ProxSARAH: An Efficient Algorithmic Framework for Stochastic Composite Nonconvex Optimization

Nhan H. Pham[†] nhanph@live.unc.edu

Lam M. Nguyen[‡] lamnguyen.mltd@ibm.com
Dzung T. Phan[‡] phandu@us.ibm.com
[‡]*IBM Research, Thomas J. Watson Research Center*
Yorktown Heights, NY USA

Quoc Tran-Dinh[†*] quoctd@email.unc.edu
[†]*Department of Statistics and Operations Research*
University of North Carolina at Chapel Hill, Chapel Hill, NC27599, USA.

Editor:

Abstract

In this paper, we propose a new stochastic algorithmic framework to solve stochastic composite nonconvex optimization problems that covers both finite-sum and expectation settings. Our algorithms rely on the SARAH estimator introduced in (Nguyen et al., 2017a) and consist of two steps: a proximal gradient step and an averaging step that are different from existing nonconvex proximal-type algorithms. The algorithms only require a smoothness assumption of the nonconvex objective term. In the finite-sum case, we show that our algorithm achieves optimal convergence rate by matching the lower-bound worst-case complexity, while in the expectation case, it attains the best-known convergence rate under only standard smoothness and bounded variance assumptions. One key step of our algorithms is a new constant step-size that helps to achieve desired convergence rate. Our step-size is much larger than existing methods including proximal SVRG schemes in the single sample case. We generalize our algorithm to mini-batches for both inner and outer loops, and adaptive step-sizes. We also specify the algorithm to the non-composite case that covers and dominates existing state-of-the-arts in terms of convergence rate. We test the proposed algorithms on two composite nonconvex optimization problems and feedforward neural networks using several well-known datasets.

Keywords: Stochastic proximal gradient descent; optimal convergence rate; composite nonconvex optimization; finite-sum minimization; expectation minimization.

1. Introduction

In this paper, we consider the following stochastic composite, nonconvex, and possibly nonsmooth optimization problem:

$$\min_{w\in\mathbb{R}^d}\Big\{F(w) := f(w) + \psi(w) \equiv \mathbb{E}\left[f(w;\xi)\right] + \psi(w)\Big\}, \tag{1}$$

where $f(w) := \mathbb{E}\left[f(w;\xi)\right]$ is the expectation of a stochastic function $f(w;\xi)$ depending on a random vector ξ in a given probability space $(\Omega, \mathbb{P})$, and $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper, closed, and convex function.

As a special case of (1), if ξ is a uniformly random vector defined on a finite support set $\Omega := \{\xi_1, \xi_2, \cdots, \xi_n\}$, then (1) reduces to the following composite nonconvex finite-sum minimization problem:

$$\min_{w\in\mathbb{R}^d} \Big\{ F(w) := f(w) + \psi(w) \equiv \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \psi(w) \Big\}, \tag{2}$$

where $f_i(w) := f(w;\xi_i)$ for $i = 1, \cdots, n$. Problem (2) is often referred to as a regularized empirical risk minimization in machine learning and finance.

Motivation: Problems (1) and (2) cover a broad range of applications in machine learning and statistics, especially in neural networks, see, e.g. (Bottou, 1998, 2010; Bottou et al., 2018; Goodfellow et al., 2016; Sra et al., 2012). Hitherto, state-of-the-art numerical optimization methods for solving these problems rely on stochastic approaches, see, e.g. (Johnson and Zhang, 2013; Schmidt et al., 2017; Shapiro et al., 2009; Defazio et al., 2014). In the convex case, both non-composite and composite settings (1) and (2) have been intensively studied with different schemes such as standard stochastic gradient (Robbins and Monro, 1951), proximal stochastic gradient (Ghadimi and Lan, 2013; Nemirovski et al., 2009), stochastic dual coordinate descent (Shalev-Shwartz and Zhang, 2013), variance reduction methods (Allen-Zhu, 2017a; Defazio et al., 2014; Johnson and Zhang, 2013; Nitanda, 2014; Schmidt et al., 2017; Xiao and Zhang, 2014), stochastic conditional gradient (Reddi et al., 2016a), and stochastic primal-dual methods (Chambolle et al., 2018). Thanks to variance reduction techniques, several efficient methods with constant step-sizes have been developed for convex settings that match the lower-bound worst-case complexity (Agarwal et al., 2010). However, methods for nonconvex settings are still limited and heavily focus on the non-composite form of (1) and (2), i.e. $\psi = 0$.

Theory and stochastic methods for nonconvex problems are still in progress and require substantial effort to obtain efficient algorithms with rigorous convergence guarantees. It is shown in (Fang et al., 2018) that there is still a gap between the upper-bound complexity in state-of-the-art methods and the lower-bound worst-case complexity for the nonconvex problem (2) under standard smoothness assumption. Motivated by this fact, we make an attempt to develop a new algorithmic framework that can reduce and at least close this gap in the composite finite-sum setting (2). Our algorithms rely on a recent biased stochastic estimator for the objective gradient, called SARAH, introduced in (Nguyen et al., 2017a) for convex problems.

Related work: In the nonconvex case, both problems (1) and (2) have been intensively studied in recent years with a vast number of research papers. While numerical algorithms for solving the non-composite setting, i.e. $\psi = 0$, are well-developed and have received considerable attention (Allen-Zhu, 2017b; Allen-Zhu and Li, 2018; Allen-Zhu and Yuan, 2016; Fang et al., 2018; Lihua et al., 2017; Nguyen et al., 2017b, 2018b, 2019; Reddi et al., 2016b; Zhou et al., 2018), methods for composite setting remain limited (Reddi et al., 2016b; Wang et al., 2018). In terms of algorithms, (Reddi et al., 2016b) studies a non-composite finite-sum problem as a special case of (2) using SVRG estimator from (Johnson and Zhang, 2013). Additionally, they extend their method to the composite setting by simply applying the proximal operator of ψ as in the well-known forward-backward scheme. Another related work using SVRG estimator can be found in (Li and Li, 2018). These algorithms have some limitation as will be discussed later. The same technique was applied in (Wang et al., 2018) to develop other variants for both (1) and (2), but using the SARAH estimator from (Nguyen et al., 2017a). The authors derive a constant large step-size, but at the same time control mini-batch size to achieve convergence. Consequently, it has an essential limitation as will also be discussed in Subsection 3.2.3. Both algorithms achieve suboptimal convergence rate with the same order. In (Reddi et al., 2016a), the authors propose a stochastic Frank-Wolfe method that can handle constraints as special cases of (2). Recently, a stochastic variance reduction method with momentum was studied in (Zhou et al., 2019) for solving (2) which can be viewed as a modification of SpiderBoost in (Wang et al., 2018).

Our algorithm remains a variance reduction stochastic method, but it is different from these works at two major points: an additional averaging step and different two constant step-sizes. Having two step-sizes allows us to flexibly trade-off them and develop an adaptive update rule. Note that our averaging step looks similar to the robust stochastic gradient method in (Nemirovski et al., 2009), but fundamentally different since it evaluates the proximal step at the averaging point. In fact, it is closely related to averaged fixed-point schemes in the literature, see, e.g. (Bauschke and Combettes, 2017).

In terms of theory, many researchers have focussed on theoretical aspects of existing algorithms. For example, (Ghadimi and Lan, 2013) appear as one of the first pioneering works studying convergence rates of stochastic gradient descent-type methods for nonconvex and non-composite finite-sum problems. They later extend it to the composite setting in (Ghadimi et al., 2016). (Wang et al., 2018) also investigate the gradient dominance case, and (Karimi et al., 2016) consider both finite-sum and composite finite-sum under different assumptions.

Whereas many researchers have been trying to improve complexity upper bounds of stochastic first-order methods using different techniques (Allen-Zhu, 2017b; Allen-Zhu and Li, 2018; Allen-Zhu and Yuan, 2016; Fang et al., 2018), other researchers attempt to construct examples for lower-bound complexity estimates. In the convex case, there exist numerous research papers including (Agarwal et al., 2010; Nemirovskii and Yudin, 1983; Nesterov, 2004). In (Fang et al., 2018; Zhou and Gu, 2019), the authors have constructed a lower-bound complexity for nonconvex finite-sum problem covered by (2). They show that the lower-bound complexity for any stochastic gradient method relied on only smoothness assumption to achieve an ε-stationary point in expectation is $\Omega\left(n^{1/2}\varepsilon^{-2}\right)$. For the expectation problem (1), the best-known complexity bound to achieve an ε-stationary point in expectation is $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right)$ as shown in (Fang et al., 2018; Wang et al., 2018), where σ is an upper bound of the variance (see Assumption 2.3). Unfortunately, we have not seen any lower-bound complexity for the nonconvex setting of (1) under standard assumptions in the literature.

Our approach and contribution: In this paper, we exploit the SARAH estimator, a biased stochastic recursive gradient estimator, in (Nguyen et al., 2017a), to design new proximal variance reduction stochastic gradient algorithms to solve both composite finite-sum (1) and expectation (2) problems. The SARAH algorithm is simply a double-loop stochastic gradient method with a flavor of SVRG (Johnson and Zhang, 2013), but using a novel biased estimator that is different from SVRG. SARAH is a recursive method as SAGA (Defazio et al., 2014), but can avoid the major issue of storing gradients as in SAGA. Our method will rely on the SARAH estimator combining with an averaging proximal-gradient scheme to solve both (1) and (2).

The contribution of this paper is a new algorithmic framework that covers different variants with optimal and best-known theoretical complexity bounds. More specifically, our main contribution can be summarized as follows:

(a) **Composite settings:** We propose a general stochastic variance reduction framework relying on the SARAH estimator to solve both finite-sum and expectation problems (2) and (1) in composite settings. We analyze our framework to design appropriate constant step-sizes instead of diminishing step-sizes as in standard stochastic gradient descent methods. As usual, the algorithm has double loops, where the outer loop can either take full gradient or mini-batch to reduce computational burden in large-scale and expectation settings. The inner loop requires single sample but can also work with mini-batch as an option. Our framework can be specified to cover adaptive step-size variants and non-composite settings.

(b) **Optimal and best-known complexity:** In the finite-sum setting (2), our method achieves $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$ complexity bound to attain an ε-stationary point in expectation under only the smoothness of f. This complexity matches the lower-bound worst-case complexity in (Fang et al., 2018; Zhou and Gu, 2019), and therefore, it is optimal. In the expectation setting (1), our algorithm requires $\mathcal{O}\left(\sigma\varepsilon^{-3}\right)$ iterations to achieve an ε-stationary point in expectation under only the smoothness of f and bounded variance σ^2. To the best of our knowledge, this is the best-known complexity so far for general problem (1) under standard assumptions.

Our result covers the non-composite setting with optimal rate in the finite-sum case (Nguyen et al., 2019), and appears to be better than the best-known complexity in (Fang et al., 2018; Wang et al., 2018) for the expectation problem (1). Since the composite setting covers a broader class of nonconvex problems including convex constraints, our method has better chance to handle new applications than non-composite methods. It also allows one to deal with neural network training problems with different regularizers such as sparsity or constraints on weights.

Comparison: Hitherto, we have found three different variance reduction algorithms of the stochastic proximal gradient method for nonconvex problems that are most related to our work: proximal SVRG (called ProxSVRG) in (Reddi et al., 2016b), ProxSVRG+ in (Li and Li, 2018), and ProxSpiderBoost in (Wang et al., 2018). Other methods such as proximal stochastic gradient descent (ProxSGD) scheme (Ghadimi et al., 2016), ProxSAGA in (Reddi et al., 2016b), and Natasha variants in (Allen-Zhu, 2017b) are quite different and already intensively compared in previous works (Li and Li, 2018; Reddi et al., 2016b; Wang et al., 2018), and hence we do not including them here.

In terms of theory, Table 1 compares different methods for solving (1) and (2) regarding the stochastic first-order oracle calls (SFO), the applicability to finite-sum and/or expectation and composite settings, step-sizes, and the use of mini-batch.

Algorithms	Finite-sum	Expectation	Composite	Step-size	Mini-batch
GD (Nesterov, 2004)	$\mathcal{O}\left(n\varepsilon^{-2}\right)$	NA	Yes	$\mathcal{O}\left(L^{-1}\right)$	Full/required
SGD (Ghadimi and Lan, 2013)	NA	$\mathcal{O}\left(\sigma^2\varepsilon^{-4}\right)$	Yes	$\mathcal{O}\left(L^{-1}\right)$	Optional
SVRG/SAGA (Reddi et al., 2016b)	$\mathcal{O}\left(n + n^{2/3}\varepsilon^{-2}\right)$	NA	Yes	$\mathcal{O}\left(L^{-1}n^{-1}\right) \to \mathcal{O}\left(L^{-1}\right)$	$1 \to n^{2/3}$
SCSG (Lihua et al., 2017)	$\mathcal{O}\left(n + n^{2/3}\varepsilon^{-2}\right)$	$\mathcal{O}\left(\sigma^2\varepsilon^{-2} + \sigma\varepsilon^{-10/3}\right)$	No	$\mathcal{O}\left(L^{-1}(n^{-2/3} \wedge \varepsilon^{4/3})\right)$	Optional
SPIDER (Fang et al., 2018)	$\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$	$\mathcal{O}\left(\sigma^2\varepsilon^{-2} + \sigma\varepsilon^{-3}\right)$	No	$\mathcal{O}\left(L^{-1}\varepsilon\right)$	Required
SpiderBoost (Wang et al., 2018)	$\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$	$\mathcal{O}\left(\sigma^2\varepsilon^{-2} + \sigma\varepsilon^{-3}\right)$	Yes	$\mathcal{O}\left(L^{-1}\right)$	$\sqrt{n}$ or ε^{-1}
SVRG+ (Li and Li, 2018)	$\mathcal{O}\left(n^{2/3}\varepsilon^{-2}\right)$	$\mathcal{O}\left(\sigma^2\varepsilon^{-10/3}\right)$	Yes	$\mathcal{O}\left(L^{-1}\right)$	Required
ProxSARAH (This work)	$\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$	$\mathcal{O}\left(\sigma\varepsilon^{-3}\right)$	Yes	$\mathcal{O}\left(L^{-1}m^{-1/2}\right) \to \mathcal{O}\left(L^{-1}\right)$	Optional

Table 1: *Comparison of results on SFO (stochastic first-order oracle) complexity for nonsmooth nonconvex optimization (both non-compsite and composite case). Here, m is the number of inner iterations (epoch length) and σ is the variance in Assumption 2.3, and "required" means that the algorithm uses mini-batch size to achieve the best complexity. Note that all the complexity bounds here must depend on the Lipschitz constant L of the smooth components and $F(\widetilde{w}^0) - F^\star$, the difference between the initial objective value $F(\widetilde{w}^0)$ and the lower-bound $F^\star$. For the sake of presentation, we assume that $L = \mathcal{O}(1)$ and ignore these quantities in the complexity bounds*

Now, let us compare in detail our algorithms and three methods: ProxSVRG, ProxSVRG+, and ProxSpiderBoost.

- **Single sample for the finite-sum case:** As shown in (Reddi et al., 2016b, Theorem 1), in the single sample case, i.e. the mini-batch size of the inner loop $\hat{b} = 1$, ProxSVRG for solving (2) has a small step-size $\eta = \frac{1}{3Ln}$, and its complexity is $\mathcal{O}\left(n/\varepsilon^2\right)$ (see (Reddi et al., 2016b, Corollary 1)). ProxSVRG+ in (Li and Li, 2018, Theorem 3) is a variant of ProxSVRG, and in the single sample case, it achieves $\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$ complexity bound but using a different step-size $\eta = \min\left\{\frac{1}{6L}, \frac{1}{6mL}\right\}$. This step-size is only better than ProxSVRG if $2m < n$. With this step-size, the complexity of ProxSVRG+ remains $\mathcal{O}\left(\frac{n}{\varepsilon^2}\right)$ as in ProxSVRG. In our ProxSARAH, we use two step-sizes $\gamma = \frac{\sqrt{2}}{\sqrt{3mL}}$ and $\eta = \frac{2\sqrt{3m}}{4\sqrt{3m}+\sqrt{2}}$, and their product presents a combined step-size, which is $\hat{\eta} := \gamma\eta = \frac{2}{4\sqrt{3m}+\sqrt{2}}$. Clearly, our step-size $\hat{\eta}$ is much larger than that of both ProxSVRG and ProxSVRG+. Moreover, with these step-sizes, our complexity bound is $\mathcal{O}\left(\frac{\sqrt{n}L}{\varepsilon^2}\right)$ and it is optimal. As we can observe from

Algorithm 1 in the sequel, the number of proximal operator calls in our method remains the same as in ProxSVRG or ProxSVRG+.

- **Mini-batch for the finite-sum case:** As indicated in (Reddi et al., 2016b, Theorem 2), if we choose the batch size $b = n^{2/3}$ and $m = \lfloor n^{1/3} \rfloor$, then the step-size η can be chosen as $\eta = \frac{1}{3L}$, and its complexity is improved up to $\mathcal{O}\left(n + n^{2/3}\varepsilon^{-2}\right)$. However, the mini-batch size $n^{2/3}$ is very large which tends to full proximal gradient methods. For ProxSVRG+ in (Li and Li, 2018), based on Theorem 1, we need to set $b = n^{2/3}$ and $m = \sqrt{\hat{b}} = n^{1/3}$ to obtain the best complexity bound, which is $\mathcal{O}\left(n^{2/3}\varepsilon^{-2}\right)$. For SpiderBoost in (Wang et al., 2018), it requires to properly set mini-batch size to achieve $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ complexity for (2) and $\mathcal{O}\left(\sigma^2\varepsilon^{-2} + \sigma\varepsilon^{-3}\right)$ complexity for (1). More precisely, from (Wang et al., 2018, Theorem 1), we can see that one needs to set $m = \sqrt{n}$ and $\hat{b} = \sqrt{n}$ to achieve such a complexity. Unfortunately, ProxSpiderBoost does not have theoretical guarantee for the single sample case (i.e., $\hat{b} = 1$). In our methods, it is flexible to choose the epoch length m and the batch size $\hat{b}$ such that we can obtain different step-sizes and complexity bounds. More details can be found in Subsection 3.2.3. In summary, it is clear that our $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$ complexity is better than the best-known result $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$. If n is larger than the order of ε^{-4}, e.g. $n = \mathcal{O}\left(\varepsilon^{-6}\right)$, then we have $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right) = \mathcal{O}\left(\varepsilon^{-5}\right)$ while $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right) = \mathcal{O}\left(\varepsilon^{-6} + \varepsilon^{-5}\right) = \mathcal{O}\left(\varepsilon^{-6}\right)$. Therefore, our complexity is clearly better in big data regime.

- **Online expectation or large-n case:** In the online or expectation case (1), SPIDER in (Fang et al., 2018, Theorem 1) achieves an $\mathcal{O}\left(\sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right)$ complexity. In the finite-sum case, it has $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ complexity. Similar results also hold for SpiderBoost in (Wang et al., 2018). To this end, the total complexity of these methods is $\mathcal{O}\left(\min\left\{n + n^{1/2}\varepsilon^{-2}, \sigma\varepsilon^{-3} + \sigma^2\varepsilon^{-2}\right\}\right)$. If $\sigma = \mathcal{O}(1)$ and $n \leq \mathcal{O}\left(\varepsilon^{-4}\right)$, then this complexity can be simplified as $\mathcal{O}\left(\min\left\{n^{1/2}\varepsilon^{-2}, \varepsilon^{-3}\right\}\right)$. However, the dependence of σ cannot simply be ignored since σ could depend on ε or can grow significantly large. In addition, in big data regime, n can be very large, e.g. $n > \mathcal{O}\left(\varepsilon^{-4}\right)$. Therefore, this complexity bound no longer holds. As shown in Theorem 8, our complexity is $\mathcal{O}\left(\sigma\varepsilon^{-3}\right)$ given that $L = \mathcal{O}(1)$. In this case, if $\sigma = \mathcal{O}\left(\varepsilon^{-2}\right)$, then our total complexity is $\mathcal{O}\left(\varepsilon^{-5}\right)$ while SPIDER and SpiderBoost have $\mathcal{O}\left(\varepsilon^{-5} + \varepsilon^{-6}\right) = \mathcal{O}\left(\varepsilon^{-6}\right)$ complexity.

From the above analysis, it is clear that our complexity results match the optimal rate in the finite-sum case and are better than others in the expectation case. From an algorithmic point of view, our method is fundamental different from existing methods due to its averaging step and large step-sizes in the composite settings. Moreover, it has a flexibility to choose parameters: the step-sizes η and γ, the epoch length m, the inner mini-batch size $\hat{b}$, and the snapshot batch size b_s to trade-off complexity bounds.

Paper organization: The rest of this paper is organized as follows. Section 2 discusses the fundamental assumptions and optimality conditions. Section 3 presents the main algorithmic framework and its convergence results for two settings. Section 4 considers extensions and special cases of our algorithms. Section 5 provides some numerical examples to verify our methods and compare them with existing state-of-the-arts.

2. Mathematical tools and preliminary results

Firstly, we recall some basic notation and concepts in optimization. They can be found in (Bauschke and Combettes, 2017; Nesterov, 2004). Then, we state our blanket assumptions and discuss the optimality condition of (1) and (2). Finally, we provide necessarily preliminary results used in the sequel.

2.1 Basic notation and concepts

We work with finite dimensional spaces, $\mathbb{R}^d$, equipped with standard inner product $\langle\cdot,\cdot\rangle$ and Euclidean norm $\|\cdot\|$. Given a function $f : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$, we use $\mathrm{dom}(f) := \big\{w\in\mathbb{R}^d \mid f(w) < +\infty\big\}$ to denote its (effective) domain. If f is proper, closed, and convex, $\partial f(w) := \big\{v\in\mathbb{R}^d \mid f(z) \geq f(w) + \langle v, z-w\rangle, \ \ \forall z\in\mathrm{dom}(f)\big\}$ denotes its subdifferential at w, and $\mathrm{prox}_f(w) := \arg\min_z\big\{f(z) + (1/2)\|z-w\|^2\big\}$ denotes its proximal operator. Note that if f is the indicator of a nonempty, closed, and convex set $\mathcal{X}$, i.e. $f(w) = \delta_{\mathcal{X}}(w)$, then $\mathrm{prox}_f(\cdot) = \mathrm{proj}_{\mathcal{X}}(\cdot)$, the projection of w onto $\mathcal{X}$. Any element $\nabla f(w)$ of $\partial f(w)$ is called a subgradient of f at w. If f is differentiable at w, then $\partial f(w) = \{\nabla f(w)\}$, the gradient of f at w. A continuous differentiable function $f : \mathbb{R}^d\to\mathbb{R}$ is said to be L_f-smooth if ∇f is Lipschitz continuous on its domain, i.e. $\|\nabla f(w) - \nabla f(z)\| \leq L_f\|w-z\|$ for $w, z\in\mathrm{dom}(f)$. We use $\mathbf{U}_p(S)$ to denote a finite set $S := \{s_1, s_2, \cdots, s_n\}$ equipped with a probability distribution p over S. If p is uniform, then we simply use $\mathbf{U}(S)$. For any real number a, $\lfloor a\rfloor$ denotes the largest integer less than or equal to a.

2.2 Fundamental assumptions

To develop numerical methods for solving (1) and (2), we rely on some basic assumptions usually used in stochastic optimization methods.

Assumption 2.1 (Bounded from below) *Both problems* (1) *and* (2) *are bounded from below. That is* $F^\star := \inf_{w\in\mathbb{R}^d} F(w) > -\infty$. *Moreover,* $\mathrm{dom}(F) := \mathrm{dom}(f)\cap\mathrm{dom}(\psi) \neq \emptyset$.

This assumption usually holds in practice since f often represents a loss function which is nonnegative or bounded from below. In addition, the regularizer ψ is also nonnegative or bounded from below, and its domain intersects $\mathrm{dom}(f)$.

Our next assumption is the smoothness of f with respect to the argument w.

Assumption 2.2 (L-smoothness) *In the expectation setting* (1), *for any realization of* $\xi\in\Omega$, $f(\cdot;\xi)$ *is* L*-smooth, i.e.* $f(\cdot;\xi)$ *is continuously differentiable and its gradient* $\nabla_w f(\cdot;\xi)$ *is Lipschitz continuous with the same Lipschitz constant* $L\in(0,+\infty)$, *i.e.:*

$$\mathbb{E}_\xi\left[\|\nabla f(w;\xi) - \nabla f(\hat{w};\xi)\|^2\right] \leq L^2\|w-\hat{w}\|^2, \quad w,\hat{w}\in\mathrm{dom}(f). \tag{3}$$

In the finite-sum setting (2), *we assume that each term* f_i *is* L*-smooth, i.e.:*

$$\|\nabla f_i(w) - \nabla f_i(\hat{w})\| \leq L\|w-\hat{w}\|, \quad w,\hat{w}\in\mathrm{dom}(f), \quad i = 1,\cdots,n. \tag{4}$$

It is well-known that the L-smooth condition leads to the following bound

$$\mathbb{E}_{\xi}\left[f(\hat{w};\xi)\right] \leq \mathbb{E}_{\xi}\left[f(w;\xi)\right] + \mathbb{E}_{\xi}\left[\langle\nabla_w f(w;\xi), \hat{w} - w\rangle\right] + \frac{L}{2}\|\hat{w} - w\|^2, \quad w, \hat{w} \in \mathrm{dom}(f). \tag{5}$$

Since

$$\begin{aligned}\|\nabla f(w) - \nabla f(\hat{w})\|^2 &= \|\mathbb{E}_{\xi}\left[\nabla f(w;\xi) - \nabla f(\hat{w};\xi)\right]\|^2 \\ &\leq \mathbb{E}_{\xi}\left[\|\nabla f(w;\xi) - \nabla f(\hat{w};\xi)\|^2\right] \\ &\leq L^2\|w - \hat{w}\|^2,\end{aligned}$$

we have $\|\nabla f(w) - \nabla f(\hat{w})\| \leq L\|w - \hat{w}\|$. Hence, using either (5) or (7), we get

$$f(\hat{w}) \leq f(w) + \langle\nabla f(w), \hat{w} - w\rangle + \frac{L}{2}\|\hat{w} - w\|^2, \quad w, \hat{w} \in \mathrm{dom}(f). \tag{6}$$

Alternatively, for (4), we have

$$f_i(\hat{w}) \leq f_i(w) + \langle\nabla f_i(w), \hat{w} - w\rangle + \frac{L}{2}\|\hat{w} - w\|^2, \quad w, \hat{w} \in \mathrm{dom}(f). \tag{7}$$

In this case, (6) still holds for f in the finite-sum minimization problem (2).

In the expectation setting (1), we need the following bounded variance condition:

Assumption 2.3 (Bounded variance) *For the expectation problem* (1)*, there exists a uniform constant $\sigma \in [0, +\infty)$ such that*

$$\mathbb{E}_{\xi}\left[\|\nabla f(w;\xi) - \nabla f(w)\|^2\right] \leq \sigma^2, \quad \forall w \in \mathbb{R}^d. \tag{8}$$

This assumption is standard in stochastic optimization and often required in almost any solution method for solving (1), see, e.g. (Ghadimi and Lan, 2013).

Remark 1 *For simplicity of analysis, we assume that all functions f_i in* (2) *are L-smooth with the same Lipschitz constant $L > 0$ for $i = 1, \cdots, n$. However, our analysis can be easily extended to the case where the Lipschitz constant of ∇f_i is not the same by appropriately choosing a distribution for sampling, or using arbitrary sampling schemes.*

2.3 Optimality conditions

Under Assumption 2.1, we have $\mathrm{dom}(f) \cap \mathrm{dom}(\psi) \neq \emptyset$. When $f(\cdot;\xi)$ is nonconvex in w, the first order optimality condition of (1) can be stated as

$$0 \in \partial F(w^\star) \equiv \nabla_w f(w^\star) + \partial\psi(w^\star) \equiv \mathbb{E}_{\xi}\left[\nabla_w f(w^\star;\xi)\right] + \partial\psi(w^\star). \tag{9}$$

Here, $w^\star$ is called a stationary point of F. We denote $\mathcal{S}^\star$ the set of all stationary points. The condition (9) is called the first-order optimality condition, and also holds for (2).

Since ψ is proper, closed, and convex, its proximal operator $\mathrm{prox}_{\eta\psi}$ satisfies the nonexpansiveness, i.e. $\|\mathrm{prox}_{\eta\psi}(w) - \mathrm{prox}_{\eta\psi}(z)\| \leq \|w - z\|$ for all $w, z \in \mathbb{R}^d$.

Now, for any fixed $\eta > 0$, we define the following quantity

$$G_{\eta}(w) := \frac{1}{\eta}\big(w - \mathrm{prox}_{\eta\psi}(w - \eta\nabla f(w))\big). \tag{10}$$

This quantity is called the gradient mapping of F (Nesterov, 2004). Indeed, if $\psi \equiv 0$, then $G_{\eta}(w) \equiv \nabla f(w)$, which is exactly the gradient of f. By using $G_{\eta}(\cdot)$, the optimality condition (9) can be equivalently written as

$$\|G_{\eta}(w^{\star})\|^2 = 0. \tag{11}$$

If we apply gradient-type methods to solve (1) or (2), then we can only aim at finding an ε-approximate stationary point $\widetilde{w}_T$ to $w^{\star}$ in (11) after at most T iterations within a given accuracy $\varepsilon > 0$, i.e.:

$$\mathbb{E}\left[\|G_{\eta}(\widetilde{w}_T)\|^2\right] \le \varepsilon^2. \tag{12}$$

The condition (12) is standard in stochastic nonconvex optimization methods. Stronger results such as approximate second-order optimality or strictly local minimum require additional assumptions and more sophisticated optimization methods such as cubic regularized Newton-type schemes, see, e.g., (Nesterov and Polyak, 2006).

2.4 Stochastic gradient estimators

One key step to design a stochastic gradient method for (1) or (2) is to query an estimator for the gradient $\nabla f(w)$ at any w. Let us recall some existing stochastic estimators.

Single sample estimators: A simple estimator of $\nabla f(w)$ can be computed as follows:

$$\widetilde{\nabla} f(w_t) := f(w_t; \xi_t), \tag{13}$$

where ξ_t is a realization of ξ. This estimator is unbiased, i.e., $\mathbb{E}\Big[\widetilde{\nabla} f(w_t) \mid \mathcal{F}_t\Big] = \nabla f(w)$, but its variance is fixed for any w_t, where $\mathcal{F}_t$ is the history of randomness collected up to the t-th iteration, i.e.:

$$\mathcal{F}_t := \sigma\big(w_0, w_1, \cdots, w_t\big). \tag{14}$$

This is a σ-field generated by random variables $\{w_0, w_1, \cdots, w_t\}$. In the finite-sum setting (2), we have $\widetilde{\nabla} f(w_t) := \nabla f_{i_t}(w_t)$, where $i_t \sim \mathbf{U}([n])$ with $[n] := \{1, 2, \cdots, n\}$.

In recent years, there has been a huge interest in designing stochastic estimators with variance reduction properties. The first variance reduction method was perhaps proposed in (Schmidt et al., 2017) since 2013, and then in (Defazio et al., 2014) for convex optimization. However, the most well-known method is SVRG introduced by Johnson and Zhang in (Johnson and Zhang, 2013) that works for both convex and nonconvex problems. The SVRG estimator for ∇f in (2) is given as

$$\widetilde{\nabla} f(w_t) := \nabla f(\widetilde{w}) + \nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}), \tag{15}$$

where $\nabla f(\widetilde{w})$ is the full gradient of f at a snapshot point $\widetilde{w}$, and i_t is a uniformly random index in $[n]$. It is clear that $\mathbb{E}\Big[\widetilde{\nabla} f(w_t) \mid \mathcal{F}_t\Big] = \nabla f(w_t)$, which shows that $\widetilde{\nabla} f(w_t)$ is an unbiased estimator of $\nabla f(w_t)$. Moreover, its variance is reduced along the iteration t.

Our methods rely on the SARAH estimator introduced in (Nguyen et al., 2017a) for the non-composite convex problem instances of (2), which is defined as follows:

$$v_t := v_{t-1} + \nabla f(w_t;\xi_t) - \nabla f(w_{t-1};\xi_t), \tag{16}$$

for a given realization ξ_t of ξ. Each evaluation of v_t requires two gradient evaluations. Clearly, the SARAH estimator is biased, since $\mathbb{E}\left[v_t \mid \mathcal{F}_t\right] = v_{t-1} + \nabla f(w_t) - \nabla f(w_{t-1}) \neq \nabla f(w_t)$. But it has a variance reduction property.

Minibatch estimators: We consider a mini-batch estimator of the gradient ∇f in (13) and of the SARAH estimator (16) respectively as follows:

$$\widetilde{\nabla} f_{\mathcal{B}_t}(w_t) := \frac{1}{b_t}\sum_{i\in\mathcal{B}_t}\nabla f(w_t;\xi_i) \ \text{ and } \ v_t := v_{t-1} + \frac{1}{b_t}\sum_{i\in\mathcal{B}_t}\left(\nabla f(w_t;\xi_i) - \nabla f(w_{t-1};\xi_i)\right), \tag{17}$$

where $\mathcal{B}_t$ is a mini-batch of the size $b_t := |\mathcal{B}_t| \geq 1$. For the finite-sum problem (2), we replace $f(\cdot;\xi_i)$ by $f_i(\cdot)$. In this case, $\mathcal{B}_t$ is a uniformly random subset of $[n]$. Clearly, if $b_t = n$, then we take the full gradient ∇f as the exact estimator.

2.5 Basic properties of stochastic and SARAH estimators

We recall some basic properties of the standard stochastic and SARAH estimators for (1) and (2). The following result was proved in (Nguyen et al., 2017a).

Lemma 2 *Let $\{v_t\}_{t\geq 0}$ be defined by (16) and $\mathcal{F}_t$ be defined by (14). Then*

$$\begin{aligned} &\mathbb{E}\left[v_t \mid \mathcal{F}_t\right] = \nabla f(w_t) + \epsilon_t \neq \nabla f(w_t), \quad \textit{where} \ \ \epsilon_t := v_{t-1} - \nabla f(w_{t-1};\xi_t).\\ &\mathbb{E}\left[\|v_t - \nabla f(w_t)\|^2 \mid \mathcal{F}_t\right] = \|v_{t-1} - \nabla f(w_{t-1})\|^2 + \mathbb{E}\left[\|v_t - v_{t-1}\|^2 \mid \mathcal{F}_t\right] \\ &\qquad\qquad - \|\nabla f(w_t) - \nabla f(w_{t-1})\|^2. \end{aligned} \tag{18}$$

Consequently, for any $t \geq 0$, we have

$$\begin{aligned} \mathbb{E}\left[\|v_t - \nabla f(w_t)\|^2\right] &= \mathbb{E}\left[\|v_0 - \nabla f(w_0)\|^2\right] + \textstyle\sum_{j=1}^t \mathbb{E}\left[\|v_j - v_{j-1}\|^2\right] \\ &\quad - \textstyle\sum_{j=1}^t \mathbb{E}\left[\|\nabla f(w_j) - \nabla f(w_{j-1})\|^2\right]. \end{aligned} \tag{19}$$

Our next result is some properties of the mini-batch estimators in (17). The proof is presented in (Harikandeh et al., 2015; Lohr, 2009; Nguyen et al., 2017b, 2018a), and we omit it here.

Lemma 3 *If $\widetilde{\nabla} f_{\mathcal{B}_t}(w_t)$ is generated by (17), then, under Assumption 2.3, we have*

$$\begin{aligned} &\mathbb{E}\left[\widetilde{\nabla} f_{\mathcal{B}_t}(w_t)\right] = \nabla f(w_t) \ \textit{and} \\ &\mathbb{E}\left[\|\widetilde{\nabla} f_{\mathcal{B}_t}(w_t) - \nabla f(w_t)\|^2\right] = \frac{\mathbb{E}\left[\|\nabla f(w_t;\xi) - \nabla f(w_t)\|^2\right]}{b_t} \leq \frac{\sigma^2}{b_t}. \end{aligned} \tag{20}$$

If $\widetilde{\nabla} f_{\mathcal{B}_t}(w_t)$ is generated by (17) *for the finite-sum problem* (2)*, then*

$$\mathbb{E}\left[\widetilde{\nabla} f_{\mathcal{B}_t}(w_t)\right] = \nabla f(w_t) \text{ and } \mathbb{E}\left[\|\widetilde{\nabla} f_{\mathcal{B}_t}(w_t) - \nabla f(w_t)\|^2\right] \le \frac{1}{b_t}\left(\frac{n-b_t}{n}\right)\sigma_n^2, \tag{21}$$

where σ_n^2 is an upper bound such that

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[\|\nabla f_i(w_t)\|^2 - \|\nabla f(w_t)\|^2\right] \le \sigma_n^2.$$

If v_t is generated by (17) *for the finite-sum problem* (2)*, then*

$$\mathbb{E}\left[\|v_t - v_{t-1}\|^2\right] \le \frac{1}{b_t}\left(\frac{n-b_t}{n-1}\right)\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_i(w_t) - \nabla f_i(w_{t-1})\|^2\right]. \tag{22}$$

Note that if $b_t = n$, i.e., we take a full gradient estimate, then the second estimate of (21) is vanished and independent of σ_n.

3. ProxSARAH framework and convergence analysis

We describe our basic algorithmic framework and then specify it to solve different instances of (1) and (2) under appropriate structures. The general algorithm is described in Algorithm 1. We abbreviate it by ProxSARAH.

Algorithm 1 (Proximal SARAH with stochastic recursive gradient estimators)

```
1:  Initialization: An initial point $\widetilde{w}_0$ and necessary parameters (will be specified).
2:  OuterLoop: For $s := 1, 2, \cdots, S$ do
3:      Generate a snapshot $v_0^{(s)}$ at $w_0^{(s)} := \widetilde{w}_{s-1}$.
4:      Update $\widehat{w}_1^{(s)} := \mathrm{prox}_{\eta_0\psi}(w_0^{(s)} - \eta_0 v_0^{(s)})$ and $w_1^{(s)} := (1-\gamma_0)w_0^{(s)} + \gamma_0\widehat{w}_1^{(0)}$.
5:      InnerLoop: For $t := 1, \cdots, m$ do
6:          Generate a proper single random sample or mini-batch $\hat{\mathcal{B}}_t^{(s)}$.
7:          Evaluate $v_t^{(s)} := v_{t-1}^{(s)} + \frac{1}{|\hat{\mathcal{B}}_t^{(s)}|}\sum_{\xi_t^{(s)}\in\hat{\mathcal{B}}_t^{(s)}}\left[\nabla f(w_t^{(s)};\xi_t^{(s)}) - \nabla f(w_{t-1}^{(s)};\xi_t^{(s)})\right]$.
8:          Update $\widehat{w}_{t+1}^{(s)} := \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)})$ and $w_{t+1}^{(s)} := (1-\gamma_t)w_t^{(s)} + \gamma_t\widehat{w}_{t+1}^{(s)}$.
9:      EndFor
10:     Set $\widetilde{w}_s := w_{m+1}^{(s)}$
11: EndFor
```

In terms of algorithm, ProxSARAH is different from SARAH where it has one proximal step followed by an additional averaging step, Step 8. However, using the gradient mapping G_η defined by (10), we can view Step 8 as follows:

$$w_{t+1}^{(s)} := w_t^{(s)} - \eta_t\gamma_t G_{\eta_t}(w_t^{(s)}).$$

Hence, this step is similar to a gradient step applying to the gradient mapping $G_{\eta_t}(w_t^{(s)})$. In particular, if we set $\gamma_t = 1$, then we obtain a vanilla proximal SARAH variant which is similar to ProxSVRG, ProxSVRG+, and ProxSpiderBoost discussed above. ProxSVRG, ProxSVRG+, and ProxSpiderBoost are simply vanilla proximal gradient-type methods in stochastic setttings. If $\psi = 0$, then ProxSARAH is reduced to SARAH in (Nguyen et al., 2017a,b, 2018b) with a step-size $\hat{\eta}_t := \gamma_t\eta_t$. Note that Step 8 can be represented as a weighted averaging step:

$$w_{t+1}^{(s)} := \frac{1}{\Sigma_t^{(s)}}\sum_{j=0}^{t}\tau_j^{(s)}\widehat{w}_{j+1}^{(s)}, \quad \text{where } \ \Sigma_t^{(s)} := \sum_{j=0}^{t}\tau_j^{(s)} \ \text{ and } \ \gamma_j^{(s)} := \frac{\tau_j^{(s)}}{\Sigma_t^{(s)}}.$$

Compared to (Ghadimi and Lan, 2012; Nemirovski et al., 2009), ProxSARAH evaluates v_t at the averaging point $w_t^{(s)}$ instead of $\widehat{w}_t^{(s)}$. Therefore, it can be written as

$$w_{t+1}^{(s)} := (1-\gamma_t)w_t^{(s)} + \gamma_t\mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)}),$$

which is similar to averaged fixed-point schemes (e.g. the Krasnosel'skiĭ – Mann scheme) in the literature, see, e.g., (Bauschke and Combettes, 2017).

In addition, we will show in our analysis a key difference in terms of step-sizes η_t and γ_t, mini-batch, and epoch length between ProxSARAH and existing methods, including SPIDER (Fang et al., 2018) and SpiderBoost (Wang et al., 2018).

3.1 Analysis of the inner-loop: Key estimates

This subsection proves two key estimates of the inner loop for $t = 0$ to m. We break our analysis into two different lemmas, which provide key estimates for our convergence analysis. We first consider the single sample case, i.e. $\hat{b} := |\hat{\mathcal{B}}_t^{(s)}| = 1$.

Lemma 4 *Let $\{(w_t, \widehat{w}_t)\}$ be generated by the inner-loop of Algorithm 1 with $|\hat{\mathcal{B}}_t^{(s)}| = 1$. Then, under Assumption 2.2, we have*

$$\begin{aligned}\mathbb{E}\left[F(w_{m+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_0^{(s)})\right] + \frac{L^2}{2}\sum_{t=0}^{m}\beta_t\sum_{j=1}^{t}\gamma_{j-1}^2\mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right] \\ &\quad + \frac{1}{2}\bar{\sigma}^{(s)}\Big(\sum_{t=0}^{m}\beta_t\Big) - \frac{1}{2}\sum_{t=0}^{m}\kappa_t\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \\ &\quad - \sum_{t=0}^{m}\frac{s_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] - \sum_{t=0}^{m}\mathbb{E}\left[\sigma_t^{(s)}\right],\end{aligned} \tag{23}$$

where $\{c_t\}$, $\{r_t\}$, and $\{s_t\}$ are given positive sequences, $\bar{\sigma}^{(s)} := \mathbb{E}\left[\|v_0^{(s)} - \nabla f(w_0^{(s)})\|^2\right] \geq 0$, $\sigma_t^{(s)} := \frac{\gamma_t}{2c_t}\|\nabla f(w_t^{(s)}) - v_t^{(s)} - c_t(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})\|^2 \geq 0$, and

$$\beta_t := \frac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2, \quad \textit{and} \quad \kappa_t := \frac{2\gamma_t}{\eta_t} - L\gamma_t^2 - \gamma_t c_t - s_t\left(1 + \frac{1}{r_t}\right). \tag{24}$$

The proof of Lemma 4 is deferred to Appendix A.1. The next lemma considers a special case by showing how to choose constant step-sizes γ and η, and other parameters to obtain a descent property. The proof of this lemma is given in Appendix A.2.

Lemma 5 *Under Assumption 2.2 and* $|\hat{\mathcal{B}}_t^{(s)}| = 1$, *let us choose* $\eta_t \equiv \eta > 0$ *and* $\gamma_t = \gamma > 0$ *in Algorithm 1 such that*

$$\gamma_t \equiv \gamma := \frac{1}{L\sqrt{\omega m}} \quad \textit{and} \quad \eta_t \equiv \eta := \frac{2\sqrt{\omega m}}{q\sqrt{\omega m}+1}, \tag{25}$$

where $q := 2 + c + \frac{1}{r}$ *and* $\omega := \frac{1}{c} + \frac{4r^2(1+r)}{((2+c)r+1)^2}$ *such that* $\omega m \geq 1$. *Then, we have*

$$\begin{aligned} \mathbb{E}\left[F(w_{m+1}^{(s)})\right] & \leq \mathbb{E}\left[F(w_0^{(s)})\right] - \frac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \\ & \quad - \sum_{t=0}^{m}\mathbb{E}\left[\sigma_t^{(s)}\right] + \frac{\gamma\theta}{2}(m+1)\bar{\sigma}^{(s)}, \end{aligned} \tag{26}$$

where $\theta := \frac{1}{c} + \frac{4r^2(1+r)\omega m}{[r+((2+c)r+1)\sqrt{\omega m}]^2}$. *In particular, if we choose* $c = r = 1$, *then* $\omega = \frac{3}{2}$ *and* $\theta = 1 + \frac{8\omega m}{(1+\sqrt{\omega m})^2} < \frac{3}{2}$.

3.2 Convergence analysis for composite finite-sum problem (2)

In this subsection, we specify missing steps of Algorithm 1 to solve the composite finite-sum problem (2). We consider two cases: single sample (i.e. $|\hat{\mathcal{B}}_t^{(s)}| = 1$), and mini-batch (i.e. $|\hat{\mathcal{B}}_t^{(s)}| > 1$) of the inner loop.

3.2.1 The single sample case: $|\hat{\mathcal{B}}_t^{(s)}| = 1$

First, there are two steps that we need to adapt in Algorithm 1:

- **Step 3:** Evaluate $v_0^{(s)} := \frac{1}{|\mathcal{B}_s|}\sum_{i\in\mathcal{B}_s}\nabla f_i(w_0^{(s)})$, where $\mathcal{B}_s$ is a mini-batch at the sth outer iteration and independent of i_t of the inner loop. Note that if $|\mathcal{B}_s| = n$, then we take a full gradient snapshot.
- **Step 7:** Evaluate $v_t^{(s)} := v_{t-1}^{(s)} + \nabla f_{i_t}(w_t^{(s)}) - \nabla f_{i_t}(w_{t-1}^{(s)})$, where $i_t \sim \mathbf{U}([n])$.

With these two modified steps in Algorithm 1, we can prove the following main result.

Theorem 6 *Let us apply Algorithm 1 to solve* (2) *with the above two modified steps and fixed step-sizes* $\gamma := \frac{1}{L\sqrt{\omega m}}$ *and* $\eta := \frac{2\sqrt{\omega m}}{q\sqrt{\omega m}+1}$ *defined in* (25). *Then, under Assumptions 2.1 and 2.2, we have the following estimate:*

$$\begin{aligned} \frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] & \leq \frac{2}{\gamma\eta^2(m+1)S}\left[F(\widetilde{w}_0) - F^\star\right] \\ & \quad + \frac{\theta\sigma_n^2}{\eta^2 S}\sum_{s=1}^{S}\Big(\frac{n-b_s}{nb_s}\Big). \end{aligned} \tag{27}$$

In particular, if we choose $b_s = n$ for all $s = 1, \cdots, S$ and $m = n$, then, with $\widetilde{w}_T \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0\to m}^{s=0\to S}\big)$, we can obtain $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$ after at most $T := (m+1)S \equiv \mathcal{O}\left(\frac{\sqrt{n}L[F(\widetilde{w}_0)-F^\star]}{\varepsilon^2}\right)$ iterations. Consequently, the number of individual gradient evaluations ∇f_i is at most $3T$, and the number of proximal operations $\mathrm{prox}_{\eta\psi}$ is at most T.

Proof The step-sizes η and γ in Theorem 6 correspond to the choice $r = c = 1$ from Lemma 5. Summing up (26) from $s = 1$ to $s = S$, and then using $w_0^{(0)} := \widetilde{w}_0$ fixed and ignoring the nonnegative term $\sum_{s=0}^S \sum_{t=0}^m \mathbb{E}\left[\sigma_t^{(s)}\right]$, we obtain

$$\frac{\gamma\eta^2}{2}\sum_{s=1}^S\sum_{t=0}^m \mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq F(\widetilde{w}_0) - \mathbb{E}\left[F(w_{m+1}^{(S)})\right] + \frac{\gamma\theta(m+1)}{2}\sum_{s=1}^S \bar{\sigma}^{(s)}.$$

By Assumption 2.1, we have $\mathbb{E}\left[F(w_{m+1}^{(S)})\right] \geq F^\star > -\infty$. Moreover, by (21), we have $\bar{\sigma}^{(s)} := \mathbb{E}\left[\|v_0^{(s)} - \nabla f(w_0^{(s)})\|^2\right] = \mathbb{E}\left[\|\widetilde{\nabla} f_{\mathcal{B}_s}(w_0^{(s)}) - \nabla f(w_0^{(s)})\|^2\right] \leq \frac{1}{b_s}\left(\frac{n-b_s}{n}\right)\sigma_n^2$. Using these estimates into the last inequality, we can overestimate it as

$$\frac{\gamma\eta^2}{2}\sum_{s=1}^S\sum_{t=0}^m \mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq \left[F(\widetilde{w}_0) - F^\star\right] + \frac{\gamma\theta\sigma_n^2(m+1)}{2}\sum_{s=1}^S \frac{1}{b_s}\left(\frac{n-b_s}{n}\right).$$

Multiplying both sides of this inequality by $\frac{2}{\gamma\eta^2(m+1)S}$ we obtain (27).

In particular, if $b_s = n$ and $m = n$, and $\widetilde{w}_T \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$, then we have

$$\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] = \frac{1}{(n+1)S}\sum_{s=1}^S\sum_{t=0}^n \mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq \frac{2}{\gamma\eta^2(n+1)S}\big[F(\widetilde{w}_0) - F^\star\big].$$

Note that since $\omega m \geq 1$, we have $\gamma = \frac{1}{L\sqrt{\omega m}} = \frac{1}{L\sqrt{\omega n}}$ and $\eta = \frac{2}{q+\frac{1}{\sqrt{\omega m}}} \geq \frac{2}{q+1}$, to get $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$, we need to impose

$$T := (n+1)S = \frac{2[F(\widetilde{w}_0) - F^\star]}{\gamma\eta^2\varepsilon^2} \leq \frac{(q+1)^2L\sqrt{\omega n}[F(\widetilde{w}_0) - F^\star]}{2\varepsilon^2} = \mathcal{O}\left(\frac{L\sqrt{n}}{\varepsilon^2}\right).$$

This is the maximum number of iterations. Finally, we can show that the number of gradient evaluations ∇f_i is $T_{\text{grad}} = S(n + 2(m+1)) = 3S(n+1) - S = 3T - S \leq 3T$, and the number of proximal operator calls $\mathrm{prox}_{\eta\psi}$ is at most T. ■

Ignoring the term $F(\widetilde{w}_0) - F^\star$, we can see that the complexity of Algorithm 1 for solving (2) is $\mathcal{O}\left(\frac{\sqrt{n}L}{\varepsilon^2}\right)$ which dominates all existing results in the literature.

3.2.2 The mini-batch for the inner-loop: $|\hat{\mathcal{B}}_t^{(s)}| > 1$

Now, we analyze the case of using mini-batch in the inner loop of Algorithm 1. Particularly, we replace **Step 7** in Algorithm 1 by the following one:

- **Step 7:** $v_t^{(s)} := v_{t-1}^{(s)} + \frac{1}{\hat{b}_t^{(s)}}\sum_{i\in\hat{\mathcal{B}}_t}\left(\nabla f_i(w_t^{(s)}) - \nabla f_i(w_{t-1}^{(s)})\right)$, where $\hat{\mathcal{B}}_t^{(s)}$ is a mini-batch of the size $\hat{b}_t^{(s)} := |\hat{\mathcal{B}}_t^{(s)}|$.

The following theorem shows the convergence of this variant, whose proof is postponed until Appendix A.3.

Theorem 7 *Let us apply Algorithm 1 to solve* (2) *with the mini-batch estimator $v_t^{(s)}$ defined as above such that $\hat{b}_t^{(s)} = \hat{b} \geq 1$, and fixed step-sizes γ and η as*

$$\gamma := \frac{1}{L\sqrt{\rho m}} \quad \text{and} \quad \eta := \frac{2\sqrt{\rho m}}{4\sqrt{\rho m}+1}, \quad \text{where} \quad \rho := \frac{3(n-\hat{b})}{2\hat{b}(n-1)}. \tag{28}$$

Then, under Assumptions 2.1 and 2.2, and the choice $m := \lfloor \frac{n}{\hat{b}} \rfloor$ and $b_s := n$, the conclusions of Theorem 6 are still valid for this algorithmic variant with $T := \mathcal{O}\left(\frac{L\sqrt{n-\hat{b}}[F(\widetilde{w}_0)-F^\star]}{\hat{b}\varepsilon^2}\right)$. The number of gradient evaluations ∇f_i is $T_{\mathrm{grad}} := \mathcal{O}\left(\frac{L\sqrt{n-\hat{b}}[F(\widetilde{w}_0)-F^\star]}{\varepsilon^2}\right)$, and the number of proximal operator calls $\mathrm{prox}_{\eta\psi}$ is T.

We note that the mini-batch variant does not change the convergence rate, but it improves the complexity bound by a constant factor due to larger step-sizes γ and η.

3.2.3 Mini-batch size and learning rate trade-offs

Although our step-size in the single sample case is much larger than that of ProxSVRG in (Reddi et al., 2016b, Theorem 1), it still depends on $\sqrt{m}$, where m is the epoch length. To obtain larger step-sizes, we can choose m and the mini-batch size $\hat{b}$ using the same trick as in (Reddi et al., 2016b, Theorem 2). Let us first fix $\gamma := \bar{\gamma} \in (0,1]$. From (28), we have $\rho m = \frac{1}{L^2\bar{\gamma}^2}$. It makes sense to choose $\bar{\gamma}$ close to 1 in order to use new information from $\widehat{w}_{t+1}^{(s)}$ instead of the old one in $w_t^{(s)}$. Our goal is to choose m and $\hat{b}$ such that $\rho m = \frac{3(n-\hat{b})m}{2\hat{b}(n-1)} = \frac{1}{L^2\bar{\gamma}^2}$. If we define $C := \frac{2}{3L^2\bar{\gamma}^2}$, then the last condition implies that $\hat{b} := \frac{mn}{Cn+m-C} \leq \frac{m}{C}$ provided that $m \geq C$. Our suggestion is to choose

$$\gamma := \bar{\gamma} \in (0,1], \quad \hat{b} := \left\lfloor \frac{mn}{Cn+m-C} \right\rfloor, \quad \text{and} \quad \eta := \frac{2}{4+L\bar{\gamma}} \tag{29}$$

If we choose $m = \lfloor n^{1/3} \rfloor$, then $\hat{b} = \mathcal{O}\left(n^{1/3}\right) \leq \frac{n^{1/3}}{C}$. This mini-batch size is much smaller than $\lfloor n^{2/3} \rfloor$ in ProxSVRG. Note that, in ProxSVRG, they set $\gamma := 1$ and $\eta := \frac{1}{3L}$. In ProxSpiderBoost (Wang et al., 2018), m and the mini-batch size $\hat{b}$ were chosen as $m = \hat{b} = \lfloor n^{1/2} \rfloor$ so that they can use constant step-sizes $\gamma = 1$ and $\eta = \frac{1}{2L}$. In our case, if $\gamma = 1$, then $\eta = \frac{2}{4+L}$. Hence, if $L = 1$, then $\eta_{\text{ProxSpiderBoost}} = \frac{1}{2} > \eta_{\text{ProxSARAH}} = \frac{2}{5} > \eta_{\text{ProxSVRG}} = \frac{1}{3}$. But if $L > 4$, then our step-size $\eta_{\text{ProxSARAH}}$ dominates $\eta_{\text{ProxSpiderBoost}}$.

If we choose $m = \mathcal{O}\left(n^{1/2}\right)$ and $\hat{b} = \mathcal{O}\left(n^{1/2}\right)$, then we maintain the same complexity bound $\mathcal{O}\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ as in Theorem 7. However, if we choose $m = \mathcal{O}\left(n^{1/3}\right)$ and $\hat{b} = \mathcal{O}\left(n^{1/3}\right)$, then the complexity bound becomes $\mathcal{O}\left(\frac{n^{2/3}+n^{1/3}}{\varepsilon^2}\right)$, which is suboptimal.

3.2.4 Lower-bound complexity and optimal complexity

Let us analyze a special case of (2) with $\psi = 0$. We consider any stochastic first-order methods to generate an iterate sequence $\{w_t\}$ as follows

$$[w_t, i_t] := \mathcal{A}^{t-1}\left(\xi, \nabla f_{i_0}(w^0), \nabla f_{i_1}(w^1), \cdots, \nabla f_{i_{t-1}}(w^{t-1})\right), \quad t \geq 1, \tag{30}$$

where $\mathcal{A}^{t-1}$ are measure mapping into $\mathbb{R}^{d+1}$, f_{i_t} is an individual function chosen by $\mathcal{A}^{t-1}$ at iteration t, $\xi \sim \mathbf{U}([0,1])$ is a random vector, and $[w^0, i_0] := \mathcal{A}^0(\xi)$. Clearly, Algorithm 1 can be cast as a special case of (30). As shown in (Fang et al., 2018, Theorem 3) and later in (Zhou and Gu, 2019), under Assumptions 2.1 and 2.2, the lower-bound complexity of Algorithm 1 to produce an output $\widetilde{w}_T$ such that $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$ is $\Omega\left(\frac{L\sqrt{n}}{\varepsilon^2}\right)$. This lower-bound clearly matches the upper bound in Theorem 6. Therefore, Algorithm 1 achieves optimal complexity for solving (2).

3.3 Convergence analysis for the composite expectation problem (1)

In this subsection, we apply Algorithm 1 to solve the general expectation setting (1). For simplicity of presentation, we only consider the single sample case, i.e. $\hat{\mathcal{B}}_t^{(s)} = 1$. The mini-batch case for inner-loop is very similar to Theorem 7, and we omit the details.

In this case, we generate the snapshot at Step 3 of Algorithm 1 as follows:

- **Step 3:** Evaluate $v_0^{(s)} := \frac{1}{b_s}\sum_{\zeta_i^{(s)} \in \mathcal{B}_s} \nabla f(w_0^{(s)}; \zeta_i^{(s)})$, where $\mathcal{B}_s := \left\{\zeta_1^{(s)}, \cdots, \zeta_{b_s}^{(s)}\right\}$ is a mini-batch of i.i.d. realizations of ξ at the s-th outer iteration and independent of ξ_t from the inner loop.

Now, we analyze the convergence of Algorithm 1 for solving (1) using **Step 3** above.

Theorem 8 *Let us apply Algorithm 1 to solve* (1) *using **Step 3** above with fixed batch-size $b_s = b \geq 1$, single sample (i.e. $|\hat{\mathcal{B}}_t^{(s)}| = 1$), and fixed step-sizes γ and η as*

$$\gamma := \frac{\sqrt{2}}{L\sqrt{3m}} \qquad \textit{and} \qquad \eta := \frac{2\sqrt{3m}}{4\sqrt{3m} + \sqrt{2}}. \tag{31}$$

Then, under Assumptions 2.1 and 2.2, we have the following estimate:

$$\frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \leq \frac{2}{\gamma\eta^2(m+1)S}\left[F(\widetilde{w}_0) - F^\star\right] + \frac{3\sigma^2}{2\eta^2 b}. \tag{32}$$

In particular, if we choose $b = \mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2}\right)$ and $m = \mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2}\right)$, then after at most

$$T := S(m+1) \equiv \mathcal{O}\left(\frac{\sigma L[F(\widetilde{w}_0) - F^\star]}{\varepsilon^3}\right).$$

iterations, we obtain $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \le \varepsilon^2$, where $\widetilde{w}_T \ \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$. Consequently, the number of gradient evaluations $\nabla_w f(\cdot,\cdot)$ is at most $T_{\text{grad}} := \mathcal{O}\left(\frac{\sigma L[F(\widetilde{w}_0) - F^\star]}{\varepsilon^3}\right)$ and the number of proximal operator call $\text{prox}_{\eta\psi}$ is at most T.

Proof Summing up (26) from $s = 1$ to $s = S$, using $w_0^{(0)} = \widetilde{w}_0$, and ignoring the nonnegative term $\mathbb{E}\left[\sigma_t^{(s)}\right]$, we obtain

$$\frac{\gamma\eta^2}{2}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \le F(\widetilde{w}_0) - \mathbb{E}\left[F(w_{m+1}^{(S)})\right] + \frac{\gamma\theta(m+1)}{2}\sum_{s=1}^{S}\bar{\sigma}^{(s)}. \tag{33}$$

Note that $\mathbb{E}\left[F(w_{m+1}^{(S)})\right] \ge F^\star$ by Assumption 2.1. Moreover, by (20), we have

$$\bar{\sigma}^{(s)} := \mathbb{E}\left[\|v_0^{(s)} - \nabla f(w_0^{(s)})\|^2\right] = \mathbb{E}\left[\|\widetilde{\nabla} f_{\mathcal{B}_s}(w_0^{(s)}) - \nabla f(w_0^{(s)})\|^2\right] \le \frac{\sigma^2}{b_s} = \frac{\sigma^2}{b}.$$

Since $c = r = 1$ in Lemma 5, we have $\theta = 1 + \frac{8\omega m}{(1+4\sqrt{\omega m})^2} < \frac{3}{2}$. Using these estimates into (33), we obtain (32).

Now, since $\widetilde{w}_T \ \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$ for $T := S(m+1)$, we have

$$\begin{aligned}\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] &= \frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] \\ &\le \frac{2}{\gamma\eta^2(m+1)S}[F(\widetilde{w}_0) - F^\star] + \frac{3\sigma^2}{2\eta^2 b}.\end{aligned}$$

Since $\eta = \frac{2\sqrt{3m}}{4\sqrt{3m}+\sqrt{2}} \ge \frac{2}{4+\sqrt{2}} \ge \frac{2}{5}$, to guarantee $\mathbb{E}\left[\|G_\eta(\widetilde{w}_T)\|^2\right] \le \varepsilon^2$, we need to set

$$\frac{2}{\gamma(m+1)S}[F(\widetilde{w}_0) - F^\star] + \frac{3\sigma^2}{2b} = \frac{4\varepsilon^2}{25}.$$

Note that since the number of iterations $T := S(m+1)$, if b is chosen such that $b > \frac{75\sigma^2}{8\varepsilon^2}$, then this inequality implies

$$T := \frac{100L\sqrt{3m}b[F(\widetilde{w}_0) - F^\star]}{\sqrt{2}(8b\varepsilon^2 - 75\sigma^2)}$$

Let us choose $b := \frac{(75+C_1)\sigma^2}{8\varepsilon^2}$ and $m := \frac{C_2\sigma^2}{\varepsilon^2}$ for some $C_1 > 0$ and $C_2 > 0$ independent of σ. Then, the number of iterations T is

$$T := S(m+1) \equiv \frac{100L\sqrt{3C_2}(75+C_1)\sigma[F(\widetilde{w}_0) - F^\star]}{\sqrt{2}C_1\varepsilon^3} = \mathcal{O}\left(\frac{L\sigma[F(\widetilde{w}_0) - F^\star]}{\varepsilon^3}\right).$$

Now, we estimate the total number of gradient evaluations

$$\begin{aligned} T_{\text{grad}} &= \textstyle\sum_{s=1}^{S} b_s + 2mS = (b+2m)S = \frac{T(2m+b)}{m+1} \\ &= \frac{(75+C_1)\sigma^2+16C_2\sigma^2}{8(C_2\sigma^2+\varepsilon^2)} T \le \frac{(75+C_1)+16C_2}{8C_2} T. \end{aligned}$$

Hence, the number of iterations is $T := \mathcal{O}\left(\frac{L\sigma[F(\widetilde{w}_0)-F^\star]}{\varepsilon^3}\right)$, the number of gradient evaluations is $\mathcal{O}\left(\frac{L\sigma[F(\widetilde{w}_0)-F^\star]}{\varepsilon^3}\right)$, and the number of proximal operator calls is also T. ■

Note that we can trade-off the number of iterations T and the number of gradient evaluations T_{grad} in Theorem 8 by choosing $m := \mathcal{O}\left(\frac{\sigma^\nu}{\varepsilon^2}\right)$ for some $\nu \ge 0$. In this case, $T := \mathcal{O}\left(\frac{L\sigma^{\frac{\nu}{2}}}{\varepsilon^3}\right)$ and $T_{\text{grad}} := \mathcal{O}\left(\frac{(\sigma^{2-\frac{\nu}{2}}+\sigma^{\frac{\nu}{2}})L}{\varepsilon^3}\right)$. In particular, if $\nu = 0$, then $T := \mathcal{O}\left(\frac{L}{\varepsilon^3}\right)$ and $T_{\text{grad}} := \mathcal{O}\left(\frac{L\sigma^2}{\varepsilon^3}\right)$. The optimal value of ν to minimize T_{grad} is $\nu = 2$.

Theorem 8 achieves the best-known complexity $\mathcal{O}\left(\frac{\sigma L}{\varepsilon^3}\right)$ for the composite expectation problem (1). Compared to the $\mathcal{O}\left(\frac{\sigma}{\varepsilon^3} + \frac{\sigma^2}{\varepsilon^2}\right)$ complexity bound of SPIDER (Fang et al., 2018) and ProxSpiderBoost (Wang et al., 2018), our bound is better if $\sigma > \mathcal{O}\left(\varepsilon^{-1}\right)$. Moreover, our method does not require to perform mini-batch in the inner loop, i.e., it is independent of $\hat{\mathcal{B}}_t^{(s)}$, and the mini-batch is independent of the number of iterations m of the inner loop, while in (Wang et al., 2018), the mini-batch size $|\hat{\mathcal{B}}_t^{(s)}|$ should be proportional to $\sqrt{|\mathcal{B}_s|} = \mathcal{O}\left(\frac{1}{\varepsilon}\right)$, where $\mathcal{B}_s$ is the mini-batch of the outer loop. This is perhaps the reason why ProxSpiderBoost can take a large constant step-size $\eta = \frac{1}{2L}$.

4. Special case and extension

In this section, we consider the non-composite settings of (1) and (2) as special cases and an extension of Algorithm 1 to adaptive step-sizes η_t and γ_t.

4.1 The non-composite case

Note that if we solely apply Algorithm 1 with constant stepsizes to solve the non-composite case of (1) and (2) when $\psi \equiv 0$, then by using the same step-size as in Theorem 8, we can obtain the same complexity as stated in Theorem 8. However, we will modify our proof of Theorem 8 to obtain a larger step-size in an adaptive manner as shown in Theorem 9, whose proof is given in Appendix A.4.

Theorem 9 *Let $\{w_t^{(s)}\}$ be the sequence generated by the variant of Algorithm 1 using single sample, i.e. $|\hat{\mathcal{B}}_t^{(s)}| = 1$ and the update:*

$$w_{t+1}^{(s)} := w_t^{(s)} - \hat{\eta}_t v_t^{(s)} \tag{34}$$

*for both Step 4 and Step 8 and using **Step 3**, where the step-size $\hat{\eta}_t$ is computed recursively backward from $t = m$ to $t = 0$ as*

$$\hat{\eta}_m = \frac{1}{L}, \quad \textit{and} \quad \hat{\eta}_{m-t} := \frac{1}{L\big(1 + L\sum_{j=1}^{t}\hat{\eta}_{m-j+1}\big)}, \quad \forall t = 1, \cdots, m. \tag{35}$$

Then, we have $\Sigma_m := \sum_{t=0}^{m}\hat{\eta}_t \geq \frac{2(m+1)}{(\sqrt{2m+3}+1)L}$.
Suppose that Assumptions 2.1 and 2.2 hold, and $\widetilde{w}_T \sim \mathbf{U}_p\big(\{w_t^{(s)}\}_{t=0\to m}^{s=1\to S}\big)$ *such that*

$$\textbf{Prob}\left(\widetilde{w}_T = w_t^{(s)}\right) = p_{(s-1)m+t} := \frac{\hat{\eta}_t}{S\Sigma_m}, \quad \forall s = 1,\cdots,S,\ t = 0,\cdots,m.$$

Then, we have

$$\begin{aligned} \mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] &= \frac{1}{S\Sigma_m}\sum_{s=1}^{S}\sum_{t=0}^{m}\hat{\eta}_t\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] \\ &\leq \frac{(\sqrt{2m+3}+1)L}{S(m+1)}\left[f(\widetilde{w}_0) - f^{\star}\right] + \frac{1}{S}\sum_{s=1}^{S}\hat{\sigma}_s, \end{aligned} \tag{36}$$

where $\hat{\sigma}_s := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right]$.

We consider two cases:

(a) *If we apply this variant of Algorithm 1 to solve the non-composite instance of* (2) (*i.e.* $\psi = 0$) *using full gradient snapshot* $b_s = n$, *and* $m = n$, *then*

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{L(1+\sqrt{2n+3})}{S(n+1)}[f(\widetilde{w}_0) - f^{\star}]. \tag{37}$$

Consequently, the total of iterations T *to achieve an* ε*-stationary point* $\widetilde{w}_T$ *such that* $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$ *is at most* $T := \mathcal{O}\left(\frac{\sqrt{n}L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^2}\right)$. *The number of gradient evaluations* ∇f_i *is at most* $T_{\mathrm{grad}} := \mathcal{O}\left(\frac{\sqrt{n}L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^2}\right)$.

(b) *If we apply this variant of Algorithm 1 to solve the non-composite expectation instance of* (1) (*i.e.* $\psi = 0$) *using mini-batch size* $b_s = b := \mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2}\right)$ *for the outer-loop and* $m = \mathcal{O}\left(\frac{\sigma^{\nu}}{\varepsilon^2}\right)$ *for some* $\nu \geq 0$, *then*

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{L(1+\sqrt{2m+3})}{S(m+1)}\left[f(\widetilde{w}_0) - f^{\star}\right] + \frac{\sigma^2}{b}. \tag{38}$$

Consequently, the total of iterations T *to achieve an* ε*-stationary point* $\widetilde{w}_T$ *such that* $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$ *is at most* $T := \mathcal{O}\left(\frac{\sigma^{\frac{\nu}{2}}L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^3}\right)$. *The number of gradient evaluations is at most* $T_{\mathrm{grad}} := \mathcal{O}\left(\frac{(\sigma^{2-\frac{\nu}{2}}+\sigma^{\frac{\nu}{2}})L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^3}\right)$. *In particular, if* $\nu = 2$, *then* $T := \mathcal{O}\left(\frac{\sigma L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^3}\right)$ *and* $T_{\mathrm{grad}} := \mathcal{O}\left(\frac{\sigma L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^3}\right)$.

Note that the first statement (a) of Theorem 9 covers the nonconvex case of (Nguyen et al., 2019) by fixing step-size $\hat{\eta}_t = \hat{\eta} = \frac{2}{L(1+\sqrt{4m+1})}$. However, this constant step-size is rather small if $m = \mathcal{O}(n)$ is large. Hence, it is better to update $\hat{\eta}_t$ adaptively increasing as in (35), where $\hat{\eta}_m = \frac{1}{L}$ is a large step-size.

Again, by combining the first statement (a) of Theorem 9 and the lower-bound complexity in (Fang et al., 2018), we can conclude that this algorithmic variant still achieves an optimal complexity $\mathcal{O}\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ for the non-composite finite-sum problem in (2) to find an ε-stationary point in expectation.

4.2 Adaptive step-size for the composite case

We can extend Theorem 6 and Theorem 8 to adaptive step-sizes variants as in Theorem 9, but to solve composite problems for both single sample and mini-batch cases.

4.2.1 The single sample case

Our goal is to choose the parameters $(\gamma_t, \eta_t, c_t, s_t, r_t) > 0$ for all t such that the following condition derived from (23) holds:

$$\tfrac{L^2}{2}\sum_{t=0}^m \beta_t \sum_{j=1}^t \gamma_{j-1}^2 \mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right] - \sum_{t=0}^m \frac{\kappa_t}{2}\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \le 0,$$

where $\beta_t := \frac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2$ and $\kappa_t := \frac{2\gamma_t}{\eta_t} - L\gamma_t^2 - \gamma_t c_t - s_t\left(1 + \frac{1}{r_t}\right)$. The last condition becomes

$$\sum_{t=0}^{m-1}\left[L^2\gamma_t^2 \sum_{j=t+1}^m \beta_j - \kappa_t\right]\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] - \kappa_m \mathbb{E}\left[\|\widehat{w}_{m+1}^{(s)} - w_m^{(s)}\|^2\right] \le 0.$$

This condition is tight if we choose the parameters such that

$$\begin{cases} \kappa_m & = 0, \\ L^2\gamma_t^2 \sum_{j=t+1}^m \beta_j - \kappa_t & = 0, \ \forall t = 0, \cdots, m-1. \end{cases} \tag{39}$$

There are several parameter choices that satisfy (39). As an example, let us fix $c_t = \frac{1}{2}$, $r_t = 1$, choose $s_t = \frac{1}{4}\gamma_t$ and $\eta_t = \eta > 0$ fixed. In this case, (39) becomes

$$\begin{cases} 1 - L\eta\gamma_m & = 0, \\ (2 + \frac{\eta^2}{2})L^2\gamma_t\sum_{j=t+1}^m \gamma_j - 1 + L\eta\gamma_t & = 0, \ \forall t = 0, \cdots, m-1. \end{cases}$$

This condition leads to the following update

$$\gamma_m := \frac{1}{L\eta}, \quad \text{and} \quad \gamma_t := \frac{1}{L\left[\eta + L\sum_{j=t+1}^m (2 + \frac{\eta^2}{2})\gamma_j\right]}, \quad t = 0, \cdots, m-1. \tag{40}$$

This update shows that $0 < \gamma_0 < \gamma_1 < \cdots < \gamma_m = \frac{1}{L\eta}$. Moreover, we have

$$\Gamma_m := \sum_{t=0}^m \gamma_t \ge \frac{2(m+1)}{(\sqrt{(4+\eta^2)m + 2\eta^2 + 4} + \eta)L} = \mathcal{O}\left(\frac{\sqrt{m}}{L}\right).$$

Clearly, the update rule (40) is larger than the fixed step-size in Theorem 6 and Theorem 8. The convergence of algorithmic variants using the update rule (40) is shown in the following corollary, whose proof is similar to that of Theorem 9 and we omit the details.

Corollary 10 *Let us apply the variant of Algorithm 1 in Theorem 6 to solve* (2) *but using the adaptive step-sizes* (40)*. Then, under Assumptions 2.1 and 2.2, the conclusions of Theorem 6 are still valid.*

4.2.2 The mini-batch case $\vert\widehat{\mathcal{B}}_t^{(s)}\vert > 1$

Now, we assume that the mini-batch size $\hat{b} = \vert\widehat{\mathcal{B}}_t^{(s)}\vert > 1$. The condition (39) becomes

$$\begin{cases} \kappa_m & = 0, \\ \frac{(n-\hat{b})}{\hat{b}(n-1)} L^2\gamma_t^2 \sum_{j=t+1}^m \beta_j - \kappa_t & = 0, ~\forall t = 0, \cdots, m-1. \end{cases} \tag{41}$$

As an example, if we fix $c_t = \frac{1}{2}$, $r_t = 1$, and choose $s_t = \frac{1}{4}\gamma_t$ and $\eta_t = \eta > 0$, then (41) leads to the following recursive update

$$\gamma_m := \frac{1}{L\eta}, \quad \text{and} \quad \gamma_t := \frac{1}{L\big[\eta + \frac{(n-\hat{b})}{\hat{b}(n-1)}(2 + \frac{\eta^2}{2})L\sum_{j=t+1}^m \gamma_j\big]}, \quad t = 0, \cdots, m-1. \tag{42}$$

This update shows that $0 < \gamma_0 < \gamma_1 < \cdots < \gamma_m = \frac{1}{L\eta}$. Obviously, (42) shows that we can obtain larger step-size γ_t when the mini-batch size $\hat{b}$ is large.

Note that the mini-batch variant of Algorithm 1 in Theorem 7 using the update rules (42) still has convergence guarantee as in Theorem 7. However, we omit the details here.

5. Numerical experiments

We present three numerical examples to illustrate our theory and compare our methods with state-of-the-art algorithms in the literature. We implement 8 different variants of our ProxSARAH algorithm:

- ProxSARAH: Single sample and fixed step-sizes $\gamma := \frac{\sqrt{2}}{L\sqrt{3m}}$ and $\eta := \frac{2\sqrt{3m}}{4\sqrt{3m}+\sqrt{2}}$.
- ProxSARAH-A-v1: Single sample and adaptive step-sizes in Subsection 4.2.1.
- ProxSARAH-v1: Fixed $\gamma := 0.95$ and mini-batch size $\hat{b} := \frac{\sqrt{n}}{C}$ and $m := \sqrt{n}$.
- ProxSARAH-v2: Fixed $\gamma := 0.99$ and mini-batch size $\hat{b} := \frac{\sqrt{n}}{C}$ and $m := \sqrt{n}$.
- ProxSARAH-v3: Fixed $\gamma := 0.95$ and mini-batch size $\hat{b} := \frac{n^{\frac{1}{3}}}{C}$ and $m := n^{\frac{1}{3}}$.
- ProxSARAH-v4: Fixed $\gamma := 0.99$ and mini-batch size $\hat{b} := \frac{n^{\frac{1}{3}}}{C}$ and $m := n^{\frac{1}{3}}$.
- ProxSARAH-A-v2: Fixed $\gamma_m := 0.99$ and mini-batch size $\hat{b} := \sqrt{n}$ and $m := \sqrt{n}$.
- ProxSARAH-A-v3: Fixed $\gamma_m := 0.99$ and mini-batch size $\hat{b} := n^{\frac{1}{3}}$ and $m := n^{\frac{1}{3}}$.

We also implement 4 other algorithms:

- ProxSVRG: The proximal SVRG algorithm in (Reddi et al., 2016b) for single sample with theoretical step-size $\eta = \frac{1}{3nL}$, and for the mini-batch case with $\hat{b} := n^{2/3}$, the epoch length $m := n^{1/3}$, and the step-size $\eta := \frac{1}{3L}$.
- ProxSpiderBoost: The proximal SpiderBoost method in (Wang et al., 2018) with $\hat{b} := \sqrt{n}$, $m := \sqrt{n}$, and step-size $\eta := \frac{1}{2L}$.
- ProxSGD: Proximal Stochastic Gradient Descent scheme (Ghadimi and Lan, 2013) with step-size $\eta_t := \frac{\eta_0}{1+\tilde{\eta}\lfloor t/n \rfloor}$, where $\eta_0 > 0$ and $\tilde{\eta} \geq 0$ given in each example.
- ProxGD: Standard Proximal Gradient Descent algorithm with step-size $\eta := \frac{1}{L}$.

All the algorithms are implemented in Python running on a single node of a Linux server (called Longleaf) with configuration: 2.50GHz Intel processors, 30M cache, and 256GB RAM. For the last example, we implement these algorithms in **TensorFlow** (https://www.tensorflow.org) running on a GPU system. To be fair for comparison, we compute the norm of gradient mapping $\|G_\eta(w_t^{(s)})\|$ for visualization at the same value $\eta := 0.5$ in all methods. In the first two examples, we run all algorithms with 30 epochs, but in the last example, we increase it up to 150 epochs.

5.1 Nonnegative principal component analysis

We reconsider the problem of non-negative principal component analysis (NN-PCA) studied in (Reddi et al., 2016b). More precisely, for a given set of samples $\{z_i\}_{i=1}^n$ in $\mathbb{R}^d$, we solve the following problem:

$$f^\star := \min_{w\in\mathbb{R}^d}\left\{f(w) := -\frac{1}{2n}\sum_{i=1}^n w^\top(z_iz_i^\top)w \mid \|w\| \leq 1,\ w \geq 0\right\}. \tag{43}$$

By defining $f_i(w) := -\frac{1}{2}w^\top(z_iz_i^\top)w$ for $i = 1,\cdots,n$, and $\psi(w) := \delta_{\mathcal{X}}(w)$, the indicator of $\mathcal{X} := \left\{w \in \mathbb{R}^d \mid \|w\| \leq 1, w \geq 0\right\}$, we can write (43) into (2).

We test all the algorithms on three different well-known datasets: `mnist` ($n = 60000$, $d = 784$), `rcv1-binary` ($n = 20242$, $d = 47236$), and `real-sim` ($n = 72309$, $d = 20958$). In ProxSGD, we set $\eta_0 := 0.1$ and $\tilde{\eta} := 1.0$ that allow us to obtain good performance.

We first verify our theory by running 5 algorithms with single sample (i.e. $\hat{b} = 1$). The relative objective residuals and the norm of gradient mappings of these algorithms after 30 epochs are plotted in Figure 1.

Figure 1 indicates that both ProxSARAH and its adaptive variant work really well and dominate all other methods. ProxSARAH-A-v1 is still better than ProxSARAH. ProxSVRG is slow since its theoretical step-size $\frac{1}{3nL}$ is too small.

Now, we consider the mini-batch case. In this test, we run all the mini-batch variants of the methods described above. The relative objective residuals and the norms of gradient mapping are plotted in Figure 2.

From Figure 2, we observe that ProxSpiderBoost works well since it has a large step-size $\eta = \frac{1}{2L}$. However, ProxSARAH-A-v2 even performs better. Although ProxSVRG takes $\eta = \frac{1}{3L}$, its batch size and m also affect the performance resulting in a slower

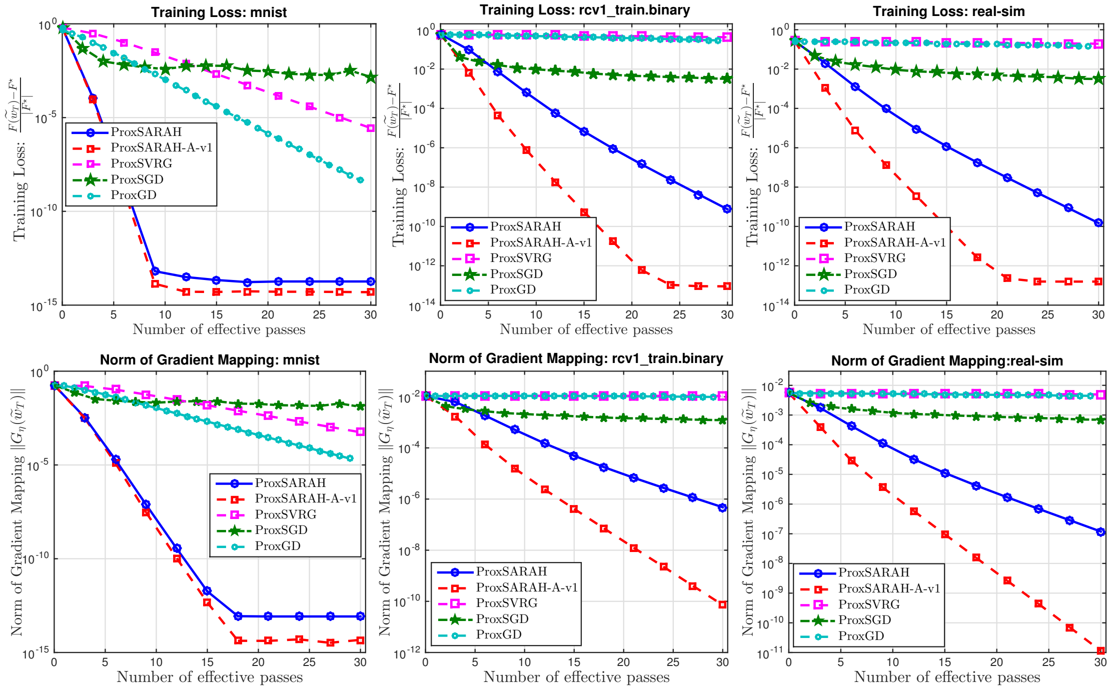

Figure 1: *The objective value residuals and gradient mapping norms of* (43) *on three datasets:* `mnist`*,* `rcv1-binary`*, and* `real-sim`*.*

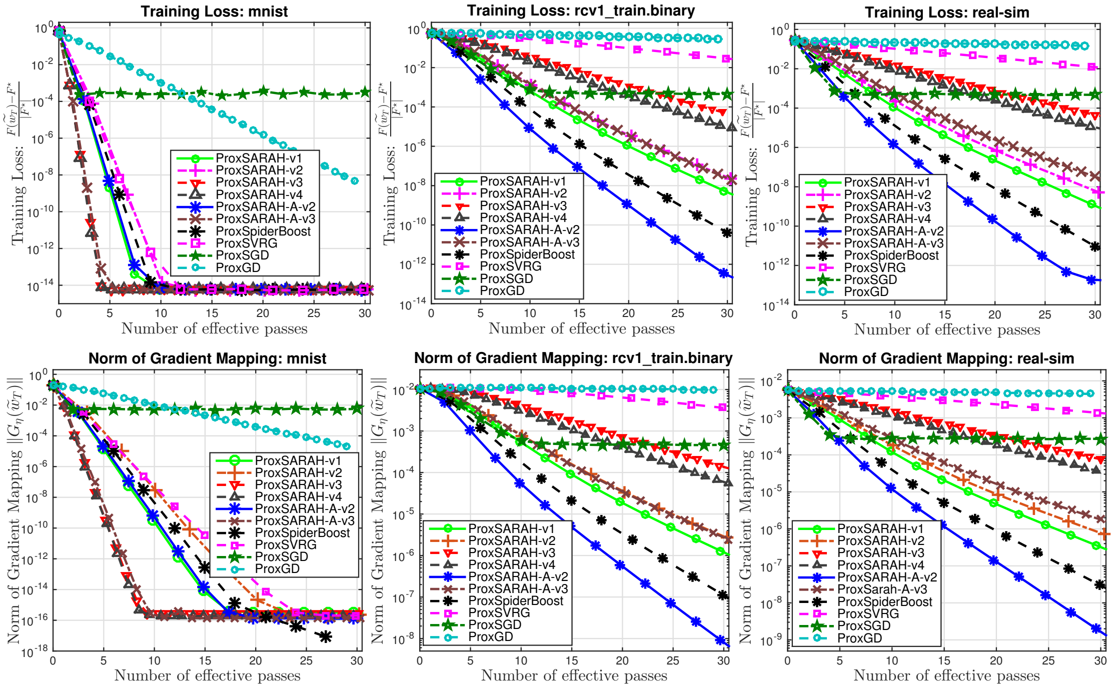

Figure 2: *The relative objective residuals and the norms of gradient mappings of 10 algorithms for solving* (43) *on three datasets:* `mnist`*,* `rcv1-binary`*, and* `real-sim`*.*

convergence. Other variants of ProxSARAH also work well, but ProxSARAH-v3 and ProxSARAH-v4 are slower than other variants. ProxSGD works well but then it is saturated around 10^{-4} accuracy. As predicted, ProxGD is inefficient in this example.

5.2 Sparse binary classification with nonconvex losses

We consider the following sparse binary classification involving nonconvex loss function:

$$\min_{w\in\mathbb{R}^d}\left\{F(w) := \frac{1}{n}\sum_{i=1}^{n}\ell(a_i^{\top}w, b_i) + \lambda\|w\|_1\right\}, \tag{44}$$

where $\{(a_i, b_i)\}_{i=1}^{n} \subset \mathbb{R}^d \times \{-1, 1\}^n$ is a given training dataset, $\lambda > 0$ is a regularization parameter, and $\ell(\cdot,\cdot)$ is a given smooth and nonconvex loss function as studied in (Zhao et al., 2010). By setting $f_i(w) := \ell(a_i^{\top}w, b_i)$ and $\psi(w) := \lambda\|w\|_1$ for $i = 1, \cdots, n$, we obtain the form (2).

The loss function ℓ is chosen from one of the following three cases:

1. $\ell_1(s,\tau) := 1 - \tanh(\omega\tau s)$ for a given $\omega > 0$. Since $|\frac{d^2\ell_1(s,\tau)}{ds^2}| \le \frac{8(2+\sqrt{3})(1+\sqrt{3})\omega^2\tau^2}{(3+\sqrt{3})^2}$ and $|\tau| = 1$, we can show that $\ell_1(\cdot,\tau)$ is L-smooth with respect to s, where $L := \frac{8(2+\sqrt{3})(1+\sqrt{3})\omega^2}{(3+\sqrt{3})^2} \approx 0.7698\omega^2$.
2. $\ell_2(s,\tau) := \left(1 - \frac{1}{1+\exp(-\tau s)}\right)^2$. For this function, we have $|\frac{d^2\ell_2(s,\tau)}{ds^2}| \le 0.15405\tau^2$. If $|\tau| = 1$, then this function is also L-smooth with $L = 0.15405$.
3. $\ell_3(s,\tau) := \ln(1 + \exp(-\tau s)) - \ln(1 + \exp(-\tau s - \omega))$ for some $\omega > 0$. With $\omega = 1$, we have $|\frac{d^2\ell_3(s,\tau)}{ds^2}| \le 0.092372\tau^2$. Therefore, if $|\tau| = 1$, then this function is also L-smooth with $L = 0.092372$.

We test the above algorithms on four datasets: `w8a` ($n = 49749$, $d = 300$), `rcv1-binary`, `real-sim`, and `news20` ($n = 11314$, $d = 130107$). We set the regularization parameter $\lambda := \frac{0.1}{\sqrt{n}}$ in all the tests, which gives us relatively sparse solutions.

Figure 3 shows the relative objective residuals and the norms of gradient mapping on three datasets: `w8a`, `rcv1-binary`, and `real-sim` for the loss function $\ell_1(\cdot)$. Similar to the first example, ProxSARAH and its adaptive variant work well, whereas ProxSARAH-A-v1 works better. ProxSVRG is still slow due to small step-size. ProxGD is again inefficient within 30 epochs.

Now, we test the loss function $\ell_2(.)$ with the mini-batch variants using the same 3 datasets. Figure 4 shows the results of 10 algorithms on these datasets. We can see that ProxSARAH-A-v2 works best, ProxSpiderBoost also performs well. ProxSVRG seems to be better than before but remains slow. Note that ProxSARAH-A-v2 still preserves the optimal complexity $\mathcal{O}\left(n^{1/2}\varepsilon^{-2}\right)$ compared to $\mathcal{O}\left(n + n^{1/2}\varepsilon^{-2}\right)$ of ProxSpiderBoost.

Finally, we test the loss function $\ell_3(\cdot)$ but using a larger dataset: `news20` ($n = 11314$, $d = 130107$). Figure 5 reveals the results of different algorithms on this `news20` dataset. Again, in the single sample case, ProxSARAH-A-v1 performs best, but ProxSVRG is even slower than ProxGD. In the mini-batch case, ProxSARAH-A-v2 works best. ProxSARAH-A-v3 and ProxSpiderBoost have similar performance. ProxSVRG are comparable with ProxSARAH-v1 and ProxSARAH-v2.

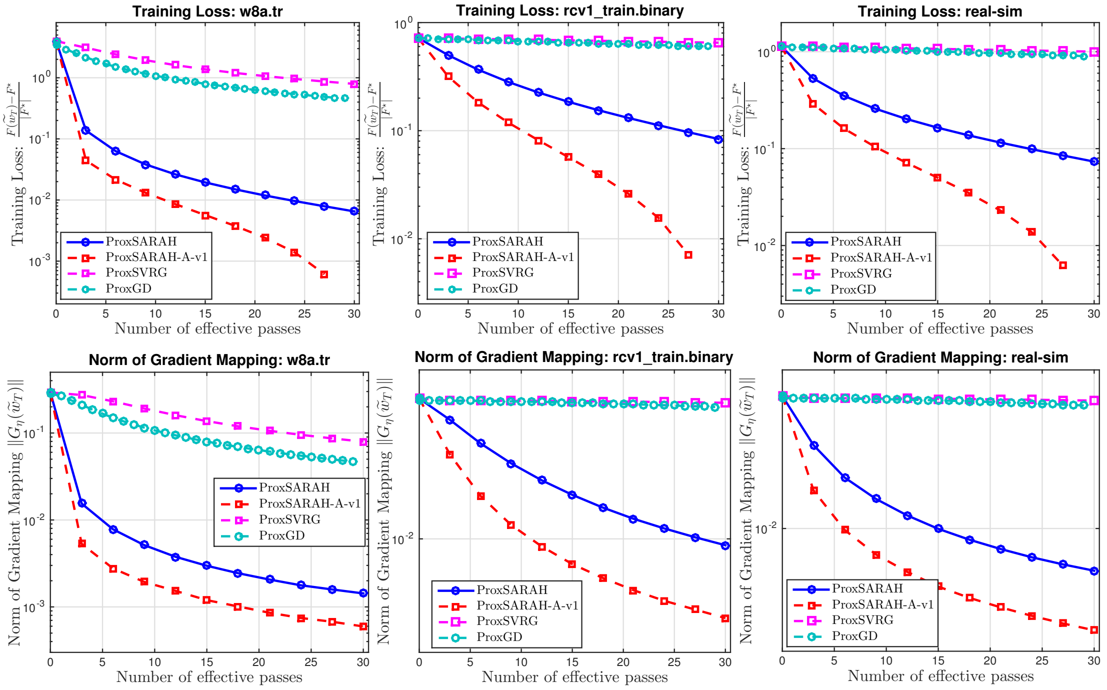

Figure 3: *The relative objective residuals and gradient mapping norms of* (44) *on three datasets using the loss* $\ell_1(s,\tau)$.

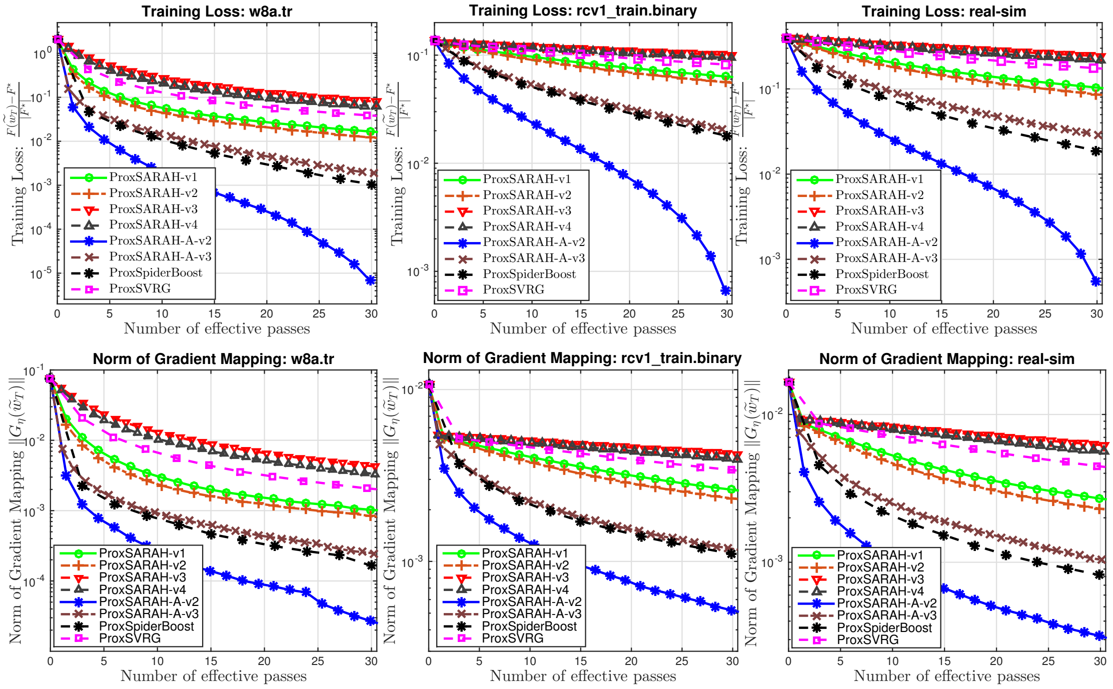

Figure 4: *The relative objective residuals and gradient mapping norms of* (44) *on three datasets using the loss* $\ell_2(s,\tau)$.

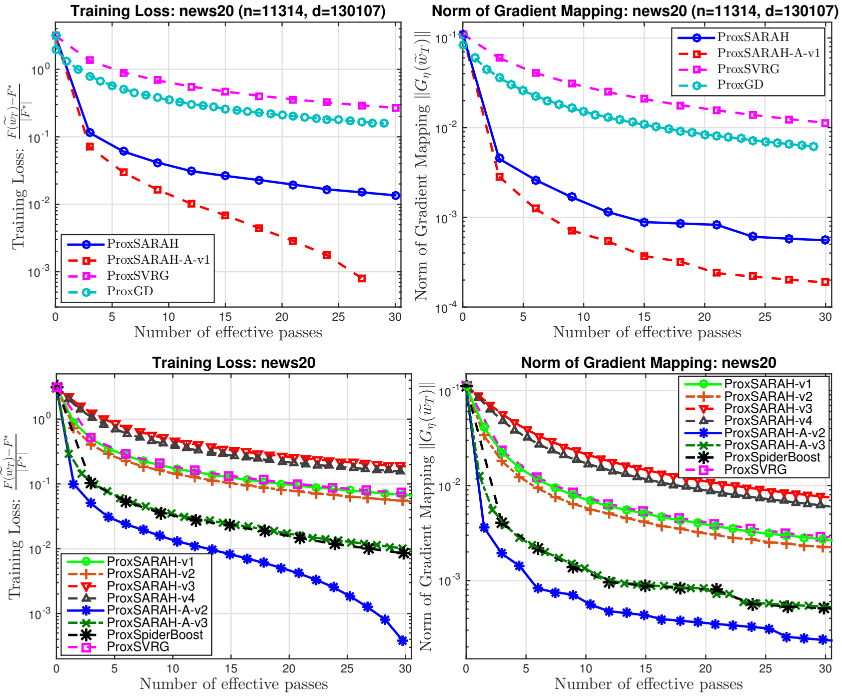

Figure 5: *The relative objective residuals and gradient mapping norms of* (44) *on* `news20` *using the loss* $\ell_3(s,\tau)$.

5.3 Feedforward Neural Network Training problem

We consider the following composite nonconvex optimization model arising from a feedforward neural network configuration:

$$\min_{w\in\mathbb{R}^d}\left\{F(w) := \frac{1}{n}\sum_{i=1}^{n}\ell\big(h(w,a_i),b_i\big)+\psi(w)\right\}, \tag{45}$$

where we concatenate all the weight matrices and bias vectors of the neural network in one vector of variable w, $\{(a_i,b_i)\}_{i=1}^n$ is a training dataset, $h(\cdot)$ is a composition between all linear transforms and activation functions as $h(w,a) := \boldsymbol{\sigma}_l(W_l\boldsymbol{\sigma}_{l-1}(W_{l-1}\boldsymbol{\sigma}_{l-2}(\cdots\boldsymbol{\sigma}_0(W_0a+\mu_0)\cdots)+\mu_{l-1})+\mu_l)$, where W_i is a weight matrix, μ_i is a bias vector, $\boldsymbol{\sigma}_i$ is an activation function, l is the number of layers, ℓ is a soft-max cross-entropy loss, and ψ is a convex regularizer (e.g., $\psi(w) := \lambda\|w\|_1$ for some $\lambda > 0$ to obtain sparse weights). Again, by defining $f_i(w) := \ell(h(w,a_i),b_i)$ for $i = 1,\cdots,n$, we obtain the same form as (2).

We implement our algorithms and other methods in TensorFlow and use two datasets `mnist` and `fashion_mnist` to compare their performance. In the first test, we use a one-hidden layer neural network: $784\times 100\times 10$ for both `mnist` and `fashion_mnist`. The activation function $\boldsymbol{\sigma}_i$ of the hidden layer is ReLU and the loss function is the soft-max cross-entropy. To estimate the Lipschitz constant L, we normalize the input data. The regularization parameter λ is set at $\lambda := \frac{0.1}{\sqrt{n}}$ and $\psi(\cdot) := \|\cdot\|_1$.

We first test ProxSARAH, ProxSVRG, ProxSpiderBoost, and ProxSGD by taking the learning rates based on theory and using mini-batch. For ProxSGD, we use the mini-batch $\hat{b} = 200$, $\eta_0 = 0.5$, and $\tilde{\eta} = 0.05$. For the `mnist` dataset, to verify the theory, we use $L = 1$, and set the learning rate for ProxSVRG at $\eta = \frac{1}{3L} = \frac{1}{3}$, and for ProxSpiderBoost at $\eta = \frac{1}{2L} = \frac{1}{2}$ as suggested by the theory. For ProxSARAH, we choose η and γ as suggested in Subsection 3.2.3. However, for the `fashion_mnist` dataset, it requires a smaller learning rate. Therefore, we choose $L = 5$ for ProxSARAH and follow the theory in Subsection 3.2.3 to set η, γ, m, and $\hat{b}$. We also tune the learning rate for ProxSVRG and ProxSpiderBoost until they are stabilized to obtain the best possible step-size in this example as $\eta_{\text{ProxSVRG}} = 0.5$ and $\eta_{\text{ProxSpiderBoost}} = 0.2$, respectively.

Figure 6 shows the convergence of different variants of ProxSARAH, ProxSpiderBoost, ProxSVRG, and ProxSGD on three criteria for `mnist`: training loss values, the norm of gradient mapping, and the test accuracy.

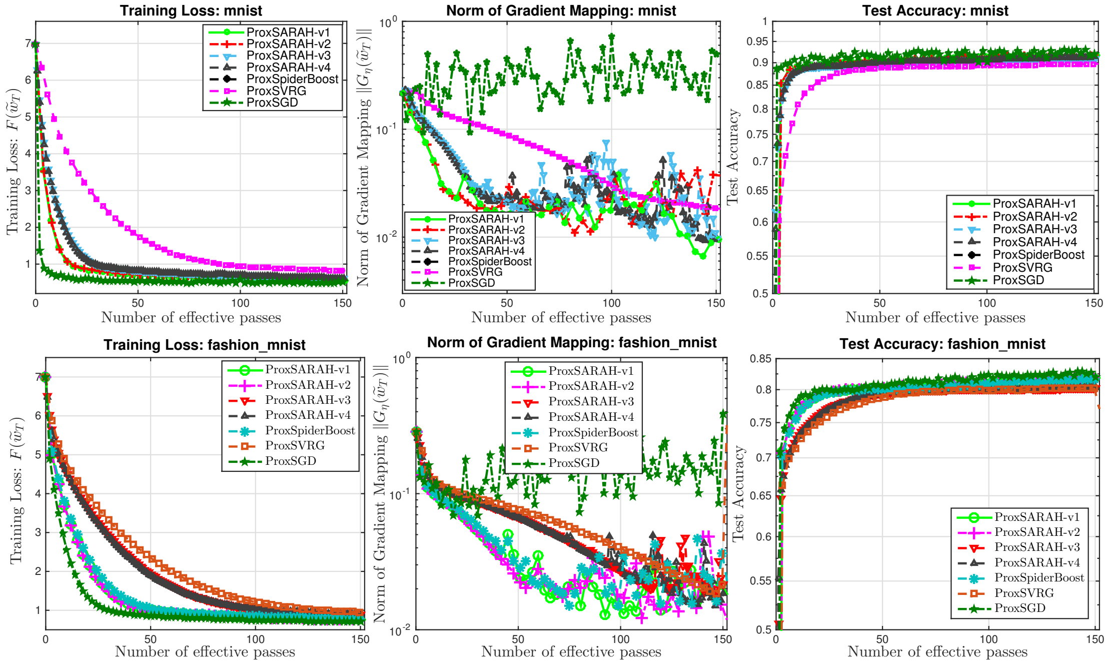

Figure 6: *The training loss, gradient mapping, and test accuracy on* `mnist` *(top line) and* `fashion_mnist` *(bottom line) of* 7 *algorithms.*

In this example, ProxSGD appears to be the best in terms of training loss and test accuracy. However, the norm of gradient mapping is rather different from others, relatively large, and oscillated. ProxSVRG is clearly slower than ProxSpiderBoost due to smaller learning rate. The four variants of ProxSARAH perform relatively well, but the first and second variants seem to be slightly better. Note that the norm of gradient mapping tends to be decreasing but still oscillated since perhaps we are taking the last iterate instead of a random choice of intermediate iterates as stated in the theory.

Next, we increase the number of hidden layers in our model to 2 and the new network becomes 784 × 300 × 100 × 10. We again test these algorithms on both `mnist` and `fashion_mnist` datasets and the results are plotted in Figure 7. We choose $L = 15$ for `mnist` and $L = 30$ for `fashion_mnist` in ProxSARAH and then follow the configuration in Subsection 3.2.3 for ProxSARAH. We also tune both ProxSVRG and ProxSpiderBoost to obtain the best possible step-sizes: $\eta_{\text{ProxSVRG}} = 0.225$ and $\eta_{\text{ProxSpiderBoost}} = 0.09$ for `mnist`, and $\eta_{\text{ProxSVRG}} = 0.12$ and $\eta_{\text{ProxSpiderBoost}} = 0.06$. For ProxSGD, we use $\eta_0 = 0.0333$ and $\tilde{\eta} = 0.5$ in both `mnist` and `fashion_mnist`.

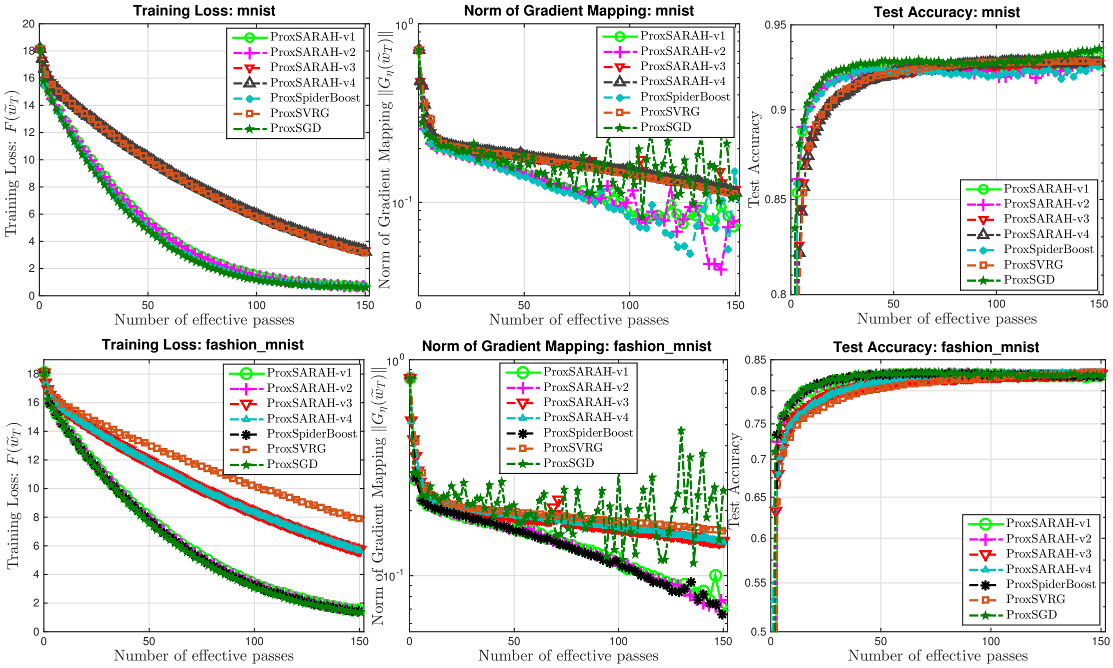

Figure 7: *The training loss, gradient mapping, and test accuracy on `mnist` (top line) and `fashion_mnist` (bottom line) of 7 algorithms.*

As we can see from Figure 7 that ProxSGD still makes a good progress on the training loss and the test accuracy, but it is very oscillated on the norms of gradient mapping in both datasets. ProxSVRG is slower than others in both cases even with tuned learning rate. ProxSpiderBoost performs comparably with ProxSARAH-v1 and -v2. This is clearly understandable since ProxSpiderBoost uses the same SARAH estimator and a large learning rate η.

Acknowledgements

We would like to acknowledge support for this project from the National Science Foundation (NSF grant DMS-1619884).

Appendix A. The proof of technical results in the main text

In this appendix, we provide the full proofs of technical results in the main text.

A.1 The proof of Lemma 4: One-iteration analysis for single sample case

From the update $w_{t+1}^{(s)} := (1-\gamma_t)w_t^{(s)} + \gamma_t\widehat{w}_{t+1}^{(s)}$, we have $w_{t+1}^{(s)} - w_t^{(s)} = \gamma_t(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})$. Firstly, using the L-smoothness of f from (6) of Assumption 2.2, we can derive

$$\begin{array}{ll} f(w_{t+1}^{(s)}) & \leq f(w_t^{(s)}) + \langle \nabla f(w_t^{(s)}), w_{t+1}^{(s)} - w_t^{(s)}\rangle + \frac{L}{2}\Vert w_{t+1}^{(s)} - w_t^{(s)}\Vert^2 \\ & = f(w_t^{(s)}) + \gamma_t\langle \nabla_w f(w_t^{(s)}), \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle + \frac{L\gamma_t^2}{2}\Vert \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\Vert^2. \end{array} \tag{46}$$

Next, using the convexity of ψ, one can show that

$$\psi(w_{t+1}^{(s)}) \leq (1-\gamma_t)\psi(w_t^{(s)}) + \gamma_t\psi(\widehat{w}_{t+1}^{(s)}) \leq \psi(w_t^{(s)}) + \gamma_t\langle \nabla\psi(\widehat{w}_{t+1}^{(s)}), \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle, \tag{47}$$

where $\nabla\psi(\widehat{w}_{t+1}^{(s)}) \in \partial\psi(\widehat{w}_{t+1}^{(s)})$.

By the optimality condition of $\widehat{w}_{t+1}^{(s)} := \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)})$, we have $\nabla\psi(\widehat{w}_{t+1}^{(s)}) = -v_t^{(s)} - \frac{1}{\eta_t}(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})$ for some $\nabla\psi(\widehat{w}_{t+1}^{(s)}) \in \partial\psi(\widehat{w}_{t+1}^{(s)})$. Substituting this expression into (47), we obtain

$$\psi(w_{t+1}^{(s)}) \leq \psi(w_t^{(s)}) + \gamma_t\langle v_t^{(s)}, w_t^{(s)} - \widehat{w}_{t+1}^{(s)}\rangle - \frac{\gamma_t}{\eta_t}\Vert \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\Vert^2. \tag{48}$$

Combining (46) and (48), and then using $F(w) := f(w) + \psi(w)$ yield

$$F(w_{t+1}^{(s)}) \leq F(w_t^{(s)}) + \gamma_t\langle \nabla f(w_t^{(s)}) - v_t^{(s)}, \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle - \Big(\frac{\gamma_t}{\eta_t} - \frac{L\gamma_t^2}{2}\Big)\Vert \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\Vert^2. \tag{49}$$

Now, for any $c_t > 0$, we have

$$\begin{array}{ll} \langle \nabla f(w_t^{(s)}) - v_t^{(s)}, \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\rangle & = \frac{1}{2c_t}\Vert \nabla f(w_t^{(s)}) - v_t^{(s)}\Vert^2 + \frac{c_t}{2}\Vert \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\Vert^2 \\ & \quad - \frac{1}{2c_t}\Vert \nabla f(w_t^{(s)}) - v_t^{(s)} - c_t(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})\Vert^2. \end{array}$$

Utilizing this inequality, we can rewrite (49) as

$$F(w_{t+1}^{(s)}) \leq F(w_t^{(s)}) + \frac{\gamma_t}{2c_t}\Vert \nabla f(w_t^{(s)}) - v_t^{(s)}\Vert^2 - \Big(\frac{\gamma_t}{\eta_t} - \frac{L\gamma_t^2}{2} - \frac{\gamma_t c_t}{2}\Big)\Vert \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\Vert^2 - \sigma_t^{(s)},$$

where $\sigma_t^{(s)} := \frac{\gamma_t}{2c_t}\Vert \nabla f(w_t^{(s)}) - v_t^{(s)} - c_t(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})\Vert^2 \geq 0$.

Taking expectation both sides of this inequality over the entire history, we obtain

$$\begin{array}{ll} \mathbb{E}\Big[F(w_{t+1}^{(s)})\Big] & \leq \mathbb{E}\Big[F(w_t^{(s)})\Big] + \frac{\gamma_t}{2c_t}\mathbb{E}\Big[\Vert \nabla f(w_t^{(s)}) - v_t^{(s)}\Vert^2\Big] \\ & \quad - \Big(\frac{\gamma_t}{\eta_t} - \frac{L\gamma_t^2}{2} - \frac{\gamma_t c_t}{2}\Big)\mathbb{E}\Big[\Vert \widehat{w}_{t+1}^{(s)} - w_t^{(s)}\Vert^2\Big] - \mathbb{E}\Big[\sigma_t^{(s)}\Big]. \end{array} \tag{50}$$

Next, recall from (10) that $G_\eta(w) := \frac{1}{\eta}\big(w - \mathrm{prox}_{\eta\psi}(w - \eta\nabla f(w))\big)$ is the gradient mapping of F. In this case, it is obvious that

$$\eta_t\Vert G_{\eta_t}(w_t^{(s)})\Vert = \Vert w_t^{(s)} - \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t\nabla f(w_t^{(s)}))\Vert.$$

Using this definition, the triangle inequality, and the nonexpansive property $\|\mathrm{prox}_{\eta\psi}(z)-\mathrm{prox}_{\eta\psi}(w)\| \leq \|z-w\|$ of $\mathrm{prox}_{\eta\psi}$, we can derive that

$$\begin{aligned}
\eta_t\|G_{\eta_t}(w_t^{(s)})\| &\leq \|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\| + \|\mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t\nabla f(w_t^{(s)})) - \widehat{w}_{t+1}^{(s)}\|\\
&= \|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\| + \|\mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t\nabla f(w_t^{(s)})) - \mathrm{prox}_{\eta_t\psi}(w_t^{(s)} - \eta_t v_t^{(s)})\|\\
&\leq \|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\| + \eta_t\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|.
\end{aligned}$$

Now, for any $r_t > 0$, the last estimate leads to

$$\eta_t^2\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] \leq \left(1+\tfrac{1}{r_t}\right)\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] + (1+r_t)\eta_t^2\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right].$$

Multiplying this inequality by $\frac{s_t}{2} > 0$ and adding the result to (50), we finally get

$$\begin{aligned}
\mathbb{E}\left[F(w_{t+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_t^{(s)})\right] - \tfrac{s_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right]\\
&+ \tfrac{1}{2}\left[\tfrac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2\right]\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right]\\
&- \tfrac{1}{2}\left[\tfrac{2\gamma_t}{\eta_t} - L\gamma_t^2 - \gamma_t c_t - s_t\left(1+\tfrac{1}{r_t}\right)\right]\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] - \mathbb{E}\left[\sigma_t^{(s)}\right].
\end{aligned}$$

Summing up this inequality from $t = 0$ to $t = m$, we obtain

$$\begin{aligned}
\mathbb{E}\left[F(w_{m+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_0^{(s)})\right] + \frac{1}{2}\sum_{t=0}^{m}\Big[\frac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2\Big]\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right]\\
&- \frac{1}{2}\sum_{t=0}^{m}\Big[\frac{2\gamma_t}{\eta_t} - L\gamma_t^2 - \gamma_t c_t - s_t\Big(1+\frac{1}{r_t}\Big)\Big]\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \qquad (51)\\
&- \sum_{t=0}^{m}\frac{s_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] - \sum_{t=0}^{m}\mathbb{E}\left[\sigma_t^{(s)}\right].
\end{aligned}$$

By the definition of $v_t^{(s)}$ and Assumption 2.2, we can show that $\|v_j^{(s)} - v_{j-1}^{(s)}\| = \|\nabla f(w_j^{(s)};\xi_j) - \nabla f(w_{j-1}^{(s)};\xi_j)\| \leq L\|w_j^{(s)} - w_{j-1}^{(s)}\|$. Using this inequality and (19), we can derive

$$\begin{aligned}
\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right] &\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + \textstyle\sum_{j=1}^t\mathbb{E}\left[\|v_j^{(s)} - v_{j-1}^{(s)}\|^2\right]\\
&\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right]\\
&+ \textstyle\sum_{j=1}^t\mathbb{E}\left[\|\nabla f(w_j^{(s)};\xi_j) - \nabla f(w_{j-1}^{(s)};\xi_j)\|^2\right] \qquad (52)\\
&\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + L^2\textstyle\sum_{j=1}^t\mathbb{E}\left[\|w_j^{(s)} - w_{j-1}^{(s)}\|^2\right]\\
&= \bar{\sigma}^{(s)} + L^2\textstyle\sum_{j=1}^t\gamma_{j-1}^2\mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right],
\end{aligned}$$

where $\bar{\sigma}^{(s)} := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \geq 0$ and $w_j^{(s)} - w_{j-1}^{(s)} = \gamma_{j-1}(\widehat{w}_j^{(s)} - w_{j-1}^{(s)})$.

Substituting the estimate (52) into (51), we finally arrive at

$$\begin{aligned}\mathbb{E}\left[F(w_{m+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_0^{(s)})\right] + \frac{L^2}{2}\sum_{t=0}^m\left[\frac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2\right]\sum_{j=1}^t\gamma_{j-1}^2\mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right]\\ &- \frac{1}{2}\sum_{t=0}^m\left[\frac{2\gamma_t}{\eta_t} - L\gamma_t^2 - \gamma_t c_t - s_t\Big(1+\frac{1}{r_t}\Big)\right]\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right]\\ &- \sum_{t=0}^m\frac{s_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] - \sum_{t=0}^m\mathbb{E}\left[\sigma_t^{(s)}\right]\\ &+ \frac{1}{2}\sum_{t=0}^m\left[\frac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2\right]\bar{\sigma}^{(s)},\end{aligned}$$

which is exactly (23). □

A.2 The proof of Lemma 5: Parameter selection - The constant case

Let us first fix all the parameters and step-sizes as constants as follows:

$$c_t := c > 0,\ \ \gamma_t := \gamma \in (0,1],\ \ \eta_t := \eta > 0,\ \ r_t := r > 0,\ \ \text{and}\ \ s_t := \gamma > 0.$$

We also denote $a_t^{(s)} := \mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \geq 0$.
Using these expressions into (23), we can easily show that

$$\begin{aligned}\mathbb{E}\left[F(w_{m+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_0^{(s)})\right] + \tfrac{L^2\gamma^3}{2}\left[\tfrac{1}{c} + (1+r)\eta^2\right]\sum_{t=0}^m\sum_{j=1}^t a_{j-1}^{(s)}\\ &- \tfrac{\gamma}{2}\left[\tfrac{2}{\eta} - L\gamma - c - \left(1+\tfrac{1}{r}\right)\right]\sum_{t=0}^m a_t^{(s)} - \tfrac{\gamma\eta^2}{2}\sum_{t=0}^m\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right]\\ &+ \tfrac{\gamma}{2}\left[\tfrac{1}{c} + (1+r)\eta^2\right](m+1)\bar{\sigma}^{(s)} - \sum_{t=0}^m\mathbb{E}\left[\sigma_t^{(s)}\right]\\ &= \mathbb{E}\left[F(w_0^{(s)})\right] - \tfrac{\gamma\eta^2}{2}\sum_{t=0}^m\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] - \sum_{t=0}^m\mathbb{E}\left[\sigma_t^{(s)}\right]\\ &+ \tfrac{\gamma}{2}\left[\tfrac{1}{c} + (1+r)\eta^2\right](m+1)\bar{\sigma}^{(s)} + \mathcal{T}_m,\end{aligned} \tag{53}$$

where $\mathcal{T}_m$ is defined as

$$\mathcal{T}_m := \frac{L^2\gamma^3}{2}\left[\frac{1}{c} + (1+r)\eta^2\right]\sum_{t=0}^m\sum_{j=1}^t a_{j-1}^{(s)} - \frac{\gamma}{2}\left[\frac{2}{\eta} - L\gamma - c - \left(1+\frac{1}{r}\right)\right]\sum_{t=0}^m a_t^{(s)}.$$

Our goal is to choose $c > 0$, $r > 0$, $\eta > 0$, and $\gamma \in (0,1]$ such that $\mathcal{T}_m \leq 0$. We first rewrite $\mathcal{T}_m$ as follows:

$$\begin{aligned}\mathcal{T}_m &= \tfrac{L^2\gamma^3}{2}\left[\tfrac{1}{c} + (1+r)\eta^2\right]\left[ma_0^{(s)} + (m-1)a_1^{(s)} + \cdots + 2a_{m-2}^{(s)} + a_{m-1}^{(s)}\right]\\ &- \tfrac{\gamma}{2}\left[\tfrac{2}{\eta} - L\gamma - c - \left(1+\tfrac{1}{r}\right)\right]\left[a_0^{(s)} + a_1^{(s)} + \cdots + a_m^{(s)}\right].\end{aligned}$$

By synchronizing the coefficients of the terms $a_0^{(s)}, a_1^{(s)}, \cdots, a_m^{(s)}$, to guarantee $\mathcal{T}_m \leq 0$, we need to choose

$$\begin{cases} mL^2\gamma^2\left[\frac{1}{c} + (1+r)\eta^2\right] - \left[\frac{2}{\eta} - L\gamma - c - \left(1+\frac{1}{r}\right)\right] & \leq 0, \\ \frac{2}{\eta} - L\gamma - c - (1+\frac{1}{r}) & > 0. \end{cases} \tag{54}$$

Assume that $\frac{2}{\eta} - L\gamma - c - (1+\frac{1}{r}) = 1 > 0$. This implies that $\eta = \frac{2}{L\gamma + q}$, where $q := 2 + c + \frac{1}{r} > 2$. Next, since

$$mL^2\gamma^2\left[\frac{1}{c} + (1+r)\eta^2\right] - \left[\frac{2}{\eta} - L\gamma - c - \left(1+\frac{1}{r}\right)\right] = mL^2\gamma^2\Big[\frac{1}{c} + \frac{4(1+r)}{(L\gamma+q)^2}\Big] - 1,$$

and $L\gamma > 0$, we can overestimate the first inequality of (54) as

$$mL^2\gamma^2\left[\frac{1}{c} + (1+r)\eta^2\right] - \left[\frac{2}{\eta} - L\gamma - c - \left(1+\frac{1}{r}\right)\right] \leq mL^2\gamma^2\Big[\frac{1}{c} + \frac{4(1+r)}{q^2}\Big] - 1 = 0.$$

The last equation and $\eta = \frac{2}{L\gamma+q}$ lead to

$$\gamma := \frac{1}{L\sqrt{\omega m}} \quad \text{and} \quad \eta := \frac{2\sqrt{\omega m}}{q\sqrt{\omega m}+1},$$

which is exactly (25), where $\omega := \frac{1}{c} + \frac{4(1+r)}{q^2} = \frac{1}{c} + \frac{4r^2(1+r)}{((2+c)r+1)^2}$. In particular, if we choose $r = c = 1$, then $\omega = \frac{3}{2}$.

Finally, using this choice (25) of the step-sizes, we can derive that

$$\mathbb{E}\left[F(w_{m+1}^{(s)}\right] \leq \mathbb{E}\left[F(w_0^{(s)})\right] - \frac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_\eta(w_t^{(s)})\|^2\right] - \sum_{t=0}^{m}\mathbb{E}\left[\sigma_t^{(s)}\right] + \frac{\gamma\theta}{2}(m+1)\bar{\sigma}^{(s)}, \tag{55}$$

which is exactly (26), where $\theta := \frac{1}{c} + \frac{4(1+r)}{(L\gamma+q)^2} = \frac{1}{c} + \frac{4r^2(1+r)\omega m}{[r+((2+c)r+1)\sqrt{\omega m}]^2}$. □

A.3 The proof of Theorem 7: Mini-batch case

From (22) of Lemma 3, the L-smoothness in (4), the choice $\hat{b}_t = \hat{b} \geq 1$, and $w_j^{(s)} - w_{j-1}^{(s)} = \gamma_{j-1}(\widehat{w}_j^{(s)} - w_{j-1}^{(s)})$, we can estimate

$$\begin{aligned} \mathbb{E}\left[\|v_j^{(s)} - v_{j-1}^{(s)}\|^2 \mid \mathcal{F}_{j-1}\right] & \leq \tfrac{1}{\hat{b}}\left(\tfrac{n-\hat{b}}{n-1}\right)\tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathbb{E}\left[\|\nabla f_i(w_j^{(s)}) - \nabla f_i(w_{j-1}^{(s)})\|^2 \mid \mathcal{F}_{j-1}\right] \\ & \overset{(4)}{\leq} \tfrac{1}{\hat{b}}\left(\tfrac{n-\hat{b}}{n-1}\right)L^2\|w_j^{(s)} - w_{j-1}^{(s)}\|^2 \\ & = \tfrac{1}{\hat{b}}\left(\tfrac{n-\hat{b}}{n-1}\right)L^2\gamma_{j-1}^2\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2. \end{aligned}$$

Using this estimate, similar to the proof of (52) and taking full expectation, we have

$$\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right] \leq \bar{\sigma}^{(s)} + \tfrac{1}{\hat{b}}\left(\tfrac{n-\hat{b}}{n-1}\right)L^2\textstyle\sum_{j=1}^{t}\gamma_{j-1}^2\mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right], \tag{56}$$

where $\bar{\sigma}^{(s)} := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \geq 0$.

Utilizing the estimate (56) instead of (52) in the proof of Lemma 5, we have the following estimate:

$$\begin{aligned}
\mathbb{E}\left[F(w_{m+1}^{(s)})\right] \leq \mathbb{E}\left[F(w_0^{(s)})\right] &+ \tfrac{L^2(n-\hat{b})}{2\hat{b}(n-1)}\sum_{t=0}^{m}\left[\frac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2\right]\sum_{j=1}^{t}\gamma_{j-1}^2\mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right] \\
&- \tfrac{1}{2}\sum_{t=0}^{m}\left[\tfrac{2\gamma_t}{\eta_t} - L\gamma_t^2 - \gamma_t c_t - s_t\left(1 + \tfrac{1}{r_t}\right)\right]\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \\
&- \sum_{t=0}^{m}\tfrac{s_t\eta_t^2}{2}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] - \sum_{t=0}^{m}\mathbb{E}\left[\sigma_t^{(s)}\right] \\
&+ \tfrac{1}{2}\textstyle\sum_{t=0}^{m}\left[\tfrac{\gamma_t}{c_t} + (1+r_t)s_t\eta_t^2\right]\bar{\sigma}^{(s)},
\end{aligned}$$

If we choose $(\gamma_t, \eta_t, c_t, s_t, r_t) = (\gamma, \eta, c, \gamma, r) > 0$ for all t, then the last estimate leads to

$$\begin{aligned}
\mathbb{E}\left[F(w_{m+1}^{(s)})\right] \leq \mathbb{E}\left[F(w_0^{(s)})\right] &+ \frac{L^2(n-\hat{b})\gamma^3}{2\hat{b}(n-1)}\left[\frac{1}{c} + (1+r)\eta^2\right]\sum_{t=0}^{m}\sum_{j=1}^{t}\mathbb{E}\left[\|\widehat{w}_j^{(s)} - w_{j-1}^{(s)}\|^2\right] \\
&- \frac{\gamma}{2}\left[\frac{2}{\eta} - L\gamma - c - \left(1 + \frac{1}{r}\right)\right]\sum_{t=0}^{m}\mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] \\
&- \frac{\gamma\eta^2}{2}\sum_{t=0}^{m}\mathbb{E}\left[\|G_{\eta_t}(w_t^{(s)})\|^2\right] - \sum_{t=0}^{m}\mathbb{E}\left[\sigma_t^{(s)}\right] \\
&+ \frac{\gamma}{2}\left[\frac{1}{c} + (1+r)\eta^2\right](m+1)\bar{\sigma}^{(s)}. \qquad (57)
\end{aligned}$$

In order to telescope this inequality, we impose the following conditions for the parameters:

$$\begin{cases}
\frac{mL^2\gamma^2}{\hat{b}}\left(\frac{n-\hat{b}}{n-1}\right)\left[\frac{1}{c} + (1+r)\eta^2\right] - \left[\frac{2}{\eta} - L\gamma - c - \left(1 + \frac{1}{r}\right)\right] \leq 0, \\
\frac{2}{\eta} - L\gamma - c - (1 + \frac{1}{r}) > 0.
\end{cases} \qquad (58)$$

For simplicity of analysis, let us choose $c = r = 1$. With the same derivation as in Lemma 5, we obtain the following step-sizes

$$\gamma := \frac{1}{L\sqrt{\rho m}} \quad \text{and} \quad \eta := \frac{2\sqrt{\rho m}}{4\sqrt{\rho m} + 1},$$

which is exactly (28), where $\rho := \frac{3(n-\hat{b})}{2(n-1)\hat{b}}$.

By the condition (58), we can simplify (57) as

$$\begin{aligned}
\mathbb{E}\left[F(w_{m+1}^{(s)})\right] &\leq \mathbb{E}\left[F(w_0^{(s)})\right] - \tfrac{\gamma\eta^2}{2}\textstyle\sum_{t=0}^{m}\mathbb{E}\left[\|G_{\eta}(w_t^{(s)})\|^2\right] \\
&\quad - \textstyle\sum_{t=0}^{m}\mathbb{E}\left[\sigma_t^{(s)}\right] + \tfrac{\gamma\hat{\theta}}{2}(m+1)\bar{\sigma}^{(s)},
\end{aligned} \qquad (59)$$

where $\hat{\theta} := 1 + \frac{8\rho m}{(4\sqrt{\rho m}+1)^2}$.

With the same argument as in the proof of Theorem 6, we obtain

$$\frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_{\eta}(w_t^{(s)})\|^2\right] \leq \frac{2}{\gamma\eta^2(m+1)S}\big[F(\widetilde{w}_0) - F^{\star}\big] + \frac{\hat{\theta}\sigma_n^2}{\eta^2 S}\sum_{s=1}^{S}\frac{(n-b_s)}{nb_s}.$$

For $\widetilde{w}_T \sim \mathbf{U}\big(\{w_t^{(s)}\}_{t=0,s=1}^{t=m,s=S}\big)$ with $T := (m+1)S$ and $b_s = n$, the last estimate implies

$$\mathbb{E}\left[\|G_{\eta}(\widetilde{w}_T)\|^2\right] = \frac{1}{(m+1)S}\sum_{s=1}^{S}\sum_{t=0}^{m}\mathbb{E}\left[\|G_{\eta}(w_t^{(s)})\|^2\right] \leq \frac{2}{\gamma\eta^2(m+1)S}\big[F(\widetilde{w}_0) - F^{\star}\big].$$

By the update rule of η, we can easily show that $\eta \geq \frac{2}{5}$. To guarantee $\mathbb{E}\left[\|G_{\eta}(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$, we choose S and m such that $\frac{25}{2\gamma T}\big[F(\widetilde{w}_0) - F^{\star}\big] = \varepsilon^2$ leading to $T = \frac{25[F(\widetilde{w}_0)-F^{\star}]}{2\gamma\varepsilon^2} = \frac{25L\sqrt{\rho m}[F(\widetilde{w}_0)-F^{\star}]}{2\varepsilon^2}$. Note that we can choose $m := \lfloor n/\hat{b}\rfloor$, one can show that $\frac{n}{\hat{b}} - 1 \leq m \leq \frac{n}{\hat{b}}$ or $\frac{n-\hat{b}}{\hat{b}} \leq m \leq \frac{n}{\hat{b}}$, which leads to $\rho m \leq \frac{3n(n-\hat{b})}{2(n-1)\hat{b}^2}$. Substituting this into T, we can show that the number of iterations T can be upper bounded by $T \leq \frac{25L[F(\widetilde{w}_0)-F^{\star}]}{2\varepsilon^2}\sqrt{\frac{3n(n-\hat{b})}{2\hat{b}^2(n-1)}}$. Clearly, if $\hat{b} < n$, we have $T = \mathcal{O}\left(\frac{L\sqrt{n-\hat{b}}[F(\widetilde{w}_0)-F^{\star}]}{\hat{b}\varepsilon^2}\right)$. The number of gradient evaluations ∇f_i is at most

$$T_{\text{grad}} = S(n+2\hat{b}(m+1)) = Sn+2\hat{b}S(m+1) \leq 3\hat{b}S(m+1) = 3\hat{b}T = \mathcal{O}\left(\frac{L\sqrt{n-\hat{b}}[F(\widetilde{w}_0)-F^{\star}]}{\varepsilon^2}\right).$$

Moreover, the number of proximal operator calls $\text{prox}_{\eta\psi}$ is at most T. □

A.4 The proof of Theorem 9: The non-composite cases

Since $\psi = 0$, we have $\widehat{w}_{t+1}^{(s)} = w_t^{(s)} - \eta_t v_t^{(s)}$. Therefore, $\widehat{w}_{t+1}^{(s)} - w_t^{(s)} = -\eta_t v_t^{(s)}$ and $w_{t+1}^{(s)} = (1-\gamma_t)w_t^{(s)} + \gamma_t\widehat{w}_{t+1}^{(s)} = w_t^{(s)} - \gamma_t\eta_t v_t^{(s)} = w_t^{(s)} - \hat{\eta}_t v_t^{(s)}$, where $\hat{\eta}_t := \gamma_t\eta_t$. Using these relations and $c_t = \frac{1}{\eta_t}$, we can easily show that

$$\begin{cases} \mathbb{E}\left[\|\widehat{w}_{t+1}^{(s)} - w_t^{(s)}\|^2\right] = \eta_t^2\mathbb{E}\left[\|v_t^{(s)}\|^2\right], \\ \sigma_t^{(s)} := \frac{\gamma_t}{2c_t}\|\nabla f(w_t^{(s)}) - v_t^{(s)} - c_t(\widehat{w}_{t+1}^{(s)} - w_t^{(s)})\|^2 = \frac{\hat{\eta}_t}{2}\|\nabla f(w_t^{(s)})\|^2. \end{cases}$$

Substituting these estimates into (50) and noting that $f = F$ and $\hat{\eta}_t := \gamma_t\eta_t$, we obtain

$$\begin{aligned} \mathbb{E}\left[f(w_{t+1}^{(s)})\right] &\leq \mathbb{E}\left[f(w_t^{(s)})\right] + \tfrac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right] \\ &\quad - \tfrac{\hat{\eta}_t}{2}\big(1 - L\hat{\eta}_t\big)\mathbb{E}\left[\|v_t^{(s)}\|^2\right] - \tfrac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right]. \end{aligned} \tag{60}$$

On the other hand, from (19), by Assumption 2.2, (16), and $w_{t+1}^{(s)} := w_t^{(s)} - \hat{\eta}_t v_t^{(s)}$, we can derive

$$\begin{aligned}
\mathbb{E}\left[\|\nabla f(w_t^{(s)}) - v_t^{(s)}\|^2\right] &\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + \textstyle\sum_{j=1}^t \mathbb{E}\left[\|v_j^{(s)} - v_{j-1}^{(s)}\|^2\right] \\
&\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \\
&+ \ \textstyle\sum_{j=1}^t \mathbb{E}\left[\|\nabla f(w_j^{(s)};\xi_j^{(s)}) - \nabla f(w_{j-1}^{(s)};\xi_j^{(s)})\|^2\right] \\
&\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + L^2\textstyle\sum_{j=1}^t \mathbb{E}\left[\|w_j^{(s)} - w_{j-1}^{(s)}\|^2\right] \\
&\leq \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] + L^2\textstyle\sum_{j=1}^t \hat{\eta}_{j-1}^2\mathbb{E}\left[\|v_{j-1}^{(s)}\|^2\right].
\end{aligned}$$

Substituting this estimate into (60), and summing up the result from $t = 0$ to $t = m$, we eventually get

$$\begin{aligned}
\mathbb{E}\left[f(w_{m+1}^{(s)})\right] \leq \mathbb{E}\left[f(w_0^{(s)})\right] - \sum_{t=0}^m \frac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] + \frac{1}{2}\Big(\sum_{t=0}^m \hat{\eta}_t\Big)\mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \\
+ \frac{L^2}{2}\sum_{t=0}^m \hat{\eta}_t \sum_{j=1}^t \hat{\eta}_{j-1}^2\mathbb{E}\left[\|v_{j-1}^{(s)}\|^2\right] - \sum_{t=0}^m \frac{\hat{\eta}_t(1-L\hat{\eta}_t)}{2}\mathbb{E}\left[\|v_t^{(s)}\|^2\right]. \qquad (61)
\end{aligned}$$

Our next step is to choose $\hat{\eta}_t$ such that

$$L^2\sum_{t=0}^m \hat{\eta}_t \sum_{j=1}^t \hat{\eta}_{j-1}^2\mathbb{E}\left[\|v_{j-1}^{(s)}\|^2\right] - \sum_{t=0}^m \hat{\eta}_t(1-L\hat{\eta}_t)\mathbb{E}\left[\|v_t^{(s)}\|^2\right] \leq 0.$$

This condition can be rewritten explicitly as

$$\begin{aligned}
&\big[L^2\hat{\eta}_0^2(\hat{\eta}_1 + \cdots + \hat{\eta}_m) - \hat{\eta}_0(1-L\hat{\eta}_0)\big]\mathbb{E}\left[\|v_0^{(s)}\|^2\right] \\
&+ \ \big[L^2\hat{\eta}_1^2(\hat{\eta}_2 + \cdots + \hat{\eta}_m) - \hat{\eta}_1(1-L\hat{\eta}_1)\big]\mathbb{E}\left[\|v_1^{(s)}\|^2\right] + \cdots \\
&+ \ \big[L^2\hat{\eta}_{m-1}^2\hat{\eta}_m - \hat{\eta}_{m-1}(1-L\hat{\eta}_{m-1})\big]\mathbb{E}\left[\|v_{m-1}^{(s)}\|^2\right] - \hat{\eta}_m(1-L\hat{\eta}_m)\mathbb{E}\left[\|v_m^{(s)}\|^2\right] \leq 0.
\end{aligned}$$

Similar to (54), to guarantee the last inequality, we impose the following conditions

$$\begin{cases}
-\hat{\eta}_m(1-L\hat{\eta}_m) & \leq 0, \\
L^2\hat{\eta}_{m-1}^2\hat{\eta}_m - \hat{\eta}_{m-1}(1-L\hat{\eta}_{m-1}) & \leq 0 \\
\cdots & \cdots \\
L^2\hat{\eta}_1^2(\hat{\eta}_2 + \cdots + \hat{\eta}_m) - \hat{\eta}_1(1-L\hat{\eta}_1) & \leq 0 \\
L^2\hat{\eta}_0^2(\hat{\eta}_1 + \cdots + \hat{\eta}_m) - \hat{\eta}_0(1-L\hat{\eta}_0) & \leq 0.
\end{cases} \qquad (62)$$

If we tighten all inequations in (62), then we can compute recursively the step-sizes as

$$\hat{\eta}_m = \frac{1}{L}, \quad \text{and} \quad \hat{\eta}_{m-t} := \frac{1}{L\big(1 + L\sum_{j=1}^t \hat{\eta}_{m-j+1}\big)}, \quad \forall t = 1, \cdots, m,$$

which is exactly (35). With this update, we have $\hat{\eta}_0 < \hat{\eta}_1 < \cdots < \hat{\eta}_m$.

Let us define $\Sigma_m := \sum_{t=0}^m \hat{\eta}_t$. Using Σ_m into (62) with equality, we can rewrite it as

$$\begin{cases} L^2\hat{\eta}_m\Sigma_m & = 1 - L\hat{\eta}_m & +L^2(\hat{\eta}_m^2 + \hat{\eta}_m\hat{\eta}_{m-1} + \hat{\eta}_m\hat{\eta}_{m-2} + \cdots + \hat{\eta}_m\hat{\eta}_0) \\ L^2\hat{\eta}_{m-1}\Sigma_m & = 1 - L\hat{\eta}_{m-1} & +L^2(\hat{\eta}_{m-1}^2 + \hat{\eta}_{m-1}\hat{\eta}_{m-2} + \hat{\eta}_{m-1}\hat{\eta}_{m-3} + \cdots + \hat{\eta}_{m-1}\hat{\eta}_0) \\ \cdots & \cdots & \cdots \\ L^2\hat{\eta}_2\Sigma_m & = 1 - L\hat{\eta}_2 & +L^2(\hat{\eta}_2^2 + \hat{\eta}_2\hat{\eta}_1 + \hat{\eta}_2\hat{\eta}_0) \\ L^2\hat{\eta}_1\Sigma_m & = 1 - L\hat{\eta}_1 & +L^2(\hat{\eta}_1^2 + \hat{\eta}_1\hat{\eta}_0) \\ L^2\hat{\eta}_0\Sigma_m & = 1 - L\hat{\eta}_0 & +L^2\hat{\eta}_0^2. \end{cases}$$

If we sum both sides of these equations, and using the definition of Σ_m and $S_m^2 := \sum_{t=0}^m \hat{\eta}_t^2$, we obtain

$$L^2\Sigma_m^2 = (m+1) - L\Sigma_m + \frac{L^2}{2}(\Sigma_m^2 + S_m^2).$$

This expression leads to $L^2\Sigma_m^2 + 2L\Sigma_m - 2(m+1) = L^2S_m^2 \geq 0$. Therefore, we can bound Σ_m as $\Sigma_m \geq \frac{2(m+1)}{L(\sqrt{2m+3}+1)}$ by solving the quadratic inequation $L^2\Sigma_m^2 + 2L\Sigma_m - 2(m+1) \geq 0$ in Σ_m with $\Sigma_m > 0$.

Using the update (35), we can simplify (61) as follows:

$$\mathbb{E}\left[f(w_{m+1}^{(s)})\right] \leq \mathbb{E}\left[f(w_0^{(s)})\right] - \sum_{t=0}^m \frac{\hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] + \frac{\sum_{t=0}^m \hat{\eta}_t}{2}\mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right].$$

Let us define $\hat{\sigma}_s := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right]$ and noting that $f^\star := F^\star \leq \mathbb{E}\left[f(w_{m+1}^{(S)})\right]$ and $\widetilde{w}_0 := w_0^{(0)}$. Summing up the last inequality from $s = 1$ to S and using these relations, we can further derive

$$\sum_{s=1}^S\sum_{t=0}^m \hat{\eta}_t\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] \leq 2\big[f(\widetilde{w}_0) - f^\star\big] + \Big(\sum_{t=0}^m \hat{\eta}_t\Big)\sum_{s=1}^S \hat{\sigma}_s.$$

Using the lower bound of Σ_m as $\Sigma_m \geq \frac{2(m+1)}{L(\sqrt{2m+3}+1)}$, the above inequality leads to

$$\frac{1}{S\Sigma_m}\sum_{s=1}^S\sum_{t=0}^m \hat{\eta}_t\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right] \leq \frac{(\sqrt{2m+3}+1)L}{S(m+1)}\big[f(\widetilde{w}_0) - f^\star\big] + \frac{1}{S}\sum_{s=1}^S \hat{\sigma}_s. \qquad (63)$$

Since $\mathbf{Prob}\left(\widetilde{w}_T = w_t^{(s)}\right) = p_{(s-1)m+t}$ with $p_{(s-1)m+t} = \frac{\hat{\eta}_t}{S\Sigma_m}$ for $s = 1, \cdots, S$ and $t = 0, \cdots, m$, we have

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] = \frac{1}{S\Sigma_m}\sum_{s=1}^S\sum_{t=0}^m \hat{\eta}_t\mathbb{E}\left[\|\nabla f(w_t^{(s)})\|^2\right].$$

Substituting this estimate into (63), we obtain (36).

Now, we consider two cases:

Case (a): If we apply this algorithm variant to solve the non-composite finite-sum problem of (2) (i.e. $\psi = 0$) using the full-gradient snapshot for the outer-loop with $b_s = n$, then $v_0^{(s)} = \nabla f(w_0^{(s)})$, which leads to $\hat{\sigma}_s = 0$. Using this fact and the choice of epoch length $m = n$ into (36), we obtain

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{L(1+\sqrt{4n+1})}{S(n+1)}\left[f(\widetilde{w}_0) - f^{\star}\right],$$

which is exactly (37).

To achieve $\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \varepsilon^2$, we impose $\frac{L(1+\sqrt{4n+1})}{S(n+1)}[f(\widetilde{w}_0) - f^{\star}] = \varepsilon^2$. Hence, the maximum number of iterations is at most $T = 3nS \sim S(n+1) = \frac{L(1+\sqrt{4n+1})}{\varepsilon^2}[f(\widetilde{w}_0) - f^{\star}]$. This shows that $T := \mathcal{O}\left(\frac{\sqrt{n}L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^2}\right)$. The number of gradient evaluations ∇f_i is at most $T_{\text{grad}} := S(n + 2(m+1)) = 3S(n+1) = 3T$. Hence, $T_{\text{grad}} = \mathcal{O}\left(\frac{\sqrt{n}L[f(\widetilde{w}_0)-f^{\star}]}{\varepsilon^2}\right)$.

Case (b): If we apply this algorithm variant to solve the non-composite expectation problem of (1) (i.e. $\psi = 0$) using mini-batch $b_s = b := \mathcal{O}\left(\frac{\sigma^2}{\varepsilon^2}\right)$ for the outer-loop and $m := \mathcal{O}\left(\frac{\sigma^{\nu}}{\varepsilon^2}\right)$ for some $\nu \geq 0$, then by (20), we have $\hat{\sigma}_s := \mathbb{E}\left[\|\nabla f(w_0^{(s)}) - v_0^{(s)}\|^2\right] \leq \frac{\sigma^2}{b_s} = \frac{\sigma^2}{b}$. Using this bound into (36), we get

$$\mathbb{E}\left[\|\nabla f(\widetilde{w}_T)\|^2\right] \leq \frac{L(1+\sqrt{4m+1})}{S(m+1)}\left[f(\widetilde{w}_0) - f^{\star}\right] + \frac{\sigma^2}{b}.$$

This is exactly (38). The last conclusion is proved similarly as in Theorem 8. □

References

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. IEEE Transactions on Information Theory, 99:1–1, 2010.

Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC), pages 1200–1205, June 2017a. Montreal, Canada.

Z. Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. arXiv preprint arXiv:1708.08694, 2017b.

Z. Allen-Zhu and Y. Li. NEON2: Finding local minima via first-order oracles. In Advances in Neural Information Processing Systems, pages 3720–3730, 2018.

Zeyuan Allen-Zhu and Yang Yuan. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In ICML, pages 1080–1089, 2016.

H. H. Bauschke and P. Combettes. Convex analysis and monotone operators theory in Hilbert spaces. Springer-Verlag, 2nd edition, 2017.

L. Bottou. Online learning and stochastic approximations. In David Saad, editor, Online Learning in Neural Networks, pages 9–42. Cambridge University Press, New York, NY, USA, 1998. ISBN 0-521-65263-4.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010, pages 177–186. Springer, 2010.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization Methods for Large-Scale Machine Learning. SIAM Rev., 60(2):223–311, 2018.

A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C.-B. Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. SIAM J. Optim., 28(4):2783–2808, 2018.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In NIPS, pages 1646–1654, 2014.

C. Fang, C. J. Li, Z. Lin, and T. Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path integrated differential estimator. arXiv preprint arXiv:1807.01695, 2018.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization: A generic algorithmic framework. SIAM J. Optim., 22(4):1469–1492, 2012.

S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. SIAM J. Optim., 23(4):2341–2368, 2013.

S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. Math. Program., 155(1-2):267–305, 2016.

I. Goodfellow, Y. Bengio, and A. Courville. Deep learning, volume 1. MIT press Cambridge, 2016.

R. Harikandeh, M. O. Ahmed, A. Virani, M. Schmidt, J. Konečný, and S. Sallinen. Stopwasting my gradients: Practical SVRG. In Advances in Neural Information Processing Systems (NIPS), pages 2251–2259, 2015.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in Neural Information Processing Systems (NIPS), pages 315–323, 2013.

H. Karimi, J. Nutini, and M. 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.

Z. Li and J. Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. arXiv preprint arXiv:1802.04477, 2018.

L. Lihua, C. Ju, J. Chen, and M. Jordan. Non-convex finite-sum optimization via SCSG methods. In Advances in Neural Information Processing Systems, pages 2348–2358, 2017.

S. L. Lohr. Sampling: Design and Analysis. Nelson Education, 2009.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. SIAM J. Optim., 19(4):1574–1609, 2009.

A. Nemirovskii and D. Yudin. Problem Complexity and Method Efficiency in Optimization. Wiley Interscience, 1983.

Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87 of Applied Optimization. Kluwer Academic Publishers, 2004.

Y. Nesterov and B.T. Polyak. Cubic regularization of Newton method and its global performance. Math. Program., 108(1):177–205, 2006.

L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In ICML, 2017a.

L. M. Nguyen, N. H. Nguyen, D. T. Phan, J. R. Kalagnanam, and K. Scheinberg. When does stochastic gradient algorithm work well? arXiv:1801.06159, 2018a.

L. M. Nguyen, K. Scheinberg, and M. Takac. Inexact SARAH Algorithm for Stochastic Optimization. arXiv preprint arXiv:1811.10105, 2018b.

L. M. Nguyen, M. van Dijk, D. T. Phan, P. H. Nguyen, T.-W. Weng, and J. R. Kalagnanam. Optimal finite-sum smooth non-convex optimization with SARAH. arXiv preprint arXiv:1901.07648, 2019.

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takác. Stochastic recursive gradient algorithm for nonconvex optimization. CoRR, abs/1705.07261, 2017b.

A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In Advances in Neural Information Processing Systems, pages 1574–1582, 2014.

S. Reddi, S. Sra, B. Póczos, and A. Smola. Stochastic Frank-Wolfe methods for nonconvex optimization. arXiv preprint arXiv:1607.08254, 2016a.

S. J. Reddi, S. Sra, B. Póczos, and A. J. Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In Advances in Neural Information Processing Systems, pages 1145–1153, 2016b.

H. Robbins and S. Monro. A stochastic approximation method. The annals of mathematical statistics, pages 400–407, 1951.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. Math. Program., 162(1-2):83–112, 2017.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. J. Mach. Learn. Res., 14:567–599, 2013.

A. Shapiro, D. Dentcheva, and A. Ruszczynski. Lectures on Stochastic Programming: Modelling and Theory. SIAM, 2009.

S. Sra, S. Nowozin, and S. J. Wright. Optimization for Machine Learning. Mit Press, 2012.

Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh. SpiderBoost: A class of faster variance-reduced algorithms for nonconvex optimization. arXiv preprint arXiv:1810.10690, 2018.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. SIAM J. Optim., 24(4):2057–2075, 2014.

L. Zhao, M. Mammadov, and J. Yearwood. From convex to nonconvex: a loss function analysis for binary classification. In IEEE International Conference on Data Mining Workshops (ICDMW), pages 1281–1288. IEEE, 2010.

D. Zhou and Q. Gu. Lower bounds for smooth nonconvex finite-sum optimization. arXiv preprint arXiv:1901.11224, 2019.

D. Zhou, P. Xu, and Q. Gu. Stochastic nested variance reduction for nonconvex optimization. arXiv preprint arXiv:1806.07811, 2018.

Y. Zhou, Z. Wang, K. Ji, Y. Liang, and V. Tarokh. Momentum schemes with stochastic variance reduction for nonconvex composite optimization. arXiv preprint arXiv:1902.02715, 2019.