{4L^2}{\mu_g}-\frac{2\rho_A}{c_\beta^2}-\left(\frac{16}{\mu_g}+\frac{8}{c_g^2}\right)L^2\frac{\ell_g^2}{\ell_f^2}\frac{\rho_A}{c_\beta^2}\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^g\|^2.\end{aligned}$$

If we choose $c_\beta = \frac{c_g}{4\sqrt{6}LL_y}$ such that $\frac{16L^2L_y^2c_\beta^2}{c_g^2} = \frac{1}{6}$, and

$$\rho_A = \min\left\{\frac{1}{12}, \frac{\mu_g\ell_f^2}{96L^2\ell_g^2}, \frac{c_g^2\ell_f^2}{48L^2\ell_g^2}, \frac{c_\beta^2L^2}{\mu_g}, \frac{c_\beta^2\ell_f^2}{16\ell_g^2}, \frac{c_g^2c_\beta^2\ell_f^2}{8\mu_g\ell_g^2}\right\}$$

such that

$$\frac{2}{3} - \frac{16L^2L_y^2c_\beta^2}{c_g^2} - 2\rho_A - \left(\frac{16}{\mu_g} + \frac{8}{c_g^2}\right)L^2\frac{\ell_g^2}{\ell_f^2}\rho_A \geq 0,$$
$$\frac{4L^2}{\mu_g} - \frac{2\rho_A}{c_\beta^2} - \left(\frac{16}{\mu_g} + \frac{8}{c_g^2}\right)L^2\frac{\ell_g^2}{\ell_f^2}\frac{\rho_A}{c_\beta^2} \geq 0,$$

we get the desired result. If $\|y_1 - y^*(x_0)\|^2$ is bounded, and this can be achieved at least by some additional steps for $\min_y g(x_0, y)$, we can choose $\alpha T = \mathcal{O}(\varepsilon^{-2})$ to obtain an ε-stationary point. The sample complexity is thus $\tilde{\mathcal{O}}(\varepsilon^{-3})$ since $K = \mathcal{O}(\log(\alpha T)) = \mathcal{O}(\log \varepsilon^{-1})$ and the total number of samples required for Algorithm 2 is $\mathcal{O}(T(S + D/Q)(1 + K))$. The requirement for $\beta \leq 1/2(\mu_g + L_g)$ in Lemma 3.7 can be satisfied by choosing α with an additional requirement that $\alpha \leq c_\beta/2(\mu_g + L_g)$ in Lemma 3.6 since $\beta = \alpha/c_\beta$. □

A.2.3 Stochastic Minimax Optimization

In this section, for Algorithm 3 to solve stochastic minimax optimization (3.15), we prove its convergence rate in Theorem 3.14 for the non-convex strongly-concave case and Theorem 3.20 for the non-convex P-Ł case.

Non-Convex Strongly-Concave Case

We first give the proof of Lemma 3.13. The proof is very much similar to the proof of Lemma 3.7 for bilevel optimization.

Proof (Lemma 3.13) By non-expansion of projection, we have

$$\begin{aligned}
\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 &\leq \mathbb{E}\|y_t + \beta h_t^y - y^*(x_t)\|^2 \\
&= \mathbb{E}\|y_t - y^*(x_t)\|^2 + \beta^2\mathbb{E}\|h_t^y\|^2 + 2\beta\mathbb{E}\left[\left((h_t^y)^\top\left(y_t - y^*(x_t)\right)\right]\right. \\
&= \mathbb{E}\|y_t - y^*(x_t)\|^2 + \beta^2\mathbb{E}\|h_t^y\|^2 + 2\beta\mathbb{E}\left[\nabla_y G(x_t, y_t)^\top\left(y_t - y^*(x_t)\right)\right] \\
&\quad + 2\beta\mathbb{E}\left[\left(h_t^y - \nabla_y G(x_t, y_t)\right)^\top\left(y_t - y^*(x_t)\right)\right]. \qquad \text{(A.10)}
\end{aligned}$$

We know that $-G(x, y)$ is L-smooth and μ-strongly convex in y. The ascent step to maximize over $G(x, \cdot)$ is equivalent to the descent step to minimize over $-G(x, \cdot)$. Therefore, the same as the proof of Lemma 3.7 for $-G(x, \cdot)$, by (A.7), we obtain that

$$\mathbb{E}\left[\nabla_y G(x_t, y_t)^\top\left(y_t - y^*(x_t)\right)\right] \leq -\frac{1}{1+\kappa}\mathbb{E}\|y_t - y^*(x_t)\|^2 - \frac{1}{2\mu(1+\kappa)}\mathbb{E}\|h_t^y\|^2 + \frac{1}{\mu(1+\kappa)}A_t^y. \qquad \text{(A.11)}$$

By (A.9) with $\gamma_1 = \frac{1}{1+\kappa}$, we have

$$2\mathbb{E}\left[\left(\nabla_y G(x_t, y_t) - h_t^y\right)^\top\left(y_t - y^*(x_t)\right)\right] \leq (1+\kappa)A_t^y + \frac{1}{1+\kappa}\mathbb{E}\|y_t - y^*(x_t)\|^2. \qquad \text{(A.12)}$$

Since $y^*(x)$ is Lipschitz continuous by Lemma 3.10, we know

$$\begin{aligned}
\mathbb{E}\|y_t - y^*(x_t)\|^2 &\leq (1+\gamma_2)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \left(1 + \frac{1}{\gamma_2}\right)\mathbb{E}\|y^*(x_{t-1}) - y^*(x_t)\|^2 \\
&\leq (1+\gamma_2)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \left(1 + \frac{1}{\gamma_2}\right)\kappa^2\mathbb{E}\|x_{t-1} - x_t\|^2 \\
&= (1+\gamma_2)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \left(1 + \frac{1}{\gamma_2}\right)\kappa^2\alpha^2\mathbb{E}\|h_{t-1}^x\|^2. \qquad \text{(A.13)}
\end{aligned}$$

Plugging (A.11), (A.12) and (A.13) into (A.10) and choosing γ_2 in the same way as in the proof of Lemma 3.7, we obtain

$$\begin{aligned}\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 &\leq \left(1 - \frac{\beta}{2(1+\kappa)}\right)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + \frac{2(1+\kappa)\kappa^2\alpha^2}{\beta}\mathbb{E}\|h_{t-1}^x\|^2 \\ &\quad - \beta\left(\frac{1}{\mu(1+\kappa)} - \beta\right)\mathbb{E}\|h_t^y\|^2 + \left(\frac{2}{\mu(1+\kappa)} + (1+\kappa)\right)\beta A_t^y \\ &\leq \left(1 - \frac{\beta}{4\kappa}\right)\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 + 4\kappa^3\frac{\alpha^2}{\beta}\mathbb{E}\|h_{t-1}^x\|^2 \\ &\quad - \beta\left(\frac{1}{2\mu\kappa} - \beta\right)\mathbb{E}\|h_t^y\|^2 + \left(\frac{1}{\mu} + 2\kappa\right)\beta A_t^y,\end{aligned}$$

since $\kappa \geq 1$. After rearranging terms and dividing both sides by $\frac{\beta}{4\kappa}$, we obtain

$$\begin{aligned}\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 &\leq \frac{4\kappa}{\beta}\Big(\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 - \mathbb{E}\|y_{t+1} - y^*(x_t)\|^2\Big) + 16\kappa^4\frac{\alpha^2}{\beta^2}\mathbb{E}\|h_{t-1}^x\|^2 \\ &\quad - \left(\frac{2}{\mu} - 4\kappa\beta\right)\mathbb{E}\|h_t^y\|^2 + \left(\frac{4\kappa}{\mu} + 8\kappa^2\right)A_t^y \\ &\leq \frac{4\kappa}{\beta}\Big(\mathbb{E}\|y_t - y^*(x_{t-1})\|^2 - \mathbb{E}\|y_{t+1} - y^*(x_t)\|^2\Big) + 16\kappa^4 c_\beta^2\mathbb{E}\|h_{t-1}^x\|^2 \\ &\quad - \frac{1}{\mu}\mathbb{E}\|h_t^y\|^2 + \left(\frac{4\kappa}{\mu} + 8\kappa^2\right)A_t^y,\end{aligned}$$

since $\frac{\alpha}{\beta} = c_\beta$ and that $\beta \leq \frac{1}{4\kappa\mu}$. Summing up from 1 to T and dividing both sides by T, we get

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|y_{t+1} - y^*(x_t)\|^2 &\leq \frac{4\kappa}{\beta T}\|y_1 - y^*(x_0)\|^2 + 16\kappa^4 c_\beta^2 \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 \\ &\quad - \frac{1}{\mu T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^y\|^2 + \left(\frac{4\kappa}{\mu} + 8\kappa^2\right)\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^y.\end{aligned}$$

The proof is complete by replacing β with α/c_β for c_β given in the proof of Theorem 3.14. □

With Lemma 3.12 and 3.13, we give the proof of Theorem 3.14.

Proof (Theorem 3.14) By (3.3) or the proof of Theorem 3.2 for the general biased oracle, we get

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\leq \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^x + \frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|B(x_t, y_{t+1})\|^2 \\ &\leq \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^x + 2L^2 \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|y_{t+1} - y_t\|^2,\end{aligned}$$

by (3.16). Since Lemma 3.13 holds and that $\kappa = \frac{L}{\mu}$, we have

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\leq \frac{2[F(x_0) - F^*] + 8\kappa L^2 c_\beta\|y_1 - y^*(x_0)\|^2}{\alpha T} \\ &\quad - \left(\frac{2}{3} - 32\kappa^4 L^2 c_\beta^2\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^x \\ &\quad - 2\kappa L \cdot \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^y\|^2 + (8\kappa^2 L + 16\kappa^2 L^2)\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^y.\end{aligned}$$

By Lemma 3.12 with an additional requirement that $\alpha \leq \frac{c_\beta}{4\kappa\mu}$ to guarantee that $\beta \leq \frac{1}{4\kappa\mu}$, we obtain

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \leq \frac{2[F(x_0)-F^*]+8\kappa L^2 c_\beta\|y_1-y^*(x_0)\|^2+(2+8\kappa^2 L+16\kappa^2L^2)\sigma^2}{\alpha T}\\ -\left(\frac{2}{3}-32\kappa^4L^2c_\beta^2-2\rho_A-(8\kappa^2L+16\kappa^2L^2)\rho_A\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2\\ -\left(2\kappa L-\frac{2\rho_A}{c_\beta^2}-(8\kappa^2L+16\kappa^2L^2)\frac{\rho_A}{c_\beta^2}\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^y\|^2.\end{aligned}$$

If we choose $c_\beta = \frac{1}{8\sqrt{3}\kappa^2 L}$ such that $32\kappa^4L^2c_\beta^2 = \frac{1}{6}$, and

$$\rho_A = \min\left\{\frac{1}{12}, \frac{1}{48\kappa^2L}, \frac{1}{96\kappa^2L^2}, \frac{\kappa L c_\beta^2}{2}, \frac{c_\beta^2}{16\kappa}, \frac{c_\beta^2}{32\kappa L}\right\}$$

such that

$$\begin{aligned}\frac{2}{3}-32\kappa^4L^2c_\beta^2-2\rho_A-(8\kappa^2L+16\kappa^2L^2)\rho_A \geq 0,\\ 2\kappa L-\frac{2\rho_A}{c_\beta^2}-(8\kappa^2L+16\kappa^2L^2)\frac{\rho_A}{c_\beta^2}\geq 0,\end{aligned}$$

we get the desired result. Note that the number of samples required for Algorithm 3 is $\mathcal{O}(T(2S+2D/Q))$. By Remark 2.9, since $\|y_1-y^*(x_0)\|^2$ is bounded by Assumption 3.9, if we choose $\alpha T = \mathcal{O}(\varepsilon^{-2})$ to guarantee an ε-stationary point, the sample complexity is $\mathcal{O}(\varepsilon^{-3})$. □

Non-Convex P-Ł Case

We give the proof of Lemma 3.19 below.

Proof (Lemma 3.19) Let $t = 1, 2, \cdots, T$. We know that $-G(x,y)$ is L-smooth w.r.t. y, thus

$$\begin{aligned}-G(x_t,y_{t+1}) &\leq -G(x_t,y_t)-\nabla_y G(x_t,y_t)^\top(y_{t+1}-y_t)+\frac{L}{2}\|y_{t+1}-y_t\|^2\\ &\overset{(a)}{=} -G(x_t,y_t)-\beta\nabla_y G(x_t,y_t)^\top h_t^y+\frac{L}{2}\beta^2\|h_t^y\|^2\\ &\overset{(b)}{=} -G(x_t,y_t)-\frac{\beta}{2}\|\nabla_y G(x_t,y_t)\|^2-\frac{\beta}{2}(1-\beta L)\|h_t^y\|^2+\frac{\beta}{2}\|h_t^y-\nabla_y G(x_t,y_t)\|^2\\ &\overset{(c)}{\leq} -G(x_t,y_t)-\beta\mu\Big(F(x_t)-G(x_t,y_t)\Big)-\frac{\beta}{2}(1-\beta L)\|h_t^y\|^2+\frac{\beta}{2}\|h_t^y-\nabla_y G(x_t,y_t)\|^2,\end{aligned}$$

where (a) follows from the update rule of y_t in Algorithm 3, (b) holds by the inequality that $2a^\top b = \|a\|^2+\|b\|^2-\|a-b\|^2$, and (c) uses Assumption 3.16 that $-G(x,y)$ satisfies P-Ł condition and the definition (3.15) that $F(x_t) = \max_y G(x_t,y)$.

Adding $F(x_t)$ on both sides of the above inequality, we obtain

$$\begin{aligned}F(x_t)-G(x_t,y_{t+1}) &\leq (1-\beta\mu)\Big(F(x_t)-G(x_t,y_t)\Big)-\frac{\beta}{2}(1-\beta L)\|h_t^y\|^2+\frac{\beta}{2}\|h_t^y-\nabla_y G(x_t,y_t)\|^2\\ &= (1-\beta\mu)\Big(F(x_{t-1})-G(x_{t-1},y_t)+G(x_{t-1},y_t)-G(x_t,y_t)+F(x_t)-F(x_{t-1})\Big)\\ &\quad -\frac{\beta}{2}(1-\beta L)\|h_t^y\|^2+\frac{\beta}{2}\|h_t^y-\nabla_y G(x_t,y_t)\|^2. \qquad (A.14)\end{aligned}$$

We first handle the term $G(x_{t-1}, y_t) - G(x_t, y_t)$. Since $-G(x, y)$ is also L-smooth w.r.t. x, we have

$$\begin{aligned} G(x_{t-1}, y_t) - G(x_t, y_t) &\leq -\nabla_x G(x_{t-1}, y_t)^\top (x_t - x_{t-1}) + \frac{L}{2}\|x_t - x_{t-1}\|^2 \\ &= \alpha \nabla_x G(x_{t-1}, y_t)^\top h_{t-1}^x + \frac{L}{2}\alpha^2 \|h_{t-1}^x\|^2 \\ &= \frac{\alpha}{2}\|\nabla_x G(x_{t-1}, y_t)\|^2 + \frac{\alpha}{2}(1 + \alpha L)\|h_{t-1}^x\|^2 - \frac{\alpha}{2}\|h_{t-1}^x - \nabla_x G(x_{t-1}, y_t)\|^2 \\ &\leq \frac{\alpha}{2}(3 + \alpha L)\|h_{t-1}^x\|^2 + \frac{\alpha}{2}\|h_{t-1}^x - \nabla_x G(x_{t-1}, y_t)\|^2, \end{aligned} \tag{A.15}$$

and the last inequality holds by the fact that

$$\begin{aligned} \|\nabla_x G(x_{t-1}, y_t)\|^2 &= \|h_{t-1}^x - (h_{t-1}^x - \nabla_x G(x_{t-1}, y_t))\|^2 \\ &\leq 2\|h_{t-1}^x\|^2 + 2\|h_{t-1}^x - \nabla_x G(x_{t-1}, y_t)\|^2. \end{aligned}$$

Let $\Delta_t = F(x_t) - G(x_t, y_{t+1})$ for simplicity, by (A.14) and (A.15), we obtain

$$\begin{aligned} \Delta_t \leq\ & (1 - \beta\mu)\Delta_{t-1} + (1 - \beta\mu)(F(x_t) - F(x_{t-1})) \\ & + \frac{\alpha}{2}(3 + \alpha L)(1 - \beta\mu)\|h_{t-1}^x\|^2 + \frac{\alpha}{2}(1 - \beta\mu)\|h_{t-1}^x - \nabla_x G(x_{t-1}, y_t)\|^2 \\ & - \frac{\beta}{2}(1 - \beta L)\|h_t^y\|^2 + \frac{\beta}{2}\|h_t^y - \nabla_y G(x_t, y_t)\|^2, \end{aligned}$$

with the choice that $\beta < 1/\mu$. Rearranging terms of the above inequality and dividing both sides by $\beta\mu$, we have

$$\begin{aligned} \Delta_{t-1} \leq\ & \frac{\Delta_{t-1} - \Delta_t}{\beta\mu} + \frac{(1 - \beta\mu)}{\beta\mu}(F(x_t) - F(x_{t-1})) \\ & + \frac{c_\beta}{2\mu}(3 + \alpha L)\|h_{t-1}^x\|^2 + \frac{c_\beta}{2\mu}\|h_{t-1}^x - \nabla_x G(x_{t-1}, y_t)\|^2 \\ & - \frac{1}{2\mu}(1 - \beta L)\|h_t^y\|^2 + \frac{1}{2\mu}\|h_t^y - \nabla_y G(x_t, y_t)\|^2, \end{aligned}$$

since $0 < 1 - \beta\mu < 1$ and the definition $c_\beta = \alpha/\beta$. Summing up the above inequality from $t = 1$ to T and dividing T on both sides, we obtain

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1}\Delta_t \leq\ & \frac{\Delta_0 - \Delta_T}{\beta\mu T} + \frac{(1 - \beta\mu)}{\beta\mu T}(F(x_T) - F(x_0)) \\ & + \frac{c_\beta}{2\mu}(3 + \alpha L)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\|h_t^x\|^2 + \frac{c_\beta}{2\mu}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\|h_t^x - \nabla_x G(x_t, y_{t+1})\|^2 \\ & - \frac{1}{2\mu}(1 - \beta L)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\|h_{t+1}^y\|^2 + \frac{1}{2\mu}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\|h_{t+1}^y - \nabla_y G(x_{t+1}, y_{t+1})\|^2. \end{aligned}$$

Note that $\Delta_0 = F(x_0) - G(x_0, y_1)$ and $\Delta_T \geq 0$ by definition of $F(x)$, finally we obtain

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Delta_t] \leq\ & \frac{F(x_0) - G(x_0, y_1)}{\beta\mu T} + \left(1 - \frac{1}{\beta\mu}\right)\frac{F(x_0) - F(x_T)}{T} \\ & + \frac{c_\beta}{2\mu}(3 + \alpha L)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + \frac{c_\beta}{2\mu}\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_t^x \\ & - \frac{1}{2\mu}(1 - \beta L)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^y\|^2 + \frac{1}{2\mu}\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^y. \end{aligned}$$

□

Now we can give the proof of Theorem 3.20.

Proof (Theorem 3.20) By Lemma 3.17, $F(x)$ is L_F-smooth with $L_F = 2\kappa L$. With the same analysis as in the proof of Theorem 3.2, since (3.17) holds, we have that

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{2[F(x_0)-F(x_T)]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^x + \frac{2}{T}\sum_{t=0}^{T-1}\mathbb{E}\|B(x_t,y_{t+1})\|^2 \\ &\le \frac{2[F(x_0)-F(x_T)]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + \frac{2}{T}\sum_{t=0}^{T-1}A_t^x + \frac{4\kappa L}{T}\sum_{t=0}^{T-1}\mathbb{E}\Big[F(x_t)-G(x_t,y_{t+1})\Big],\end{aligned}$$

Since Lemma 3.19 holds,

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \le{}& \frac{(2+4\kappa^2(\mu\alpha-c_\beta))[F(x_0)-F(x_T)]}{\alpha T} + \frac{4\kappa^2 c_\beta(F(x_0)-G(x_0,y_1))}{\alpha T} \\ &- \left(\frac{2}{3} - 2\kappa^2 L c_\beta\cdot\alpha - 6\kappa^2 c_\beta\right)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 + (2+2\kappa^2 c_\beta)\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_t^x \\ &- 2\kappa^2(1-\beta L)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^y\|^2 + 2\kappa^2\cdot\frac{1}{T}\sum_{t=0}^{T-1}A_{t+1}^y.\end{aligned}$$

By Lemma 3.12 with an additional requirement that $\alpha < \frac{c_\beta}{\mu}$ to guarantee that $\beta < \frac{1}{\mu}$, we obtain

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \le{}& \frac{(2+4\kappa^2(\mu\alpha-c_\beta))[F(x_0)-F(x_T)] + 4\kappa^2 c_\beta\Delta_0 + (2+2\kappa^2+2\kappa^2 c_\beta)\sigma^2}{\alpha T} \\ &- \left(\frac{2}{3} - 2\kappa^2 L c_\beta\cdot\alpha - 6\kappa^2 c_\beta - (2+2\kappa^2+2\kappa^2 c_\beta)\rho_A\right)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t^x\|^2 \\ &- \left(2\kappa^2 - 2\kappa^2 L\cdot\beta - (2+2\kappa^2+2\kappa^2 c_\beta)\frac{\rho_A}{c_\beta^2}\right)\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_{t+1}^y\|^2.\end{aligned}$$

We choose $c_\beta = \frac{1}{36\kappa^2}$ and $\alpha \le \frac{3}{L}$ such that $6\kappa^2 c_\beta = \frac{1}{6}$ and $2\kappa^2 L c_\beta\cdot\alpha \le \frac{1}{6}$. We choose $\beta \le \frac{1}{4L}$ such that $2\kappa^2 L\cdot\beta \le \frac{\kappa^2}{2}$. Then if ρ_A satisfies

$$\rho_A = \min\left\{\frac{1}{18}, \frac{1}{18\kappa^2}, \frac{1}{18\kappa^2 c_\beta}, \frac{\kappa^2 c_\beta^2}{4}, \frac{c_\beta^2}{4}, \frac{c_\beta}{4}\right\} = \frac{c_\beta^2}{4},$$

such that

$$\begin{aligned}&\frac{2}{3} - 2\kappa^2 L c_\beta\cdot\alpha - 6\kappa^2 c_\beta - (2+2\kappa^2+2\kappa^2 c_\beta)\rho_A \ge 0 \\ &2\kappa^2 - 2\kappa^2 L\cdot\beta - (2+2\kappa^2+2\kappa^2 c_\beta)\frac{\rho_A}{c_\beta^2} \ge 0,\end{aligned}$$

we get that

$$\begin{aligned}\mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{(2+4\kappa^2(\mu\alpha-c_\beta))[F(x_0)-F(x_T)] + 4\kappa^2 c_\beta\Delta_0 + (2+2\kappa^2+2\kappa^2 c_\beta)\sigma^2}{\alpha T} \\ &\le \frac{2[F(x_0)-F^*] + \Delta_0/9 + (37/18+2\kappa^2)\sigma^2}{\alpha T},\end{aligned}$$

where the last inequality holds since for $\alpha < \frac{c_\beta}{\mu}$,

$$\begin{aligned}(2+4\kappa^2(\mu\alpha-c_\beta))[F(x_0)-F(x_T)] &= (2-1/9+4\kappa^2\mu\alpha)[F(x_0)-F(x_T)] \\ &\le (2-1/9+4\kappa^2\mu\alpha)[F(x_0)-F^*] \\ &\le 2[F(x_0)-F^*].\end{aligned}$$

Note that the number of samples required for Algorithm 3 is $\mathcal{O}(T(2S+2D/Q))$. By Remark 2.9, if we choose $\alpha T = \mathcal{O}(\varepsilon^{-2})$ to guarantee an ε-stationary point, the sample complexity is $\mathcal{O}(\varepsilon^{-3})$. □

A.2.4 Stochastic Compositional Optimization

The last example is the stochastic compositional optimization defined in (3.18). We provide a detailed proof of Lemma 3.22 and Theorem 3.23 to show the $\mathcal{O}(\varepsilon^{-3})$ sample complexity of Algorithm 4 in this section.

Before proving Lemma 3.22, we first verify that the biased oracle $\nabla f_2(x_t;\tilde{\zeta}_t)^\top \nabla f_1(y_t;\xi_t)$ satisfies Assumption 3.1, i.e. bounded variance condition (3.1) and average smoothness condition (3.2). This allows us to apply the general analysis in Chapter 2.

The bounded variance condition holds with constant $\sigma^2 := 2M_{f_2}^2\sigma_{f_1}^2 + 2M_{f_1}^2\sigma_{f_2'}^2$ as shown below.

$$\begin{aligned}
&\mathbb{E}_{\tilde{\zeta},\xi}\|\nabla f_2(x;\tilde{\zeta})^\top \nabla f_1(y;\xi) - \nabla F_2(x)^\top \nabla F_1(y)\|^2 \\
&= \mathbb{E}_{\tilde{\zeta},\xi}\|\nabla f_2(x;\tilde{\zeta})^\top (\nabla f_1(y;\xi) - \nabla F_1(y)) + (\nabla f_2(x;\tilde{\zeta}) - \nabla F_2(x))^\top \nabla F_1(y)\|^2 \\
&\le 2\mathbb{E}_{\tilde{\zeta}}\|\nabla f_2(x;\tilde{\zeta})\|^2 \cdot \mathbb{E}_{\xi}\|\nabla f_1(y;\xi) - \nabla F_1(y)\|^2 + 2\mathbb{E}_{\tilde{\zeta}}\|\nabla f_2(x;\tilde{\zeta}) - \nabla F_2(x)\|^2 \cdot \|\nabla F_1(y)\|^2 \\
&\le 2M_{f_2}^2\sigma_{f_1}^2 + 2M_{f_1}^2\sigma_{f_2'}^2 := \sigma^2,
\end{aligned} \tag{A.16}$$

since $f_2(x;\tilde{\zeta})$ is M_{f_2}-Lipschitz and $F_1(y)$ is M_{f_1}-Lipschitz by Assumption 3.21. The average smoothness condition can be proved in a similar way.

$$\begin{aligned}
&\mathbb{E}_{\tilde{\zeta},\xi}\|\nabla f_2(x_1;\tilde{\zeta})^\top \nabla f_1(y_1;\xi) - \nabla f_2(x_2;\tilde{\zeta})^\top \nabla f_1(y_2;\xi)\|^2 \\
&= \mathbb{E}_{\tilde{\zeta},\xi}\|\nabla f_2(x_1;\tilde{\zeta})^\top (\nabla f_1(y_1;\xi) - \nabla f_1(y_2;\xi)) + (\nabla f_2(x_1;\tilde{\zeta}) - \nabla f_2(x_2;\tilde{\zeta}))^\top \nabla f_1(y_2;\xi)\|^2 \\
&\le 2\mathbb{E}_{\tilde{\zeta}}\|\nabla f_2(x_1;\tilde{\zeta})\|^2 \cdot \mathbb{E}_{\xi}\|\nabla f_1(y_1;\xi) - \nabla f_1(y_2;\xi)\|^2 \\
&\quad + 2\mathbb{E}_{\tilde{\zeta}}\|\nabla f_2(x_1;\tilde{\zeta}) - \nabla f_2(x_2;\tilde{\zeta})\|^2 \cdot \mathbb{E}_{\xi}\|\nabla f_1(y_2;\xi)\|^2 \\
&\le 2M_{f_2}^2 L_{f_1}^2\|y_1 - y_2\|^2 + 2M_{f_1}^2 L_{f_2}^2\|x_1 - x_2\|^2,
\end{aligned} \tag{A.17}$$

with all constants defined in Assumption 3.21. Now we give the proof of Lemma 3.22.

Proof (Lemma 3.22) We use h_t to estimate $\nabla F_2(x_t)^\top \nabla F_1(y_t)$ via $\nabla f_2(x_t;\tilde{\zeta}_t)^\top \nabla f_1(y_t;\xi_t)$ by the general framework. Let $A_t = \mathbb{E}\|h_t - \nabla F_2(x_t)^\top \nabla F_1(y_t)\|^2$ be the estimation error. Note that $\mathbb{E}_{\tilde{\zeta}_t,\xi_t}[\nabla f_2(x_t;\tilde{\zeta}_t)^\top \nabla f_1(y_t;\xi_t)] = \nabla F_2(x_t)^\top \nabla F_1(y_t)$, similar to (A.2) in the proof of Lemma 2.5, by (A.16) and (A.17), we obtain

$$\begin{aligned}
A_t &\le (1-\eta)A_{t-1} + 2\sigma^2\frac{\eta^2}{S} + \frac{2}{S}\mathbb{E}\|\nabla f_2(x_t;\tilde{\zeta}_t)^\top \nabla f_1(y_t;\xi_t) - \nabla f_2(x_{t-1};\tilde{\zeta}_t)^\top \nabla f_1(y_{t-1};\xi_t)\|^2 \\
&\le (1-\eta)A_{t-1} + 2\sigma^2\frac{\eta^2}{S} + \frac{4M_{f_2}^2 L_{f_1}^2}{S}\mathbb{E}\|y_t - y_{t-1}\|^2 + \frac{4M_{f_1}^2 L_{f_2}^2}{S}\mathbb{E}\|x_t - x_{t-1}\|^2.
\end{aligned} \tag{A.18}$$

Then we analyze the term $\mathbb{E}\|y_t - y_{t-1}\|^2$. The equivalent term in bilevel and minimax

case is handled by lower-level descent. Here we bound it by the recursive update of y.

$$\begin{aligned}
\mathbb{E}\|y_t - y_{t-1}\|^2 &\overset{(a)}{=} \mathbb{E}\left\|\frac{1}{S}\sum_{i=1}^{S} f_2(x_t;\zeta_t^i) - \eta y_{t-1} - (1-\eta)\frac{1}{S}\sum_{i=1}^{S} f_2(x_{t-1};\zeta_t^i)\right\|^2 \\
&= \mathbb{E}\left\|\frac{1}{S}\sum_{i=1}^{S}\Big(f_2(x_t;\zeta_t^i) - f_2(x_{t-1};\zeta_t^i)\Big) - \eta\left(y_{t-1} - \frac{1}{S}\sum_{i=1}^{S} f_2(x_{t-1};\zeta_t^i)\right)\right\|^2 \\
&\le \frac{2}{S^2}\mathbb{E}\left\|\sum_{i=1}^{S}\Big(f_2(x_t;\zeta_t^i) - f_2(x_{t-1};\zeta_t^i)\Big)\right\|^2 + 2\eta^2\mathbb{E}\left\|y_{t-1} - \frac{1}{S}\sum_{i=1}^{S} f_2(x_{t-1};\zeta_t^i)\right\|^2 \\
&\overset{(b)}{\le} \frac{2}{S}\sum_{i=1}^{S}\mathbb{E}\|f_2(x_t;\zeta_t^i) - f_2(x_{t-1};\zeta_t^i)\|^2 + 4\eta^2\mathbb{E}\|y_{t-1} - F_2(x_{t-1})\|^2 \\
&\quad + 4\eta^2\mathbb{E}\left\|F_2(x_{t-1}) - \frac{1}{S}\sum_{i=1}^{S} f_2(x_{t-1};\zeta_t^i)\right\|^2 \\
&\overset{(c)}{\le} 2M_{f_2}^2\mathbb{E}\|x_t - x_{t-1}\|^2 + \frac{4\eta^2}{S}\sigma_{f_2}^2 + 4\eta^2\mathbb{E}\|y_{t-1} - F_2(x_{t-1})\|^2,
\end{aligned}$$

where (a) follows from the update step of y_t in Algorithm 4, (b) holds by the inequality that $\|a_1 + a_2 + \cdots + a_k\|^2 \le k(\|a_1\|^2 + \|a_2\|^2 + \cdots \|a_k\|^2)$ which comes from Cauchy-Schwarz inequality, and (c) uses Assumption 3.21 that $f_2(x;\zeta_t^i)$ is M_{f_2}-Lipschitz continuous. Plugging the bound for $\mathbb{E}\|y_t - y_{t-1}\|^2$ back into (A.18), we obtain

$$\begin{aligned}
A_t &\le (1-\eta)A_{t-1} + 2(\sigma^2 + 8M_{f_2}^2L_{f_1}^2\sigma_{f_2}^2/S)\frac{\eta^2}{S} + (4M_{f_2}^4L_{f_1}^2 + 2M_{f_1}^2L_{f_2}^2)\frac{2}{S}\mathbb{E}\|x_t - x_{t-1}\|^2 \\
&\quad + 16M_{f_2}^2L_{f_1}^2\frac{\eta^2}{S}\mathbb{E}\|y_{t-1} - F_2(x_{t-1})\|^2 \\
&= (1-\eta)A_{t-1} + 2(\sigma^2 + 8M_{f_2}^2L_{f_1}^2\sigma_{f_2}^2/S)\frac{\eta^2}{S} + 2\ell_f^2\frac{\alpha^2}{S}\mathbb{E}\|h_{t-1}\|^2 + 16M_{f_2}^2L_{f_1}^2\frac{\eta^2}{S}B_{t-1},
\end{aligned}$$

where we use $x_{t+1} = x_t - \alpha h_t$, $B_t = \mathbb{E}\|y_t - \nabla F_2(x_t)\|$ and let $\ell_f^2 := 4M_{f_2}^4L_{f_1}^2 + 2M_{f_1}^2L_{f_2}^2$ for notation simplicity. The equation above is almost the same as Lemma 2.5, but has one additional term in the order of $\eta^2 B_{t-1}$. We then show this term does not affect too much.

When $\eta = 0$, by Lemma 2.6 or Lemma A.1 in Appendix A.1, we have

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \le \frac{\sigma^2}{D} + 2\ell_f^2\frac{Q\alpha^2}{S}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2,$$

thus by parameter selection rule (i) in Theorem 2.7, we obtain that

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \le \frac{\sigma^2}{\alpha T} + \rho_A\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2.$$

When $\eta \neq 0$, by Lemma 2.6 or Lemma A.2 in Appendix A.1, we have

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1} A_t &\le \frac{1}{c_\eta\alpha D}\cdot\frac{\sigma^2}{\alpha T} + \frac{2c_\eta}{S}\sum_{t=0}^{T-1}\alpha^3\cdot\frac{\sigma^2 + 8M_{f_2}^2L_{f_1}^2\sigma_{f_2}^2/S}{\alpha T} + \frac{2\ell_f^2}{c_\eta S}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|h_t\|^2 \\
&\quad + 16M_{f_2}^2L_{f_1}^2\frac{c_\eta\alpha^2}{S}\cdot\frac{1}{T}\sum_{t=0}^{T-1}B_t,
\end{aligned}$$

and then by parameter selection rule (ii) in Theorem 2.7, we obtain that

$$\frac{1}{T}\sum_{t=0}^{T-1} A_t \le \frac{\sigma^2 + 8M_{f_2}^2 L_{f_1}^2 \sigma_{f_2}^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2 + \frac{4M_{f_2}^2 L_{f_1}^2}{\alpha T} \cdot \frac{1}{T}\sum_{t=0}^{T-1} B_t, \tag{A.19}$$

since $c_\eta \alpha^2 / S = 1/(4\alpha T)$.

Then we analyse B_t, i.e. the estimation error of y_t, in a similar way. We use y_t to estimate $F_2(x_t)$ via $f_2(x_t;\zeta_t)$. Under Assumption 3.21, similar to (A.2) in the proof of Lemma 2.5, the error term satisfies

$$\begin{aligned} B_t &\le (1-\eta)B_{t-1} + \frac{2\eta^2\sigma_{f_2}^2}{S} + \frac{2}{S}\mathbb{E}\|f_2(x_t;\zeta) - f_2(x_{t-1};\zeta)\|^2 \\ &\le (1-\eta)B_{t-1} + \frac{2\eta^2\sigma_{f_2}^2}{S} + 2M_{f_2}^2\frac{\alpha^2}{S}\mathbb{E}\|h_{t-1}\|^2. \end{aligned}$$

Note that we use the same set of parameters for the estimation of $F_2(x_t)$ and $\nabla F_2(x_t)^\top \nabla F_1(y_t)$ for simplicity. By Lemma 2.6 and the parameter selection rules in Theorem 2.7, we obtain that

$$\frac{1}{T}\sum_{t=0}^{T-1} B_t \le \frac{\sigma_{f_2}^2}{\alpha T} + \rho_A \frac{M_{f_2}^2}{\ell_f^2} \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2.$$

Without loss of generality, when T is large enough, we have $\alpha T \ge \mathcal{O}(T^{2/3}) \ge 1$. Replacing $\frac{1}{T}\sum_{t=0}^{T-1} B_t$ in (A.19) by the above inequality, we have

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1} A_t &\le \frac{\sigma^2 + 8M_{f_2}^2 L_{f_1}^2 \sigma_{f_2}^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2 + 4M_{f_2}^2 L_{f_1}^2 \cdot \frac{1}{T}\sum_{t=0}^{T-1} B_t \\ &\le \frac{\sigma^2 + 12M_{f_2}^2 L_{f_1}^2 \sigma_{f_2}^2}{\alpha T} + \rho_A \left(1 + \frac{4M_{f_2}^4 L_{f_1}^2}{\ell_f^2}\right) \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2 \\ &\le \frac{\sigma^2 + 12M_{f_2}^2 L_{f_1}^2 \sigma_{f_2}^2}{\alpha T} + 2\rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2, \end{aligned}$$

since $\ell_f^2 = 4M_{f_2}^4 L_{f_1}^2 + 2M_{f_1}^2 L_{f_2}^2$. Thus the proof is complete. □

Lemma 3.22 is a direct consequence of Lemma 2.6 and Theorem 2.7. Then we can prove Theorem 3.23 and explain how we select ρ_A.

Proof (Theorem 3.23) With Assumption 3.21, it is easy to verify [30, Appendix A] that $F(x)$ is L_F-smooth with $L_F = M_{f_2}^2 L_{f_1} + M_{f_1} L_{f_2}$. Similar to (3.3) or the proof of Theorem 3.2, we obtain

$$\begin{aligned} \mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 &\le \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2 + \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t - \nabla F(x_t)\|^2 \\ &\le \frac{2[F(x_0) - F^*]}{\alpha T} - \frac{2}{3T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2 + \frac{2}{T}\sum_{t=0}^{T-1} A_t + 2M_{f_2}^2 L_{f_1}^2 \frac{1}{T}\sum_{t=0}^{T-1} B_t, \end{aligned}$$

by (3.19). The bound for A_t has two cases in Lemma 3.22, and we use the second one since it is slightly worse. By results in Lemma 3.22, we obtain

$$\begin{aligned} \mathbb{E}_\tau\|\nabla F(x_\tau)\|^2 \le{}& \frac{2[F(x_0) - F^*] + 2\sigma^2 + 26M_{f_2}^2 L_{f_1}^2 \sigma_{f_2}^2}{\alpha T} \\ &- \left(\frac{2}{3} - 4\rho_A - \frac{2M_{f_2}^4 L_{f_1}^2}{\ell_f^2}\rho_A\right) \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t\|^2. \end{aligned}$$

If we choose ρ_A in Lemma 3.22 as $\rho_A = \min\left\{\frac{1}{12}, \frac{\ell_f^2}{6M_{f_2}^4 L_{f_1}^2}\right\} = \frac{1}{12}$ since $\ell_f^2 \geq 4M_{f_2}^4 L_{f_1}^2$, we have

$$\frac{2}{3} - 4\rho_A - \frac{2M_{f_2}^4 L_{f_1}^2}{\ell_f^2}\rho_A \geq 0,$$

and the proof is complete. Note that the total number of samples required for Algorithm 4 is $\mathcal{O}(T(3S + 3D/Q))$, which is $\mathcal{O}(\varepsilon^{-3})$ by setting $\alpha T = \mathcal{O}(\varepsilon^{-2})$ to achieve an ε-stationary point by similar arguments in Remark 2.9. □

A.3 Proofs of Results in Chapter 4

We provide proofs of Lemma 4.6 and 4.7 for completeness. We first state the concept of the optimaity condition for reference. For constrained convex optimization $\min_{x\in\mathcal{X}} f(x)$, x^* is an optimal solution if and only if the following holds true:

$$\nabla f(x^*)^\top (x - x^*) \geq 0, \quad \forall x \in \mathcal{X}. \tag{A.20}$$

If $f(x)$ is not differentiable, the above also holds by replacing $\nabla f(x^*)$ with some subgradient $g^* \in \partial f(x^*)$.

Proof (Lemma 4.6) The optimization problem in the update (4.2) is convex. Then by its optimality condition as shown in (A.20) and Definition 4.1 of Bregman distance, for a subgradient $g_{t+1} \in \partial\, r(x_{t+1})$ and $\forall u \in \mathcal{X}$, we have

$$\Big(g_{t+1} + h_t + \frac{1}{\alpha}[\nabla\omega(x_{t+1}) - \nabla\omega(x_t)]\Big)^\top (u - x_{t+1}) \geq 0.$$

Let $u = x_t$ in the above inequality, we obtain

$$\begin{aligned} h_t^\top (x_t - x_{t+1}) &\geq g_{t+1}^\top (x_{t+1} - x_t) + \frac{1}{\alpha}\Big(\nabla\omega(x_{t+1}) - \nabla\omega(x_t)\Big)^\top (x_{t+1} - x_t) \\ &\geq r(x_{t+1}) - r(x_t) + \frac{\mu}{\alpha}\|x_{t+1} - x_t\|^2, \end{aligned}$$

where the last inequality holds since $r(x)$ is convex and $\omega(x)$ is μ-strongly convex. Dividing both sides of the above inequality by α, we have

$$h_t^\top \mathcal{G}_\alpha(x_t, h_t) \geq \frac{1}{\alpha}\Big(r(x_{t+1}) - r(x_t)\Big) + \mu\|\mathcal{G}_\alpha(x_t, h_t)\|^2,$$

where $\mathcal{G}_\alpha(x_t, h_t) = \frac{1}{\alpha}(x_t - x_{t+1})$ is the gradient mapping. □

Proof (Lemma 4.7) By the definition of $\mathcal{G}_\alpha(x_t, h_t)$ and $\mathcal{G}_\alpha(x_t)$ in (4.3) and (4.4), we have

$$\mathcal{G}_\alpha(x_t, h_t) - \mathcal{G}_\alpha(x_t) = \frac{1}{\alpha}(x_t - x_{t+1}) - \frac{1}{\alpha}(x_t - x_t^+) = \frac{1}{\alpha}(x_t^+ - x_{t+1}).$$

By the optimality condition of (4.2), for any $u \in \mathcal{X}$ and some subgradient $g_{t+1} \in \partial\, r(x_{t+1})$ we have

$$\begin{aligned} h_t^\top (u - x_{t+1}) &\geq g_{t+1}^\top (x_{t+1} - u) + \frac{1}{\alpha}\Big(\nabla\omega(x_{t+1}) - \nabla\omega(x_t)\Big)^\top (x_{t+1} - u) \\ &\geq r(x_{t+1}) - r(u) + \frac{1}{\alpha}\Big(\nabla\omega(x_{t+1}) - \nabla\omega(x_t)\Big)^\top (x_{t+1} - u), \end{aligned}$$

where the last inequality follows from the convexity of $r(x)$. Setting $u = x_t^+$ in the above inequality, we obtain

$$h_t^\top (x_t^+ - x_{t+1}) \geq r(x_{t+1}) - r(x_t^+) + \frac{1}{\alpha}\Big(\nabla\omega(x_{t+1}) - \nabla\omega(x_t)\Big)^\top (x_{t+1} - x_t^+). \tag{A.21}$$

By the optimality condition of (4.5), for any $u \in \mathcal{X}$ and some subgradient $g_t^+ \in \partial r(x_t^+)$ we have

$$\begin{aligned}\nabla F(x_t)^\top (u - x_t^+) &\geq (g_t^+)^\top (x_t^+ - u) + \frac{1}{\alpha}\Big(\nabla\omega(x_t^+) - \nabla\omega(x_t)\Big)^\top (x_t^+ - u)\\ &\geq r(x_t^+) - r(u) + \frac{1}{\alpha}\Big(\nabla\omega(x_t^+) - \nabla\omega(x_t)\Big)^\top (x_t^+ - u),\end{aligned}$$

where the last inequality follows from the convexity of $r(x)$. Setting $u = x_{t+1}$ in the above inequality, we obtain

$$\nabla F(x_t)^\top (x_{t+1} - x_t^+) \geq r(x_t^+) - r(x_{t+1}) + \frac{1}{\alpha}\Big(\nabla\omega(x_t^+) - \nabla\omega(x_t)\Big)^\top (x_t^+ - x_{t+1}). \tag{A.22}$$

Summing up (A.21) and (A.22), we have

$$\begin{aligned}(\nabla F(x_t) - h_t)^\top (x_{t+1} - x_t^+) &\geq \frac{1}{\alpha}\Big(\nabla\omega(x_{t+1}) - \nabla\omega(x_t^+)\Big)^\top (x_{t+1} - x_t^+)\\ &\geq \frac{\mu}{\alpha}\|x_{t+1} - x_t^+\|^2,\end{aligned}$$

since $\omega(x)$ is μ-strongly convex. Then by Cauchy-Schwarz inequality,

$$\frac{\mu}{\alpha}\|x_{t+1} - x_t^+\|^2 \leq \|h_t - \nabla F(x_t)\|\,\|x_{t+1} - x_t^+\|.$$

Since $x_{t+1} \neq x_t^+$ in general, we know

$$\frac{\mu}{\alpha}\|x_{t+1} - x_t^+\| \leq \|h_t - \nabla F(x_t)\|,$$

and that

$$\|\mathcal{G}_\alpha(x_t, h_t) - \mathcal{G}_\alpha(x_t)\|^2 = \frac{1}{\alpha^2}\|x_{t+1} - x_t^+\|^2 \leq \frac{1}{\mu^2}\|h_t - \nabla F(x_t)\|^2. \qquad \square$$

With the help of the Lemma 4.6 and 4.7, we show the proof of Theorem 4.8 below.

Proof (Theorem 4.8) Since $F(x)$ is L_F-smooth by Assumption 4.4, we have

$$\begin{aligned}F(x_{t+1}) &\leq F(x_t) + \nabla F(x_t)^\top (x_{t+1} - x_t) + \frac{L_F}{2}\|x_{t+1} - x_t\|^2\\ &\overset{(a)}{=} F(x_t) - \alpha\nabla F(x_t)^\top \mathcal{G}_\alpha(x_t, h_t) + \frac{L_F\alpha^2}{2}\|\mathcal{G}_\alpha(x_t, h_t)\|^2\\ &= F(x_t) - \alpha h_t^\top \mathcal{G}_\alpha(x_t, h_t) + \alpha(h_t - \nabla F(x_t))^\top \mathcal{G}_\alpha(x_t, h_t) + \frac{L_F\alpha^2}{2}\|\mathcal{G}_\alpha(x_t, h_t)\|^2\\ &\overset{(b)}{\leq} F(x_t) - r(x_{t+1}) + r(x_t) - \frac{\alpha}{2}(2\mu - \alpha L_F)\|\mathcal{G}_\alpha(x_t, h_t)\|^2 + \alpha(h_t - \nabla F(x_t))^\top \mathcal{G}_\alpha(x_t, h_t),\end{aligned}$$

where (a) follows from the definition of $\mathcal{G}_\alpha(x_t, h_t)$ in (4.3) and (b) holds by Lemma 4.6. Note that $\Phi(x) = F(x) + r(x)$ by (4.1), thus

$$\begin{aligned}\Phi(x_{t+1}) &\leq \Phi(x_t) - \frac{\alpha}{2}(2\mu - \alpha L_F)\|\mathcal{G}_\alpha(x_t, h_t)\|^2 + \alpha(h_t - \nabla F(x_t))^\top \mathcal{G}_\alpha(x_t, h_t)\\ &\leq \Phi(x_t) - \frac{\alpha}{2}(\mu - \alpha L_F)\|\mathcal{G}_\alpha(x_t, h_t)\|^2 + \frac{\alpha}{2\mu}\|h_t - \nabla F(x_t)\|^2,\end{aligned} \tag{A.23}$$

where in the last inequality we use Young's inequality that for any $\gamma > 0$,

$$2(h_t - \nabla F(x_t))^\top \mathcal{G}_\alpha(x_t, h_t) \leq \gamma \|\mathcal{G}_\alpha(x_t, h_t)\|^2 + \frac{1}{\gamma}\|h_t - \nabla F(x_t)\|^2,$$

and we choose $\gamma = \mu > 0$. Rearranging terms of (A.23) and taking full expectations, we have

$$\frac{\alpha}{2}(\mu - \alpha L_F)\mathbb{E}\|\mathcal{G}_\alpha(x_t, h_t)\|^2 \leq \mathbb{E}[\Phi(x_t) - \Phi(x_{t+1})] + \frac{\alpha}{2\mu}\mathbb{E}\|h_t - \nabla F(x_t)\|^2.$$

Summing up the above inequality from $t = 0$ to $T - 1$ and then dividing both sides by $\alpha T/2$, we obtain

$$(\mu - \alpha L_F) \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\mathcal{G}_\alpha(x_t, h_t)\|^2 \leq \frac{2[\Phi(x_0) - \Phi^*]}{\alpha T} + \frac{1}{\mu T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t - \nabla F(x_t)\|^2, \quad \text{(A.24)}$$

since $\Phi(x_T) \geq \Phi^*$. We use h_t to estimate $\nabla F(x_t)$ by the general variance reduced framework, thus by Theorem 2.7 in Chapter 2 with the constant ρ_A to be determined, we have that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|h_t - \nabla F(x_t)\|^2 \leq \frac{\sigma^2}{\alpha T} + \rho_A \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\mathcal{G}_\alpha(x_t, h_t))\|^2. \quad \text{(A.25)}$$

Plugging (A.25) into (A.24), we obtain

$$\left(\mu - \alpha L_F - \frac{\rho_A}{\mu}\right) \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\mathcal{G}_\alpha(x_t, h_t)\|^2 \leq \frac{2[\Phi(x_0) - \Phi^*] + \sigma^2/\mu}{\alpha T}.$$

With the choice that $\alpha \leq \mu/(3L_F)$ and $\rho_A = \mu^2/3$ such that $\mu - \alpha L_F - \rho_A/\mu \geq \mu/3$, we then obtain that

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\mathcal{G}_\alpha(x_t, h_t)\|^2 \leq \frac{6[\Phi(x_0) - \Phi^*]/\mu + 3\sigma^2/\mu^2}{\alpha T}. \quad \text{(A.26)}$$

What we need to bound is $\mathbb{E}_\tau\|\mathcal{G}_\alpha(x_\tau)\|^2 = \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\mathcal{G}_\alpha(x_\tau)\|^2$. By Young's inequality,

$$\begin{aligned}
\|\mathcal{G}_\alpha(x_t)\|^2 &= \|(\mathcal{G}_\alpha(x_t) - \mathcal{G}_\alpha(x_t, h_t)) + \mathcal{G}_\alpha(x_t, h_t)\|^2 \\
&\leq 2\|\mathcal{G}_\alpha(x_t) - \mathcal{G}_\alpha(x_t, h_t)\|^2 + 2\|\mathcal{G}_\alpha(x_t, h_t)\|^2 \\
&\leq 2\|\mathcal{G}_\alpha(x_t, h_t)\|^2 + \frac{2}{\mu^2}\|h_t - \nabla F(x_t)\|^2,
\end{aligned}$$

where the last inequality follows from Lemma 4.7. Thus

$$\begin{aligned}
\mathbb{E}_\tau\|\mathcal{G}_\alpha(x_\tau)\|^2 &\leq 2 \cdot \frac{1}{T}\sum_{t=0}^{T-1} \|\mathcal{G}_\alpha(x_t, h_t)\|^2 + \frac{2}{\mu^2} \cdot \frac{1}{T}\sum_{t=0}^{T-1} \|h_t - \nabla F(x_t)\|^2 \\
&\leq \frac{2\sigma^2}{\mu^2 \alpha T} + \left(\frac{2\rho_A}{\mu^2} + 2\right) \cdot \frac{1}{T}\sum_{t=0}^{T-1} \|\mathcal{G}_\alpha(x_t, h_t)\|^2 \\
&\leq \frac{2\sigma^2}{\mu^2 \alpha T} + \left(\frac{2\rho_A}{\mu^2} + 2\right) \cdot \frac{6[\Phi(x_0) - \Phi^*]/\mu + 3\sigma^2/\mu^2}{\alpha T} \\
&= \frac{16[\Phi(x_0) - \Phi^*]/\mu + 10\sigma^2/\mu^2}{\alpha T},
\end{aligned}$$

by (A.25), (A.26) and the choice that $\rho_A = \mu^2/3$. □

Bibliography

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[4] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[5] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

[6] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[7] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 689–699, 2018.

[8] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. In *Advances in Neural Information Processing Systems*, pages 2406–2416, 2019.

[9] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2613–2621. JMLR. org, 2017.

[10] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in Neural Information Processing Systems*, 32:15236–15245, 2019.

[11] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

[12] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[13] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[14] Qi Qian, Shenghuo Zhu, Jiasheng Tang, Rong Jin, Baigui Sun, and Hao Li. Robust optimization over multiple domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4739–4746, 2019.

[15] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 113–124, 2019.

[16] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR, 2019.

[17] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

[18] Tianyi Lin, Chi Jin, and Michael Jordan. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, pages 6083–6093. PMLR, 2020.

[19] Mengdi Wang, Ethan X Fang, and Han Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2):419–449, 2017.

[20] Yifan Hu, Xin Chen, and Niao He. Sample complexity of sample average approximation for conditional stochastic optimization. *SIAM Journal on Optimization*, 30(3):2103–2133, 2020.

[21] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

[22] Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory*, pages 2738–2779. PMLR, 2020.

[23] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[24] Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor J Hastie. 1-norm support vector machines. In *Advances in neural information processing systems*, page None. Citeseer, 2003.

[25] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[26] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, pages 6286–6295. PMLR, 2021.

[27] Bin Hu, Peter Seiler, and Laurent Lessard. Analysis of biased stochastic gradient descent using sequential semidefinite programs. *Mathematical Programming*, pages 1–26, 2020.

[28] Ahmad Ajalloeian and Sebastian U Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

[29] Belhal Karimi, Blazej Miasojedow, Eric Moulines, and Hoi-To Wai. Non-asymptotic analysis of biased stochastic approximation scheme. In *Conference on Learning Theory*, pages 1944–1974, 2019.

[30] Junyu Zhang and Lin Xiao. A stochastic composite gradient method with incremental variance reduction. *Advances in Neural Information Processing Systems*, 32:9078–9088, 2019.

[31] Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33, 2020.

[32] Tianyi Chen, Yuejiao Sun, and Wotao Yin. A single-timescale stochastic bilevel optimization method. *arXiv preprint arXiv:2102.04671*, 2021.

[33] Tengyu Xu, Zhe Wang, Yingbin Liang, and H Vincent Poor. Gradient free minimax optimization: Variance reduction and faster convergence. *arXiv preprint arXiv:2006.09361*, 2020.

[34] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Accelerated zeroth-order and first-order momentum methods from mini to minimax optimization. *arXiv preprint arXiv:2008.08170*, 2020.

[35] Tianyi Chen, Yuejiao Sun, and Wotao Yin. Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization. *IEEE Transactions on Signal Processing*, 2021.

[36] Junyu Zhang and Lin Xiao. Multilevel composite stochastic optimization via nested variance reduction. *SIAM Journal on Optimization*, 31(2):1131–1157, 2021.

[37] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.

[38] Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5569–5579, 2018.

[39] Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

[40] Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1195–1204. PMLR, 2019.

[41] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[42] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1082–1092. PMLR, 2020.

[43] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

[44] Sanjeev Arora, Simon Du, Sham Kakade, Yuping Luo, and Nikunj Saunshi. Provable representation learning for imitation learning via bi-level optimization. In *International Conference on Machine Learning*, pages 367–376. PMLR, 2020.

[45] Zalán Borsos, Mojmír Mutnỳ, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems 33*, 2020.

[46] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning*, pages 4882–4892. PMLR, 2021.

[47] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A momentum-assisted single-timescale stochastic approximation algorithm for bilevel optimization. *arXiv e-prints*, pages arXiv–2102, 2021.

[48] Tianyi Chen, Yuejiao Sun, and Wotao Yin. Tighter analysis of alternating stochastic gradient method for stochastic nested problems. *arXiv preprint arXiv:2106.13781*, 2021.

[49] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *arXiv preprint arXiv:2102.07367*, 2021.

[50] Zhishuai Guo and Tianbao Yang. Randomized stochastic variance-reduced methods for stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*, 2021.

[51] Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *arXiv preprint arXiv:2106.04692*, 2021.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[53] Soroosh Shafieezadeh Abadeh, Peyman M Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. *Advances in Neural Information Processing Systems*, 28:1576–1584, 2015.

[54] Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *NIPS*, volume 29, pages 2208–2216, 2016.

[55] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050:9, 2017.

[56] Hoi To Wai, Zhuoran Yang, Mingyi Hong, and Zhaoran Wang. Multi-agent reinforcement learning via double averaging primal-dual optimization. *Advances in Neural Information Processing Systems*, 2018:9649–9660, 2018.

[57] Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International Conference on Machine Learning*, pages 1125–1134. PMLR, 2018.

[58] Paul Tseng. On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 60(1-2):237–252, 1995.

[59] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020.

[60] Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[61] Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2):319–344, 2007.

[62] TaeHo Yoon and Ernest K Ryu. Accelerated algorithms for smooth convex-concave minimax problems with o (1/kˆ 2) rate on squared gradient norm. In *International Conference on Machine Learning*, pages 12098–12109. PMLR, 2021.

[63] H Rafique, M Liu, Q Lin, and T Yang. Non-convex min–max optimization: provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 1810.

[64] Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. *arXiv preprint arXiv:1902.00618*, 2019.

[65] John M Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966.

[66] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

[67] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

[68] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476. PMLR, 2018.

[69] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

[70] Qi Cai, Mingyi Hong, Yongxin Chen, and Zhaoran Wang. On the global convergence of imitation learning: A case for linear quadratic regulator. *arXiv preprint arXiv:1901.03674*, 2019.

[71] Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *arXiv preprint arXiv:1902.08297*, 2019.

[72] Jiahao Xie, Chao Zhang, Yunsong Zhang, Zebang Shen, and Hui Qian. A federated learning framework for nonconvex-pl minimax problems. *arXiv preprint arXiv:2105.14216*, 2021.

[73] Junchi Yang, Negar Kiyavash, and Niao He. Global convergence and variance-reduced optimization for a class of nonconvex-nonconcave minimax problems. *arXiv preprint arXiv:2002.09621*, 2020.

[74] Darinka Dentcheva, Spiridon Penev, and Andrzej Ruszczyński. Statistical estimation of composite risk functionals and risk optimization problems. *Annals of the Institute of Statistical Mathematics*, 69(4):737–760, 2017.

[75] Christoph Dann, Gerhard Neumann, Jan Peters, et al. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.

[76] Mengdi Wang, Ji Liu, and Ethan X Fang. Accelerating stochastic composition optimization. *Journal of Machine Learning Research*, 2017.

[77] Saeed Ghadimi, Andrzej Ruszczynski, and Mengdi Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

[78] Wenqing Hu, Chris Junchi Li, Xiangru Lian, Ji Liu, and Huizhuo Yuan. Efficient smooth non-convex stochastic compositional optimization via stochastic recursive gradient descent. *Advances in Neural Information Processing Systems*, 32:6929–6937, 2019.

[79] Jun Liu, Jianhui Chen, and Jieping Ye. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 547–556, 2009.

[80] Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in neural information processing systems*, 29:1145–1153, 2016.

[81] Wenjie Li, Zhanyu Wang, Yichen Zhang, and Guang Cheng. Variance reduction on adaptive stochastic mirror descent. *arXiv preprint arXiv:2012.13760*, 2020.

[82] Aaron Defazio and Leon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *Advances in Neural Information Processing Systems*, 32:1755–1765, 2019.

[83] Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, pages 148–156. PMLR, 2017.

[84] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[85] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[86] David Harrison Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

[87] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[88] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[89] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[90] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Acknowledgements

This work was done within the Optimization and Decision Intelligence Group at the Department of Computer Science of ETH Zurich. First of all, I would like to thank my supervisor Prof. Dr. Niao He for giving me this great opportunity to work with the group. She provided many insightful suggestions and excellently guided me to delve into this interesting research area. Most of the work was also co-supervised by one of the PhD students in the group, Yifan Hu. He first brought up this topic and provided me with a lot of help in understanding the area. The ideas of the thesis came from discussions with Prof. Niao He and Yifan Hu during our weekly meetings. I am grateful to their support and time. The main part of the thesis was based on our paper "The All-in-one Recipe for Structured Nonconvex Smooth Stochastic Optimization" which has been submitted to NeurIPS. I would also like to thank all the useful comments given by the reviewers of this paper. The suggestions helped to make the thesis in better shape.

The thesis was written between March and September in 2021, which was still a difficult time because of COVID-19. It was the company of my parents and girlfriend Jingpu Guo that supported me to get through this time and successfully complete the thesis. The daily chatting and video calls with my girlfriend gave me great comfort. Although we were in different countries, I could still feel her love and care. Finally I would like to thank my roommates and friends for their kind help during my life at Zurich. I am very lucky to meet them.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Variance Reduction for Non-Convex Stochastic Optimization: General Analysis and New Applications

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Zhang	Liang

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, on 09.09.2021	Liang Zhang

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.