A Stochastic Composite Gradient Method with Incremental Variance Reduction

Junyu Zhang
University of Minnesota
Minneapolis, Minnesota 55455
zhan4393@umn.edu

Lin Xiao
Microsoft Research
Redmond, Washington 98052
lin.xiao@microsoft.com

Abstract

We consider the problem of minimizing the composition of a smooth (nonconvex) function and a smooth vector mapping, where the inner mapping is in the form of an expectation over some random variable or a finite sum. We propose a stochastic composite gradient method that employs an incremental variance-reduced estimator for both the inner vector mapping and its Jacobian. We show that this method achieves the same orders of complexity as the best known first-order methods for minimizing expected-value and finite-sum nonconvex functions, despite the additional outer composition which renders the composite gradient estimator biased. This finding enables a much broader range of applications in machine learning to benefit from the low complexity of incremental variance-reduction methods.

1 Introduction

In this paper, we consider stochastic composite optimization problems

$$\underset{x\in\mathbf{R}^d}{\text{minimize}} \quad f\big(\mathbf{E}_\xi[g_\xi(x)]\big) + r(x), \tag{1}$$

where $f : \mathbf{R}^p \to \mathbf{R}$ is a smooth and possibly nonconvex function, ξ is a random variable, $g_\xi : \mathbf{R}^d \to \mathbf{R}^p$ is a smooth vector mapping for $a.e.\ \xi$, and r is convex and lower-semicontinuous. A special case we will consider separately is when ξ is a discrete random variable with uniform distribution over $\{1, 2, \ldots, n\}$. In this case the problem is equivalent to a deterministic optimization problem

$$\underset{x\in\mathbf{R}^d}{\text{minimize}} \quad f\left(\frac{1}{n}\sum_{i=1}^n g_i(x)\right) + r(x)\,. \tag{2}$$

The formulations in (1) and (2) cover a broader range of applications than classical stochastic optimization and empirical risk minimization (ERM) problems where each g_ξ is a scalar function ($p = 1$) and f is the scalar identity map. Interesting examples include the policy evaluation in reinforcement learning (RL) [e.g., 30], the risk-averse mean-variance optimization ([e.g., 28, 29], through a reformulation by [35]), the stochastic variational inequality ([e.g., 12, 15] through a reformulation in [10]), the 2-level composite risk minimization problems [7], etc.

For the ease of notation, we define

$$g(x) := \mathbf{E}_\xi[g_\xi(x)], \qquad F(x) := f(g(x)), \qquad \Phi(x) := F(x) + r(x). \tag{3}$$

In addition, let f' and F' denote the gradients of f and F respectively, and $g'_\xi(x) \in \mathbf{R}^{p\times d}$ denote the Jacobian matrix of g_ξ at x. Then we have

$$F'(x) = \nabla\Big(f\big(\mathbf{E}_\xi[g_\xi(x)]\big)\Big) = \Big(\mathbf{E}_\xi[g'_\xi(x)]\Big)^T f'\big(\mathbf{E}_\xi[g_\xi(x)]\big)\,.$$

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Table 1: Sample complexities of CIVR (Composite Incremental Variance Reduction)

Problem	Assumptions (common: f and g_ξ Lipschitz and smooth, thus F smooth)		
	F nonconvex r convex	F ν-gradient dominant $r \equiv 0$	F convex, r convex Φ μ-optimally strongly convex
(1)	$O(\epsilon^{-3/2})$	$O\left((\nu\epsilon^{-1})\log\epsilon^{-1}\right)$	$O\left((\mu^{-1}\epsilon^{-1})\log\epsilon^{-1}\right)$
(2)	$O(\min\{\epsilon^{-3/2}, n^{1/2}\epsilon^{-1}\})$	$O\left((n+\nu n^{1/2})\log\epsilon^{-1}\right)$	$O\left((n+\mu^{-1}n^{1/2})\log\epsilon^{-1}\right)$

In practice, computing $F'(x)$ exactly can be very costly if not impossible. Due to the nonlinearity of the outer composition, simply multiplying the unbiased estimators $\mathbf{E}[\tilde{g}(x)] = g(x)$ and $\mathbf{E}[\tilde{g'}(x)] = g'(x)$ results in a biased estimator for $F'(x)$, namely, $\mathbf{E}[[\tilde{g'}(x)]^T f'(\tilde{g}(x))] \neq F'(x)$, see [e.g., 35]. This is in great contrast to the classical stochastic optimization problem

$$\underset{x\in\mathbf{R}^d}{\text{minimize}} \quad \mathbf{E}_\xi\left[g_\xi(x)\right] + r(x), \tag{4}$$

where one can always get an unbiased gradient estimator for the smooth part. This fact makes such composition structure be of independent interest for research on stochastic and randomized algorithms.

In this paper, we develop an efficient stochastic composite gradient method called CIVR (Composite Incremental Variance Reduction), for solving problems of the forms (1) and (2). We measure efficiency by the sample complexity of the individual functions g_ξ and their Jacobian g'_ξ, i.e., the total number of times they need to be evaluated at some point, in order to find an ϵ-approximate solution. For nonconvex functions, an ϵ-approximate solution is some random output of the algorithm $\bar{x} \in \mathbf{R}^d$ that satisfies $\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] \leq \epsilon$, where $\mathcal{G}(\bar{x})$ is the *proximal gradient mapping* of the objective function Φ at $\bar{x}$ (see details in Section 2). If $r \equiv 0$, then $\mathcal{G}(\bar{x}) = F'(\bar{x})$ and the criteria for ϵ-approximation becomes $\mathbf{E}[\|F'(\bar{x})\|^2] \leq \epsilon$. If the objective Φ is convex, we require $\mathbf{E}[\Phi(\bar{x}) - \Phi^\star] \leq \epsilon$ where $\Phi^\star = \inf_x \Phi(x)$. For smooth *and* convex functions, these two notions are compatible, meaning that the dependence of the sample complexity on ϵ in terms of both notions are of the same order.

Table 1 summarizes the sample complexities of the CIVR method under different assumptions obtained in this paper. We can define a condition number $\kappa = O(\nu)$ for ν-gradient dominant functions and $\kappa = O(1/\mu)$ for μ-optimally strongly convex functions, then the complexities become $O\left((\kappa\epsilon^{-1})\log\epsilon^{-1}\right)$ and $O\left((n+\kappa n^{1/2})\log\epsilon^{-1}\right)$ for (1) and (2) respectively. In order to better position our contributions, we next discuss related work and then putting these results into context.

1.1 Related Work

We first discuss the nonconvex stochastic optimization problem (4), which is a special cases of (1). When $r \equiv 0$ and $g(x) = \mathbf{E}_\xi[g_\xi(x)]$ is smooth, Ghadimi and Lan [9] developed a randomized stochastic gradient method with iteration complexity $O(\epsilon^{-2})$. Allen-Zhu [2] obtained $O(\epsilon^{-1.625})$ with additional second-order guarantee. There are also many recent works on solving its finite-sum version

$$\underset{x\in\mathbf{R}^d}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^n g_i(x) + r(x), \tag{5}$$

which is also a special case of (2). By extending the variance reduction techniques SVRG [13, 34] and SAGA [6] to nonconvex optimization, Allen-Zhu and Hazan [3] and Reddi et al. [24, 25, 26] developed randomized algorithms with sample complexity $O(n + n^{2/3}\epsilon^{-1})$. Under additional assumptions of gradient dominance or strong convexity, they obtained sample complexity $O((n+\kappa n^{2/3})\log\epsilon^{-1})$, where κ is a suitable condition number. Allen-Zhu [1] and Lei et al. [17] obtained $O(\min\{\epsilon^{-5/3}, n^{2/3}\epsilon^{-1}\})$.

Based on a new variance reduction technique called SARAH [21], Nguyen et al. [22] and Pham et al. [23] developed nonconvex extensions to obtain sample complexities $O(\epsilon^{-3/2})$ and $O(n + n^{1/2}\epsilon^{-1})$ for solving the expectation and finite-sum cases respectively. Fang et al. [8] introduced another variance reduction technique called Spider, which can be viewed as a more general variant of SARAH. They obtained sample complexities $O(\epsilon^{-3/2})$ and $O(\min\{\epsilon^{-3/2}, n^{1/2}\epsilon^{-1}\})$ for the two cases respectively, but require small step sizes that are proportional to ϵ. Wang et al. [33] extended Spider to obtain the same complexities with constant step sizes and $O((n+\kappa^2)\log\epsilon^{-1})$ under the gradient-dominant condition. In addition, Zhou et al. [36] obtained similar results using a nested SVRG approach.

In addition to the above works on solving special cases of (1) and (2), there are also considerable recent works on a more general, two-layer stochastic composite optimization problem

$$\underset{x\in\mathbf{R}^d}{\text{minimize}} \quad \mathbf{E}_\nu\left[f_\nu\left(\mathbf{E}_\xi[g_\xi(x)]\right)\right] + r(x), \tag{6}$$

where f_ν is parametrized by another random variables ν, which is independent of ξ. For the case $r \equiv 0$, Wang et al. [31] derived algorithms to find an ϵ-approximate solution with sample complexities $O(\epsilon^{-4})$, $O(\epsilon^{-3.5})$ and $O(\epsilon^{-1.25})$ for the smooth nonconvex case, smooth convex case and smooth strongly convex case respectively. For nontrivial convex r, Wang et al. [32] obtained improved sample complexity of $O(\epsilon^{-2.25})$, $O(\epsilon^{-2})$ and $O(\epsilon^{-1})$ for the three cases mentioned above respectively.

As a special case of (6), the following finite-sum problem also received significant attention:

$$\underset{x\in\mathbf{R}^d}{\text{minimize}} \quad \frac{1}{m}\sum_{j=1}^{m} f_j\left(\frac{1}{n}\sum_{i=1}^{n} g_i(x)\right) + r(x)\,. \tag{7}$$

When $r \equiv 0$ and the overall objective function is strongly convex, Lian et al. [19] derived two algorithms based on the SVRG scheme to attain sample complexities $O((m+n+\kappa^3)\log\epsilon^{-1}))$ and $O((m+n+\kappa^4)\log\epsilon^{-1}))$ respectively, where κ is some suitably defined condition number. Huo et al. [11] also used the SVRG scheme to obtain an $O(m+n+(m+n)^{2/3}\epsilon^{-1})$ complexity for the smooth nonconvex case and $O((m+n+\kappa^3)\log\epsilon^{-1}))$ for strongly convex problems with nonsmooth r. More recently, Zhang and Xiao [35] proposed a composite randomized incremental gradient method based on the SAGA estimator [6], which matches the best known $O(m+n+(m+n)^{2/3}\epsilon^{-1})$ complexity when F is smooth and nonconvex, and obtained an improved complexity $O\big((m+n+\kappa(m+n)^{2/3})\log\epsilon^{-1}\big)$ under either gradient dominant or strongly convex assumptions. When applied to the special cases (1) and (2) we focus on in this paper ($m = 1$), these results are strictly worse than ours in Table 1.

1.2 Contributions and Outline

We develop the CIVR method by extending the variance reduction technique of SARAH [21–23] and SPIDER [8, 33] to solve the composite optimization problems (1) and (2). The complexities of CIVR in Table 1 match the best results for solving the non-composite problems (4) and (5), despite the additional outer composition and the composite-gradient estimator always being biased. In addition:

- By setting f and g_ξ's to be the identity mapping and scalar mappings respectively, problem (2) includes problem (5) as a special case. Therefore, the lower bounds in [8] for the non-composite finite-sum optimization problem (5) indicates that our $O\big(\min\{\epsilon^{-3/2}, n^{1/2}\epsilon^{-1}\}\big)$ complexity for solving the more general composite finite-sum problem (2) is near-optimal.
- Under the assumptions of gradient dominance or strong convexity, the $O\left(\left(n+\kappa n^{1/2}\right)\log\epsilon^{-1}\right)$ complexity only appeared for the special case (5) in the recent work [18].

Our results indicate that the additional smooth composition in (1) and (2) does not incur higher complexity compared with (4) and (5), despite the difficulty of dealing with biased estimators. We believe these results can also be extended to the two-layer problems (6) and (7), by replacing n with $m+n$ in Table 1. But the extensions require quite different techniques and we will address them in a separate paper.

The rest of this paper is organized as follows. In Section 2, we introduce the CIVR method. In Section 3, we present convergence results of CIVR for solving the composite optimization problems (1) and (2) and the required parameter settings. Better complexities of CIVR under the gradient-dominant and optimally strongly convex conditions are given in Section 4. In Section 5, we present numerical experiments for solving a risk-averse portfolio optimization problem (5) on real-world datasets.

2 The composite incremental variance reduction (CIVR) method

With the notations in (3), we can write the composite stochastic optimization problem (1) as

$$\underset{x\in\mathbf{R}^d}{\text{minimize}} \quad \big\{\Phi(x) = F(x) + r(x)\big\}, \tag{11}$$

where F is smooth and r is convex. The proximal operator of r with parameter η is defined as

$$\mathbf{prox}_r^\eta(x) := \underset{y}{\text{argmin}}\left\{r(y) + \frac{1}{2\eta}\|y-x\|^2\right\}. \tag{12}$$

Algorithm 1: Composite Incremental Variance Reduction (CIVR)

input: initial point x_0^1, step size $\eta > 0$, number of epochs $T \geq 1$, and a set of triples $\{\tau_t, B_t, S_t\}$ for $t = 1, \ldots, T$, where τ_t is the epoch length and B_t and S_t are sample sizes in epoch t.

for $t = 1, ..., T$ **do**

Sample a set $\mathcal{B}_t$ with size B_t from the distribution of ξ, and construct the estimates

$$y_0^t = \frac{1}{B_t} \sum_{\xi \in \mathcal{B}_t} g_\xi(x_0^t), \qquad z_0^t = \frac{1}{B_t} \sum_{\xi \in \mathcal{B}_t} g'_\xi(x_0^t), \tag{8}$$

Compute $\tilde{\nabla} F(x_0^t) = (z_0^t)^T f'(y_0^t)$ and update: $x_1^t = \mathbf{prox}_r^\eta \left(x_0^t - \eta \tilde{\nabla} F(x_0^t)\right)$.

for $i = 1, ..., \tau_t - 1$ **do**

Sample a set $\mathcal{S}_i^t$ with size S_t from the distribution of ξ, and construct the estimates

$$y_i^t = y_{i-1}^t + \frac{1}{S_t} \sum_{\xi \in \mathcal{S}_i^t} \left(g_\xi(x_i^t) - g_\xi(x_{i-1}^t)\right), \tag{9}$$

$$z_i^t = z_{i-1}^t + \frac{1}{S_t} \sum_{\xi \in \mathcal{S}_i^t} \left(g'_\xi(x_i^t) - g'_\xi(x_{i-1}^t)\right). \tag{10}$$

Compute $\tilde{\nabla} F(x_i^t) = (z_i^t)^T f'(y_i^t)$ and update: $x_{i+1}^t = \mathbf{prox}_r^\eta \left(x_i^t - \eta \tilde{\nabla} F(x_i^t)\right)$.

end

Set $x_0^{t+1} = x_{\tau_t}^t$.

end

output: $\bar{x}$ randomly chosen from $\left\{x_i^t\right\}_{i=0,\ldots,\tau_t-1}^{t=1,\ldots,T}$.

We assume that r is relatively simple, meaning that its proximal operator has a closed-form solution or can be computed efficiently. The proximal gradient method [e.g., 20, 4] for solving problem (11) is

$$x^{t+1} = \mathbf{prox}_r^\eta \left(x^t - \eta F'(x^t)\right), \tag{13}$$

where η is the step size. The *proximal gradient mapping* of Φ is defined as

$$\mathcal{G}_\eta(x) \triangleq \frac{1}{\eta}\Big(x - \mathbf{prox}_r^\eta \left(x - \eta F'(x)\right)\Big). \tag{14}$$

As a result, the proximal gradient method (13) can be written as $x^{t+1} = x^t - \eta\, \mathcal{G}(x^t)$. Notice that when $r \equiv 0$, $\mathbf{prox}_r^\eta(\cdot)$ becomes the identity mapping and we have $\mathcal{G}_\eta(x) \equiv F'(x)$ for any $\eta > 0$.

Suppose $\bar{x}$ is generated by a randomized algorithm. We call $\bar{x}$ an ϵ-stationary point in expectation if

$$\mathbf{E}\left[\|\mathcal{G}_\eta(\bar{x})\|^2\right] \leq \epsilon. \tag{15}$$

(We assume that η is a constant that does not depend on ϵ.) As we mentioned in the introduction, we measure the efficiency of an algorithm by its sample complexity of g_ξ and their Jacobian g'_ξ, i.e., the total number of times they need to be evaluated, in order to find a point $\bar{x}$ that satisfies (15). Our goal is to develop a randomized algorithm that has low sample complexity.

We present in Algorithm 1 the Composite Incremental Variance Reduction (CIVR) method. This methods employs a two time-scale variance-reduced estimator for both the inner function value of $g(\cdot) = \mathbf{E}_\xi[g_\xi(\cdot)]$ and its Jacobian $g'(\cdot)$. At the beginning of each outer iteration t (each called an epoch), we construct a relatively accurate estimate y_0^t for $g(x_0^t)$ and z_0^t for $g'(x_0^t)$ respectively, using a relatively large sample size B_t. During each inner iteration i of the tth epoch, we construct an estimate y_i^t for $g(x_i^t)$ and z_i^t for $g'(x_i^t)$ respectively, using a smaller sample size S_t *and* incremental corrections from the previous iterations. Note that the epoch length τ_t and the sample sizes B_t and S_t are all adjustable for each epoch t. Therefore, besides setting a constant set of parameters, we can also adjust them gradually in order to obtain better theoretical properties and practical performance.

This variance-reduction technique was first proposed as part of SARAH [21] where it is called *recursive* variance reduction. It was also proposed in [8] in the form of a *Stochastic Path-Integrated Differential EstimatoR* (SPIDER). Here we simply call it *incremental* variance reduction. A distinct

feature of this incremental estimator is that the inner-loop estimates y_i^t and z_i^t are biased, i.e.,

$$\begin{cases} \mathbf{E}[y_i^t | x_i^t] = g(x_i^t) - g(x_{i-1}^t) + y_{i-1}^t \neq g(x_i^t), \\ \mathbf{E}[z_i^t | x_i^t] = g'(x_i^t) - g'(x_{i-1}^t) + z_{i-1}^t \neq g'(x_i^t). \end{cases} \tag{16}$$

This is in contrast to two other popular variance-reduction techniques, namely, SVRG [13] and SAGA [6], whose gradient estimators are always unbiased. Note that unbiased estimators for $g(x_i^t)$ and $g'(x_i^t)$ are not essential here, because the composite estimator $\tilde{\nabla} F(x_i^t) = (z_i^t)^T f'(y_i^t)$ is always biased. Therefore the our main task is to control the variance and bias altogether for the proposed estimator.

3 Convergence Analysis

In this section, we present theoretical results on the convergence properties of CIVR (Algorithm 1) when the composite function F is smooth. More specifically, we make the following assumptions.

Assumption 1. *The following conditions hold concerning problems* (1) *and* (2)*:*

- $f : \mathbf{R}^p \to \mathbf{R}$ *is a* C^1 *smooth and* ℓ_f*-Lipschitz function and its gradient* f' *is* L_f*-Lipschitz.*
- *Each* $g_\xi : \mathbf{R}^d \to \mathbf{R}^p$ *is a* C^1 *smooth and* ℓ_g*-Lipschitz vector mapping and its Jacobian* g'_ξ *is* L_g*-Lipschtiz. Consequently,* g *in* (3) *is* ℓ_g*-Lipschitz and its Jacobian* g' *is* L_g*-Lipschitz.*
- $r : \mathbf{R}^d \to \mathbf{R} \cup \{\infty\}$ *is a convex and lower-semicontinuous function.*
- *The overall objective function* Φ *is bounded below, i.e.,* $\Phi^* = \inf_x \Phi(x) > -\infty$.

Assumption 2. *For problem* (1)*, we further assume that there exist constants* σ_g *and* $\sigma_{g'}$ *such that*

$$\mathbf{E}_\xi[\|g_\xi(x) - g(x)\|^2] \leq \sigma_g^2, \qquad \mathbf{E}_\xi[\|g'_\xi(x) - g'(x)\|^2] \leq \sigma_{g'}^2. \tag{17}$$

As a result of Assumption 1, $F(x) = f\big(g(x)\big)$ is smooth and F' is L_F-Lipschitz continuous with

$$L_F = \ell_g^2 L_f + \ell_f L_g$$

(see proof in the supplementary materials). For convenience, we also define two constants

$$G_0 := 2\big(\ell_g^4 L_f^2 + \ell_f^2 L_g^2\big), \qquad \text{and} \qquad \sigma_0^2 := 2\big(\ell_g^2 L_f^2 \sigma_g^2 + \ell_f^2 \sigma_{g'}^2\big). \tag{18}$$

It is important to notice that $G_0 = O(L_F^2)$, hence the step size used later is $\eta = \Theta(1/\sqrt{G_0}) = \Theta(1/L_F)$. We are allowed to use this constant step size mainly due to the assumption that each $g_\xi(\cdot)$ is smooth, instead of the weaker assumption that $E_\xi[g_\xi(x)]$ is smooth as in classical stochastic optimization.

In the next two subsections, we present complexity analysis of CIVR for solving problem (1) and (2) respectively. Due to the space limitation, all proofs are provided in the supplementary materials.

3.1 The composite expectation case

The following results for solving problem (1) are presented with notations defined in (3), (14) and (18).

Theorem 1. *Suppose Assumptions 1 and 2 hold. Given any* $\epsilon > 0$*, we set* $T = \lceil 1/\sqrt{\epsilon} \rceil$ *and*

$$\tau_t = \tau = \lceil 1/\sqrt{\epsilon} \rceil, \quad B_t = B = \lceil \sigma_0^2/\epsilon \rceil, \quad S_t = S = \lceil 1/\sqrt{\epsilon} \rceil, \quad \textit{for} \quad t = 1, \ldots, T.$$

Then as long as $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12 G_0}}$*, the output* $\bar{x}$ *of Algorithm 1 satisfies*

$$\mathbf{E}\big[\|\mathcal{G}_\eta(\bar{x})\|^2\big] \leq \Big(8\big(\Phi(x_0^1) - \Phi^*\big)\eta^{-1} + 6\Big) \cdot \epsilon = O(\epsilon). \tag{19}$$

As a result, the sample complexity of obtaining an ϵ*-approximate solution is* $TB + 2T\tau S = O\big(\epsilon^{-3/2}\big)$.

Note that in the above scheme, the epoch lengths τ_t and all the batch sizes B_t and S_t are set to be constant (depending on a pre-fixed ϵ) without regard of t. Intuitively, we do not need as many samples in the early stage of the algorithm as in the later stage. In addition, it will be useful in practice to have a variant of the algorithm that can adaptively choose τ_t, B_t and S_t throughout the epochs without dependence on a pre-fixed precision. This is done in the following theorem.

Theorem 2. *Suppose Assumptions 1 and 2 hold. We set* $\tau_t = S_t = \lceil at + b \rceil$ *and* $B_t = \lceil \sigma_0^2 (at+b)^2 \rceil$ *where* $a > 0$ *and* $b \geq 0$. *Then as long as* $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12 G_0}}$, *we have for any* $T \geq 1$,

$$\mathbf{E}\big[\|\mathcal{G}_\eta(\bar{x})\|^2\big] \leq \frac{2}{aT^2 + (a+2b)T} \left(\frac{8\big(\Phi(x_0^1) - \Phi^*\big)}{\eta} + \frac{6}{a+b} + \frac{6}{a} \ln\left(\frac{aT+b}{a+b} \right) \right) = O\Big(\frac{\ln T}{T^2} \Big). \quad (20)$$

As a result, obtaining an ϵ*-approximate solution requires* $T = \tilde{O}(1/\sqrt{\epsilon})$ *epochs and a total sample complexity of* $\tilde{O}\big(\epsilon^{-3/2}\big)$, *where the* $\tilde{O}(\cdot)$ *notation hides logarithmic factors.*

3.2 The composite finite-sum case

In this section, we consider the composite finite-sum optimization problem (2). In this case, the random variable ξ has a uniform distribution over the finite index set $\{1, ..., n\}$. At the beginning of each epoch in Algorithm 1, we use the full sample size $\mathcal{B}_t = \{1, \ldots, n\}$ to compute y_0^t and z_0^t. Therefore $B_t = n$ for all t and Equation (8) in Algorithm 1 becomes

$$y_0^t = g(x_0^t) = \frac{1}{n} \sum_{j=1}^n g_j(x_0^t), \qquad z_0^t = g'(x_0^t) = \frac{1}{n} \sum_{j=1}^n g_j'(x_0^t)\,. \quad (21)$$

Also in this case, we no longer need Assumption 2.

Theorem 3. *Suppose Assumptions 1 holds. Let the parameters in Algorithm 1 be set as* $\mathcal{B}_t = \{1, \ldots, n\}$ *and* $\tau_t = S_t = \lceil \sqrt{n} \rceil$ *for all* t. *Then as long as* $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12 G_0}}$, *we have for any* $T \geq 1$,

$$\mathbf{E}\big[\|\mathcal{G}_\eta(\bar{x})\|^2\big] \leq \frac{8\big(\Phi(x_0^1) - \Phi^*\big)}{\eta \sqrt{n} T} = O\left(\frac{1}{\sqrt{n} T} \right), \quad (22)$$

As a result, obtaining an ϵ*-approximate solution requires* $T = O\big(1/(\sqrt{n}\epsilon)\big)$ *epochs and a total sample complexity of* $TB + 2T\tau S = O\big(n + \sqrt{n}\epsilon^{-1}\big)$.

Similar to the previous section, we can also choose the epoch lengths and sample sizes adaptively to save the sampling cost in the early stage of the algorithm. However, due to the finite-sum structure of the problem, when the batch size B_t reaches n, we will start to take the full batch at the beginning of each epoch to get the exact $g(x_0^t)$ and $g'(x_0^t)$. This leads to the following theorem.

Theorem 4. *Suppose Assumptions 1 holds. For some positive constants* $a > 0$ *and* $0 \leq b < \sqrt{n}$, *denote* $T_0 := \lceil \frac{\sqrt{n} - b}{a} \rceil = O\big(\sqrt{n}\big)$. *When* $t \leq T_0$ *we set the parameters to be* $\tau_t = S_t = \sqrt{B_t} = \lceil at + b \rceil$; *when* $t > T_0$, *we set* $\mathcal{B}_t = \{1, \ldots, n\}$ *and* $\tau_t = S_t = \lceil \sqrt{n} \rceil$. *Then as long as* $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12 G_0}}$,

$$\mathbf{E}\big[\|\mathcal{G}_\eta(\bar{x})\|^2\big] \leq \begin{cases} O\big(\frac{\ln T}{T^2}\big) & \text{if } T \leq T_0\,, \\ O\big(\frac{\ln n}{\sqrt{n}(T - T_0 + 1)}\big) & \text{if } T > T_0\,. \end{cases} \quad (23)$$

As a result, the total sample complexity of Algorithm 1 for obtaining an ϵ*-approximate solution is* $\tilde{O}\big(\min\big\{\sqrt{n}\epsilon^{-1}, \epsilon^{-3/2}\big\}\big)$, *where* $\tilde{O}(\cdot)$ *hides logarithmic factors.*

4 Fast convergence rates under stronger conditions

In this section we consider two cases where fast linear convergence can be guaranteed for CIVR.

4.1 Gradient-dominant function

The first case is when $r \equiv 0$ and F is ν*-gradient dominant*, i.e., there is some $\nu > 0$ such that

$$F(x) - \inf_y F(y) \leq \frac{\nu}{2} \|F'(x)\|^2, \qquad \forall\, x \in \mathbf{R}^d. \quad (24)$$

Note that a μ-strongly convex function is $(1/\mu)$-gradient dominant by this definition. Hence strong convexity is a special case of the gradient dominant condition, which in turn is a special case of the Polyak-Łojasiewicz condition with the Łojasiewicz exponent equal to 2 [see, e.g., 14].

In order to solve (1) with a pre-fixed precision ϵ, we use a periodic restart strategy as depicted in Algorithm 2. For this restarted version of CIVR, we have the following results.

Algorithm 2: Restarted CIVR

input: initial point $\bar{x}^0$, step size $\eta > 0$, number of restarts K, number of epochs $T \geq 1$, and a set of triples $\{\tau_t, B_t, S_t\}$ for $t = 1, \ldots, T$.
for $k = 0, ..., K-1$ **do**
  Generate $\bar{x}^{k+1}$ by Algorithm 1, with parameters $T, \eta, \{\tau_t, B_t, S_t\}$ and initial point $\bar{x}^k$.
end
output: $\bar{x}^K$.

Theorem 5. *Consider* (1) *with* $r \equiv 0$. *Suppose Assumptions 1 and 2 hold and* F *is* ν*-gradient dominant. For Algorithm 2, given any* $\epsilon > 0$, *let* $\tau_t = S_t = \lceil \frac{1}{\sqrt{\epsilon}} \rceil$, $B_t = \lceil \frac{12\nu\sigma_0^2}{\epsilon} \rceil$ *and* $T = \lceil \frac{16\nu\sqrt{\epsilon}}{\eta} \rceil$. *Then as long as* $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12G_0}}$,

$$\mathbf{E}\big[F(\bar{x}^{k+1}) - F^*\big] \leq \frac{1}{2}\big(F(\bar{x}^k) - F^*\big) + \frac{1}{2}\epsilon. \tag{25}$$

Consequently, $\mathbf{E}[F(\bar{x}^k) - F^*]$ *converges linearly to* ϵ *with a factor of* $\frac{1}{2}$ *per period. The sample complexity for finding an* ϵ*-solution is* $O\big(\big(\nu\epsilon^{-1}\big)\ln\epsilon^{-1}\big)$.

The restart strategy also applies to the finite-sum case.

Theorem 6. *Consider problem* (2) *with* $r \equiv 0$. *Suppose Assumption 1 hold and* F *is* ν*-gradient dominant. In Algorithm 2, if we set* $\tau_t = S_t = \sqrt{B_t} = \lceil \sqrt{n} \rceil$ *and* $T = \lceil \frac{16\nu}{\sqrt{n}\eta} \rceil$, *then as long as* $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12G_0}}$,

$$\mathbf{E}\big[F(\bar{x}^{k+1}) - F^* | \bar{x}^k\big] \leq \frac{1}{2}\big(F(\bar{x}^k) - F^*\big). \tag{26}$$

As a result, the sample complexity for finding an ϵ*-solution is* $O\big(\big(n + \frac{\nu\sqrt{n}}{\eta}\big)\ln\frac{1}{\epsilon}\big)$.

It is worth noting that for both cases, the number of epochs $T \propto \eta^{-1}$. When we take more conservative values η, it will directly result in worse complexity results. This comment also applies to the optimally strongly convex objective function case in the next section.

4.2 Optimally strongly convex function

In this part, we assume a μ*-optimally strongly convex* condition on the function $\Phi(x) = F(x) + r(x)$, i.e., there exists a $\mu > 0$ such that

$$\Phi(x) - \Phi(x^*) \geq \frac{\mu}{2}\|x - x^*\|^2, \qquad \forall x \in \mathbf{R}^d. \tag{27}$$

We have the following two results for solving problems (1) and (2) respectively.

Theorem 7. *Consider problem* (1). *Suppose Assumptions 1 and 2 hold and* Φ *is* μ*-optimally strongly convex. In Algorithm 2, let us set* $\tau_t = S_t = \lceil \frac{1}{\sqrt{\epsilon}} \rceil$, $B_t = \lceil \frac{9\sigma_0^2}{2\mu\epsilon} \rceil$ *and* $T = \lceil \frac{5\sqrt{\epsilon}}{\mu\eta} \rceil$. *Then if we choose* $\eta < \frac{2}{L_F + \sqrt{L_F^2 + 36G_0}}$,

$$\mathbf{E}\big[\Phi(\bar{x}^{k+1}) - \Phi^*\big] \leq \frac{1}{2}\big(\Phi(\bar{x}^k) - \Phi^*\big) + \frac{1}{2}\epsilon. \tag{28}$$

Consquently, $\mathbf{E}[\Phi(\bar{x}^k) - \Phi^*]$ *converges linearly to* ϵ. *The total sample complexity for finding an* ϵ*-solution is* $O\big(\mu^{-1}\epsilon^{-1}\ln\epsilon^{-1}\big)$.

Theorem 8. *Consider the finite-sum problem* (2). *Suppose Assumption 1 hold and* Φ *is* μ*-optimally strongly convex. In Algorithm 2, let us set* $\tau_t = S_t = \sqrt{B_t} = \lceil \sqrt{n} \rceil$ *and* $T = \lceil \frac{5}{\sqrt{n}\mu\eta} \rceil$. *Then if we choose* $\eta < \frac{2}{L_F + \sqrt{L_F^2 + 36G_0}}$,

$$\mathbf{E}\big[\Phi(\bar{x}^{k+1}) - \Phi^*\big] \leq \frac{1}{2}\big(\Phi(\bar{x}^k) - \Phi^*\big). \tag{29}$$

The sample complexity of finding an ϵ*-solution is* $O\big(\big(n + \frac{\sqrt{n}}{\mu\eta}\big)\ln\frac{1}{\epsilon}\big)$.

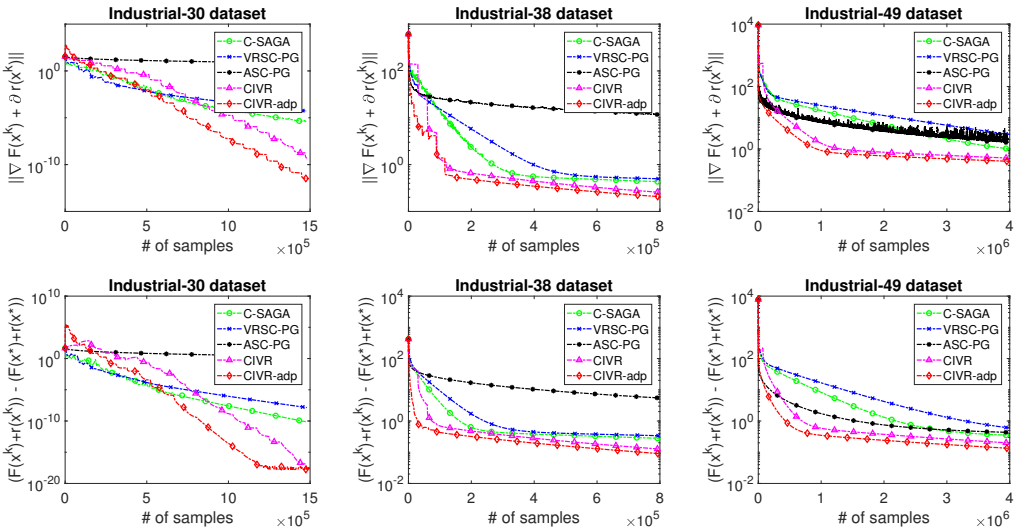

Figure 1: Experiments on the risk-averse portfolio optimization problem.

If we define a condition number $\kappa = L_F/\mu$, then since $\eta = \Theta(1/L_F)$, we have $1/(\mu\eta) = O(\kappa)$ and the above complexities become $O\left(\left(\kappa\epsilon^{-1}\right)\ln\epsilon^{-1}\right)$ and $O\left(\left(n+\kappa n^{1/2}\right)\ln\epsilon^{-1}\right)$.

For Algorithm 2 in both gradient-dominant and strongly convex cases, we have the following remarks.

Remark 1. *In Algorithm 2, each run of Algorithm 1 includes a random selection of output. In average it wastes half of the iterates. This waste can be prevented by pre-generating the "stop times" or the output indeces. We can stop Algorithm 1 and output the last iterate whenever the method hits this time.*

Remark 2. *In Algorithm 2, the linear-convergence is achieved by restarting. This strategy is proposed partly due to the epoch structure of Algorithm 1. Therefore, if we break this epoch structure by the loopless variance reduction techniques introduced in [16], the restarts may be avoided.*

5 Numerical Experiments

In this section, we present numerical experiments for a risk-averse portfolio optimization problem. Suppose there are d assets that one can invest during n time periods labeled as $\{1, ..., n\}$. Let $R_{i,j}$ be the return or payoff per unit of asset j at time i, and R_i be the vector consists of $R_{i,1}, \ldots, R_{i,d}$. Let $x \in \mathbf{R}^d$ be the decision variable, where each component x_j represent the amount of investment or percentage of the total investment allocated to asset j, for $j = 1, \ldots, d$. The same allocations or percentages of allocations are repeated over the n time periods. We would like to maximize the average return over the n periods, but with a penalty on the variance of the returns across the n periods (in other words, we would like different periods to have similar returns).

This problem is formulated as a mean-variance trade-off:

$$\underset{x\in\mathbf{R}^d}{\text{maximize}} \quad \left\{\mathbf{E}\left[h_\xi(x)\right] - \lambda\mathbf{Var}\left(h_\xi(x)\right) + r(x) \;\equiv\; \mathbf{E}\left[h_\xi(x)\right] - \lambda\left(\mathbf{E}\left[h_\xi^2(x)\right] - \mathbf{E}\left[h_\xi(x)\right]^2\right) + r(x)\right\},$$

where the random variable $\xi \in \{1, \ldots, n\}$ takes discrete values uniformly at random and hence makes the problem a finite-sum. The functions $h_i(x) = \langle R_i, x\rangle$ for $i = 1, \ldots, n$ are the rewards. The function r can be chosen as the indicator function of an ℓ_1 ball, or a soft ℓ_1 regularization term. We choose the latter one in our experiments to obtain a sparse asset allocation. By using the mappings

$$g_\xi(x) : \mathbf{R}^d \to \mathbf{R}^2 = \begin{bmatrix} h_\xi(x) & h_\xi^2(x)\end{bmatrix}^T, \qquad f(y,z) : \mathbf{R}^2 \to \mathbf{R} = -y + \lambda y^2 - \lambda z,$$

it can be further transformed into the composite finite-sum problem (2), hence readily solved by the CIVR method. Here, the intermediate dimension is very low, i.e., $p = 2$. This leads to very little overhead in computation compared with stochastic optimization without composition.

For comparison, we implement the C-SAGA algorithm [35] as a benchmark. As another benchmark, this problem can also be formulated as a two-layer composite finite-sum problem (7), which was done

in [11] and [19]. We solve the two-layer formulation by ASC-PG [32] and VRSC-PG [11]. Finally, we also implemented CIVR-adp, which is the adaptive sampling variant described in Theorem 4.

We test these algorithms on three real world portfolio datasets, which contain 30, 38 and 49 industrial portfolios respectively, from the Keneth R. French Data Library[1]. For the three datasets, the daily data of the most recent 24452, 10000 and 24400 days are extracted respectively to conduct the experiments. We set the parameter $\lambda = 0.2$ in (5) and use an ℓ_1 regularization $r(x) = 0.01\|x\|_1$. The experiment results are shown in Figure 1. The curves are averaged over 20 runs and are plotted against the number of samples of the component functions (the horizontal axis).

Throughout the experiments, VRSC-PG and C-SAGA algorithms use the batch size $S = \lceil n^{2/3} \rceil$ while CIVR uses the batch size $S = \lceil \sqrt{n} \rceil$, all dictated by their complexity theory. CIVR-adp employs the adaptive batch size $S_t = \left\lceil \min\{10t + 1, \sqrt{n}\} \right\rceil$ for $t = 1, ..., T$. For Industrial-30 dataset, all of VRSC-PG, C-SAGA, CIVR and CIVR-adp use the same step size $\eta = 0.1$. They are chosen from the set $\eta \in \{1, 0.1, 0.01, 0.001, 0.0001\}$ by experiments. And $\eta = 0.1$ works best for all four tested methods simultaneously. Similarly, $\eta = 0.001$ is chosen for the Industrial-38 dataset and $\eta = 0.0001$ is chosen for the Industrial-49 dataset. For ASC-PG, we set its step size parameters $\alpha_k = 0.001/k$ and $\beta_k = 1/k$ [see details in 32]. They are hand-tuned to ensure ASC-PG converges fast among a range of tested parameters. Overall, CIVR and CIVR-adp outperform other methods.

References

[1] Zeyuan Allen-Zhu. Natasha: Faster non-convex stochastic optimization via strongly non-convex parameter. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 89–97, Sydney, Australia, 2017.

[2] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. In *Advances in Neural Information Processing Systems 31*, pages 2675–2686. Curran Associates, Inc., 2018.

[3] Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 699–707, 2016.

[4] Amir Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. SIAM, 2017.

[5] Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: a survey and comparison. *Journal of Machine Learning Research*, 15(1):809–883, 2014.

[6] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27*, pages 1646–1654, 2014.

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

[8] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems 31*, pages 689–699. Curran Associates, Inc., 2018.

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

[10] Saeed Ghadimi, Andrzej Ruszczyński, and Mengdi Wang. A single time-scale stochastic approximation method for nested stochastic optimization. Preprint, arXiv:1812.01094, 2018.

[11] Zhouyuan Huo, Bin Gu, Ji Jiu, and Heng Huang. Accelerated method for stochastic composition optimization with nonsmooth regularization. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 3287–3294, 2018.

[1] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

[12] A. N. Iusem, A. Jofré, R. I. Oliveira, and P. Phompson. Extragradient method with variance reduction for stochastic variational inequalities. *SIAM Journal on Optimization*, 27(2):686–724, 2017.

[13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

[14] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient method and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Database - European Conference, Proceedings*, pages 795–811, 2016.

[15] J. Koshal, A. Nedić, and U. B. Shanbhag. Regularized iterative stochastic approximation methods for stochastic variational inequality problems. *IEEE Transactions on Automatic Control*, 58(3): 594–609, 2013.

[16] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. *arXiv preprint arXiv:1901.08689*, 2019.

[17] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via SCSG methods. In *Advances in Neural Information Processing Systems 30*, pages 2348–2358. Curran Associates, Inc., 2017.

[18] Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems 31*, pages 5564–5574. Curran Associates, Inc., 2018.

[19] Xiangru Lian, Mengdi Wang, and Ji Liu. Finite-sum composition optimization via variance reduced gradient descent. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1159–1167, 2017.

[20] Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[21] Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research (PMLR)*, pages 2613–2621, Sydney, Australia, 2017.

[22] Lam M. Nguyen, Marten van Dijk, Dzung T. Phan, Phuong Ha Nguyen, Tsui-Wei Weng, and Jayant R. Kalagnanam. Finite-sum smooth optimization with sarah. arXiv preprint, arXiv:1901.07648, 2019.

[23] Nhan H. Pham, Lam M. Nguyen, Dzung T. Phan, and Quoc Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. arXiv preprint, arXiv:1902.05679, 2019.

[24] Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 314–323, New York, New York, USA, 2016.

[25] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for smooth nonconvex optimization. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1971–1977. IEEE, 2016.

[26] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems 29*, pages 1145–1153, 2016.

[27] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[28] R. Tyrrell Rockafellar. Coherent approaches to risk in optimization under uncertainty. *INFORMS TutORials in Operations Research*, 2007.

[29] Andrzej Ruszczyński. Advances in risk-averse optimization. *INFORMS TutORials in Operation Research*, 2013.

[30] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

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

[32] Mengdi Wang, Ji Liu, and Ethan Fang. Accelerating stochastic composition optimization. *Journal of Machine Learning Research*, 18(105):1–23, 2017.

[33] Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. SpiderBoost: A class of faster variance-reduced algorithms for nonconvex optimization. arXiv preprint, arXiv:1810.10690, 2018.

[34] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[35] Junyu Zhang and Lin Xiao. A composite randomized incremental gradient method. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, number 97 in Proceedings of Machine Learning Research (PMLR), Long Beach, California, 2019.

[36] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduced gradient descent for nonconvex optimization. In *Advances in Neural Information Processing Systems 31*, pages 3921–3932. Curran Associates, Inc., 2018.

Appendices

A Convergence analysis for composite expectation case

In this section, we focus on convergence analysis of CIVR for solving the stochastic composite optimization problem (1), and prove Theorems 1 and 2.

First, we show that under Assumption 1, the composite function $F(x) = f(g(x))$ is smooth and F' has Lipschitz constant $L_f = \ell_g^2 L_f + \ell_f L_g$.

$$\begin{aligned}
\|F'(x) - F'(y)\| &= \left\|g'(x)^T f'(g(x)) - g'(y)^T f'(g(y))\right\| \\
&= \left\|g'(x)^T f'(g(x)) - g'(x)^T f'(g(y)) + g'(x)^T f'(g(y)) - g'(y)^T f'(g(y))\right\| \\
&\leq \left\|g'(x)^T f'(g(x)) - g'(x)^T f'(g(y))\right\| + \left\|g'(x)^T f'(g(y)) - g'(y)^T f'(g(y))\right\| \\
&\leq \|g'(x)\| \, \|f'(g(x)) - f'(g(y))\| + \|f'(g(y))\| \, \|g'(x) - g'(y)\| \\
&\leq \|g'(x)\| \cdot L_f \, \|g(x) - g(y)\| + \|f'(g(y))\| \cdot L_g \, \|x - y\| \\
&\leq \ell_g L_f \ell_g \, \|x - y\| + \ell_f L_g \, \|x - y\| \\
&= \left(\ell_g^2 L_f + \ell_f L_g\right) \|x - y\|,
\end{aligned}$$

where we used $\|g'(x)\| \leq \ell_g$ and $\|f'(g(y))\| \leq \ell_f$, which are implied by the Lipschitz conditions on g and f respectively.

Although the incremental estimators used in CIVR are biased, as shown in (16), we can still bound their squared distances from the targets. This is given in the following lemma.

Lemma 1. *Suppose Assumption 1 holds. Let y_i^t and z_i^t be constructed according to* (8) *and* (9) *in Algorithm 1. For any $t \geq 1$ and $1 \leq i \leq \tau_t - 1$, we have the following mean squared error (MSE) bounds*

$$\begin{cases}
\mathbf{E}\left[\|y_i^t - g(x_i^t)\|^2\right] \leq \mathbf{E}\left[\|y_0^t - g(x_0^t)\|^2\right] + \sum_{r=1}^{i} \frac{\ell_g^2}{S_t} \mathbf{E}\left[\|x_r^t - x_{r-1}^t\|^2\right], \\
\mathbf{E}\left[\|z_i^t - g'(x_i^t)\|^2\right] \leq \mathbf{E}\left[\|z_0^t - g'(x_0^k)\|^2\right] + \sum_{r=1}^{i} \frac{L_g^2}{S_t} \mathbf{E}\left[\|x_r^t - x_{r-1}^t\|^2\right].
\end{cases} \tag{30}$$

Proof. We first state a fact that allows us to decompose the MSE into a squared bias term and a variance term, that is, for an arbitrary random vector ζ and a constant vector u, we have

$$\mathbf{E}[\|\zeta - u\|^2] = \|\mathbf{E}[\zeta] - u\|^2 + \mathbf{Var}(\zeta), \tag{31}$$

where $\mathbf{Var}(\zeta) := \mathbf{E}[\|\zeta - \mathbf{E}[\zeta]\|^2]$. As a result,

$$\mathbf{E}\left[\|y_i^t - g(x_i^t)\|^2 | x_i^t\right] = \left\|\mathbf{E}[y_i^t | x_i^t] - g(x_i^t)\right\|^2 + \mathbf{Var}\left(y_i^t | x_i^t\right).$$

For the bias term, we have $\mathbf{E}[y_i^t | x_i^t] - g(x_i^t) = y_{i-1}^t - g(x_{i-1}^t)$. For the variance term, we have

$$\begin{aligned}
\mathbf{Var}\left(y_i^t | x_i^t\right) &= \mathbf{Var}\left(y_{i-1}^t + \frac{1}{S_t} \sum_{\xi \in \mathcal{S}_i^t} \left(g_\xi(x_i^t) - g_\xi(x_{i-1}^t)\right) \,\middle|\, x_i^t\right) \\
&= \frac{1}{S_t} \mathbf{Var}\left(g_\xi(x_i^t) - g_\xi(x_{i-1}^t) \mid x_i^t\right) \\
&\leq \frac{1}{S_t} \mathbf{E}\left[\|g_\xi(x_i^t) - g_\xi(x_{i-1}^t)\|^2 | x_i^t\right] \\
&\leq \frac{\ell_g^2}{S_t} \|x_i^t - x_{i-1}^t\|^2,
\end{aligned}$$

where the second equality is due to the fact that y_{i-1}^t is a constant conditioning on x_i^t and in the last inequality we used the ℓ_g-Lipschitz continuity of g_ξ. Consequently,

$$\mathbf{E}\left[\|y_i^t - g(x_i^t)\|^2\right] \leq \mathbf{E}\left[\|y_{i-1}^t - g(x_{i-1}^t)\|^2\right] + \frac{\ell_g^2}{S_t} \mathbf{E}\left[\|x_i^t - x_{i-1}^t\|^2\right].$$

Recursively applying the above procedure yields

$$\mathbf{E}\left[\|y_i^t - g(x_i^t)\|^2\right] \le \mathbf{E}\left[\|y_0^t - g(x_0^t)\|^2\right] + \sum_{r=1}^{i} \frac{\ell_g^2}{S_t}\mathbf{E}\left[\|x_r^t - x_{r-1}^t\|^2\right]. \tag{32}$$

Similarly, the bound on $\mathbf{E}\left[\|z_i^t - g'(x_i^t)\|^2\right]$ can be shown by using the L_g-Lipschitz continuity of g'_ξ. □

In Algorithm 1, we approximate the gradient of $F(x) := f(g(x))$ by $\tilde{\nabla}F(x_i^t) = (z_i^t)^T f'(y_i^t)$. The next lemma bounds the MSE of this estimator.

Lemma 2. *Suppose Assumptions 1 and 2 hold. Then we have*

$$\mathbf{E}[\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2] \le \frac{G_0}{S_t}\sum_{r=1}^{i}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2] + \frac{\sigma_0^2}{B_t}, \tag{33}$$

where

$$G_0 := 2\left(\ell_g^4 L_f^2 + \ell_f^2 L_g^2\right) \quad \textit{and} \quad \sigma_0^2 := 2\left(\ell_g^2 L_f^2 \sigma_g^2 + \ell_f^2 \sigma_{g'}^2\right).$$

Proof. Using Assumption 1, one immediately gets

$$\begin{aligned}
& \mathbf{E}\left[\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2\right] \\
= \;& \mathbf{E}\left[\|(z_i^t)^T f'(y_i^t) - (g'(x_i^t))^T f'(g(x_i^t))\|^2\right] \\
= \;& \mathbf{E}\left[\|(z_i^t)^T f'(y_i^t) - (g'(x_i^t))^T f'(y_i^t) + (g'(x_i^t))^T f'(y_i^t) - (g'(x_i^t))^T f'(g(x_i^t))\|^2\right] \\
\le \;& 2\mathbf{E}\left[\|(z_i^t)^T f'(y_i^t) - (g'(x_i^t))^T f'(y_i^t)\|^2\right] + 2\mathbf{E}\left[\|(g'(x_i^t))^T f'(y_i^t) - (g'(x_i^t))^T f'(g(x_i^t))\|^2\right] \\
\le \;& 2\ell_f^2\mathbf{E}\left[\|z_i^t - g(x_i^t)\|^2\right] + 2\ell_g^2 L_f^2\mathbf{E}\left[\|y_i^t - g'(x_i^t)\|^2\right].
\end{aligned} \tag{34}$$

Therefore, by substituting the MSE bounds provided in Lemma 1 into inequality (34), we obtain

$$\begin{aligned}
\mathbf{E}\left[\|\tilde{\nabla}F(x_i^t) - F'(x)\|^2\right] \;\le\; & \frac{2\left(\ell_g^4 L_f^2 + \ell_f^2 L_g^2\right)}{S_t}\sum_{r=1}^{i}\mathbf{E}\left[\|x_r^t - x_{r-1}^t\|^2\right] \\
& + 2\ell_g^2 L_f^2\mathbf{E}\left[\|y_0^t - g(x_0^t)\|^2\right] + 2\ell_f^2\mathbf{E}\left[\|z_0^t - g'(x_0^k)\|^2\right].
\end{aligned} \tag{35}$$

Under Assumption 2, we can bound the MSE of the estimates in (8) as

$$\mathbf{E}\left[\|y_0^t - g(x_0^t)\|^2\right] \le \frac{\sigma_g^2}{B_t}, \qquad \mathbf{E}\left[\|z_0^t - g'(x_0^k)\|^2\right] \le \frac{\sigma_{g'}^2}{B_t}.$$

Combining these MSE bounds with (35) yields the desired result. □

For the proximal gradient type of algorithms, no matter deterministic or stochastic, a common metric to quantify the optimality of x_i^t is the norm of the so-called *proximal gradient mapping*

$$\mathcal{G}_\eta(x_i^t) := \frac{1}{\eta}\left(x_i^t - \hat{x}_{i+1}^t\right), \tag{36}$$

where η is the step size used to produce the update

$$\hat{x}_{i+1}^t = \mathbf{prox}_r^\eta\left(x_i^t - \eta F'(x_i^t)\right).$$

Since we use a constant η throughout this paper, we will omit the subscript η and use $\mathcal{G}(x)$ to denote the proximal gradient mapping at x.

Our goal is to find a point x with $\mathbf{E}\left[\|\mathcal{G}(x)\|^2\right] \le \epsilon$. However, in Algorithm 1, we only have the approximate proximal gradient mapping

$$\tilde{\mathcal{G}}(x_i^t) := \frac{1}{\eta}(x_i^t - x_{i+1}^t), \tag{37}$$

where x_{i+1}^t is computed using the estimated gradient $\tilde{\nabla}F(x_i^t)$:

$$x_{i+1}^t = \mathbf{prox}_r^\eta\left(x_i^t - \eta\tilde{\nabla}F(x_i^t)\right).$$

Hence we need to establish the connection between $\mathcal{G}(x_i^t)$ and $\tilde{\mathcal{G}}(x_i^t)$, which is done in the next lemma.

Lemma 3. *For the two gradient mappings defined in* (36) *and* (37)*, we have*

$$\mathbf{E}\left[\|\mathcal{G}(x_i^t)\|^2\right] \le 2\mathbf{E}\left[\|\tilde{\mathcal{G}}(x_i^t)\|^2\right] + 2\mathbf{E}\left[\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2\right]. \tag{38}$$

Proof. Using the inequality $\|x_i^t - \hat{x}_{i+1}^t\|^2 \le 2\|x_i^t - x_{i+1}^t\|^2 + 2\|x_{i+1}^t - \hat{x}_{i+1}^t\|^2$ and the definitions of $\mathcal{G}(x_i^t)$ and $\tilde{\mathcal{G}}(x_i^t)$, we have

$$\begin{aligned}
\mathbf{E}\left[\|\mathcal{G}(x_i^t)\|^2\right] &\le 2\mathbf{E}\left[\|\tilde{\mathcal{G}}(x_i^t)\|^2\right] + \frac{2}{\eta^2}\left\|x_{i+1}^t - \hat{x}_{i+1}^t\right\|^2 \\
&= 2\mathbf{E}\left[\|\tilde{\mathcal{G}}(x_i^t)\|^2\right] + \frac{2}{\eta^2}\left\|\mathbf{prox}_r^\eta\left(x_i^t - \eta F'(x_i^t)\right) - \mathbf{prox}_r^\eta\left(x_i^t - \eta\tilde{\nabla}F(x_i^t)\right)\right\|^2 \\
&\le 2\mathbf{E}\left[\|\tilde{\mathcal{G}}(x_i^t)\|^2\right] + \frac{2}{\eta^2}\left\|x_i^t - \eta F'(x_i^t) - \left(x_i^t - \eta\tilde{\nabla}F(x_i^t)\right)\right\|^2 \\
&= 2\mathbf{E}\left[\|\tilde{\mathcal{G}}(x_i^t)\|^2\right] + 2\mathbf{E}\left[\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2\right],
\end{aligned}$$

where in the second inequality we used the non-expansive property of proximal mapping [e.g., 27, Section 31]. □

The next lemma bounds the amount of expected descent per iteration in Algorithm 1.

Lemma 4. *Let the sequence $\{x_i^t\}$ be generated by Algorithm 1. Then for all $t \ge 1$ and $0 \le i \le \tau_t - 1$, we have the following two inequalities*

$$\mathbf{E}[\Phi(x_{i+1}^t)] \le \mathbf{E}[\Phi(x_i^t)] - \left(\frac{\eta}{2} - \frac{L_F\eta^2}{2}\right)\mathbf{E}\left[\|\tilde{\mathcal{G}}(x_i^t)\|^2\right] + \frac{\eta}{2}\mathbf{E}[\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2], \tag{39}$$

and

$$\begin{aligned}
\mathbf{E}[\Phi(x_{i+1}^t)] \le\ & \mathbf{E}[\Phi(x_i^t)] - \frac{\eta}{8}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] + \frac{3\eta}{4}\mathbf{E}[\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2] \\
&- \left(\frac{1}{4\eta} - \frac{L_F}{2}\right)\mathbf{E}\left[\|x_i^t - x_{i+1}^t\|^2\right].
\end{aligned} \tag{40}$$

Proof. By applying the L_F-Lipschitz continuity of F' and the optimality of the $\frac{1}{\eta}$-strongly convex subproblem, we have

$$\begin{aligned}
\Phi(x_{i+1}^t) &= F(x_{i+1}^t) + r(x_{i+1}^t) \\
&\le F(x_i^t) + \langle F'(x_i^t), x_{i+1}^t - x_i^t\rangle + \frac{L_F}{2}\|x_{i+1}^t - x_i^t\|^2 + r(x_{i+1}^t) \\
&= F(x_i^t) + \langle\tilde{\nabla}F(x_i^t), x_{i+1}^t - x_i^t\rangle + \frac{1}{2\eta}\|x_{i+1}^t - x_i^t\|^2 + r(x_{i+1}^t) \\
&\quad + \langle F'(x_i^t) - \tilde{\nabla}F(x_i^t), x_{i+1}^t - x_i^t\rangle - (\frac{1}{2\eta} - \frac{L_F}{2})\|x_{i+1}^t - x_i^t\|^2 \\
&\le F(x_i^t) + r(x_i^t) - \frac{1}{2\eta}\|x_{i+1}^t - x_i^t\|^2 - (\frac{1}{2\eta} - \frac{L_F}{2})\|x_{i+1}^t - x_i^t\|^2 \\
&\quad + \frac{\eta}{2}\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2 + \frac{1}{2\eta}\|x_{i+1}^t - x_i^t\|^2 \\
&= \Phi(x_i^t) - (\frac{1}{2\eta} - \frac{L_F}{2})\|x_{i+1}^t - x_i^t\|^2 + \frac{\eta}{2}\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2.
\end{aligned}$$

Taking the expectation on both sides completes the proof of inequality (39). By inequality (38), we know that

$$-\frac{\eta}{4}\mathbf{E}[\|\tilde{\mathcal{G}}(x_i^t)\|^2] \le -\frac{\eta}{8}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] + \frac{\eta}{4}\mathbf{E}[\|\tilde{\nabla}F(x_i^t) - F'(x_i^t)\|^2].$$

Adding this inequality in to (39) yields (40). □

A.1 Proof of Theorem 1

Proof. Because all τ_t, B_t and S_t are taking their values independent of t. We denote $\tau = \tau_t$, $B = B_t$ and $S = S_t$ for all t for clarity. By Lemma 4, summing up inequality (40) throughout the t-th epoch and applying (33) gives

$$\begin{aligned}\frac{\eta}{8}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] &\leq \mathbf{E}[\Phi(x_0^t)] - \mathbf{E}[\Phi(x_\tau^t)] - \left(\frac{1}{4\eta} - \frac{L_F}{2}\right)\sum_{r=1}^{\tau}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2] \\ &\quad + \frac{3G_0\eta}{4S}\sum_{i=1}^{\tau-1}\sum_{r=1}^{i}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2] + \frac{3\sigma_0^2\eta}{4B}\tau \\ &\leq \mathbf{E}[\Phi(x_0^t)] - \mathbf{E}[\Phi(x_\tau^t)] - \left(\frac{1}{4\eta} - \frac{L_F}{2} - \tau\frac{3G_0\eta}{4S}\right)\sum_{r=1}^{\tau}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2] \\ &\quad + \frac{3\sigma_0^2\eta}{4B}\tau,\end{aligned}$$

where the second inequality is due to the fact that

$$\sum_{i=1}^{\tau-1}\sum_{r=1}^{i}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2] \leq \tau\sum_{r=1}^{\tau}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2].$$

When we choose the parameters satisfying $\tau \leq S$, then the coefficient $\frac{1}{4\eta} - \frac{L_F}{2} - \tau\frac{3G_0\eta}{4S} \geq \frac{1}{4\eta} - \frac{L_F}{2} - \frac{3G_0\eta}{4}$ which depends only on the parameter η and some constant. If we choose the η according to the theorem, then $\frac{1}{4\eta} - \frac{L_F}{2} - \frac{3G_0\eta}{4} \geq 0$, yielding that

$$\frac{\eta}{8}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \;\leq\; \mathbf{E}[\Phi(x_0^t)] - \mathbf{E}[\Phi(x_\tau^t)] + \frac{3\sigma_0^2\eta}{4B}\tau. \tag{41}$$

Summing this up throughout the epochs gives

$$\frac{\eta}{8}\sum_{t=1}^{T}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \;\leq\; \mathbf{E}[\Phi(x_0^1)] - \mathbf{E}[\Phi(x_\tau^T)] + \frac{3\sigma_0^2\eta}{4B}\tau T \leq \Phi(x_0^1) - \Phi^* + \frac{3\sigma_0^2\eta}{4B}\tau T,$$

where we have applied the fact that $x_0^t = x_\tau^{t-1}$. By the random sampling scheme for output $\bar{x}$, we have

$$\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] = \frac{1}{\tau T}\sum_{t=1}^{T}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \leq \frac{8(\Phi(x_0^1) - \Phi^*)}{\tau T\eta} + \frac{6\sigma_0^2}{B}. \tag{42}$$

Substitute the values of T, τ and B gives (19). □

To simplify presentation, we omit $\lceil\cdot\rceil$ on integer parameters in the following discussion.

- With $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12G_0}}$, and letting $T = 1/\sqrt{\epsilon}$, $B = \sigma_0^2/\epsilon$, and $\tau = S = 1/\sqrt{\epsilon}$, we have

  $$\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] \leq 8\big((\Phi(x_0^1) - \Phi^*)\eta^{-1} + 6\big)\epsilon,$$

  and the sample complexity is $T(B + 2\tau S) = O\big(\sigma_0^2\epsilon^{-3/2} + \epsilon^{-3/2}\big)$, as in our theorem.

- With $\eta \leq \frac{4}{L_F + \sqrt{L_F^2 + 12G_0}}$, and letting $T = 1/\epsilon$, $B = 1 + \sigma_0^2/\epsilon$, and $\tau = S = 1$, we again obtain

  $$\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] \leq 8\big((\Phi(x_0^1) - \Phi^*)\eta^{-1} + 6\big)\epsilon,$$

  but the sample complexity is $T(B + 2\tau S) = O\big(\sigma_0^2\epsilon^{-2} + \epsilon^{-1}\big)$, which is same as in Ghadimi and Lan [9]. For deterministic optimization with $\sigma_0 = 0$, this recovers the $O(\epsilon^{-1})$ complexity.

A.2 Proof of Theorem 2

Proof. Note that for this set of parameters, we still have the relationship that $\tau_t = S_t$. Therefore, within each epoch, (41) is still true with epoch specific τ_t and B_t. Summing this up gives

$$\frac{\eta}{8}\sum_{t=1}^{T}\sum_{i=0}^{\tau_t-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \le \Phi(x_0^1) - \Phi^* + \sum_{t=1}^{T}\frac{3\sigma_0^2\eta}{4B_t}\tau_t. \tag{43}$$

By the random selection rule of $\bar{x}$, we have

$$\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] = \frac{1}{\sum_{t=1}^{T}\tau_t}\sum_{t=1}^{T}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \le \frac{8(\Phi(x_0^1)-\Phi^*)}{\eta\sum_{t=1}^{T}\tau_t} + 6\sigma_0^2\cdot\frac{\sum_{t=1}^{T}\tau_t/B_t}{\sum_{t=1}^{T}\tau_t}. \tag{44}$$

Note that $\tau_t = \lceil at+b\rceil$ and $B_t = \lceil\sigma_0^2(at+b)^2\rceil$. We have

$$\sum_{t=1}^{T}\tau_t \ge \sum_{t=1}^{T} at+b = \frac{a}{2}T(T+1) + bT = O(T^2)$$

and

$$\sigma_0^2\sum_{t=1}^{T}\tau_t/B_t \le \sum_{t=1}^{T}\frac{1}{at+b} \le \frac{1}{a+b} + \int_1^T\frac{dt}{at+b} = \frac{1}{a+b} + \frac{1}{a}\ln\left(\frac{aT+b}{a+b}\right) = O(\ln T).$$

Substituting the above bounds into inequality (44) gives (20). As a result, the total sample complexity is

$$\sum_{t=1}^{T}\left(B_t + 2\tau_t S_t\right) \le \sum_{t=1}^{T}\left(\sigma_0^2(at+b)^2 + 2(at+b)^2\right) = O(\sigma_0^2T^3 + T^3)\,.$$

Setting $T = \tilde{O}(\epsilon^{-1/2})$ so that $\mathbf{E}[\|\mathcal{G}\|\bar{x}\|^2] \le \epsilon$, we get sample complexity $\tilde{O}(\sigma_0^2\epsilon^{-3/2} + \epsilon^{-3/2})$. □

We can also choose a different set of parameters. With $\eta \le \frac{4}{L_F+\sqrt{L_F^2+12G_0}}$, and letting $B = 1+\sigma_0^2(at+b)$, and $\tau = S = 1$, we also have

$$\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] \le \frac{8(\Phi(x_0^1)-\Phi^*)}{\eta T} + \frac{6\ln T}{T},$$

but the sample complexity, by setting $T = \tilde{O}(\epsilon^{-1})$ so that the above bound is less than ϵ, is

$$\sum_{t=1}^{T}\left(B_t + 2\tau_t S_t\right) \le \sum_{t=1}^{T}\left(\sigma_0^2(at+b) + 2\right) = O(\sigma_0^2T^2 + T) = \tilde{O}(\sigma_0^2\epsilon^{-2} + \epsilon^{-1})\,.$$

This is more close to the classical results on stochastic optimization.

B Convergence analysis for composite finite-sum case

In this section, we consider the composite finite-sum problem (2) and prove Theorems 3 and 4.

In this case, the random variable ξ uniformly takes value from the finite index set $\{1, ..., n\}$. At the beginning of each epoch in Algorithm 1, we can choose to estimate $g(x_0^t)^t$ and $g'(x_0^t)$ by their exact value rather than the approximate ones constructed by subsampling. Namely, in (8) of Algorithm 1, we choose $\mathcal{B}_t = \{1, \ldots, n\}$ for all $t \ge 1$. Therefore,

$$y_0^t = g(x_0^t) = \frac{1}{n}\sum_{j=1}^{n} g_j(x_0^t), \qquad z_0^t = g'(x_0^t) = \frac{1}{n}\sum_{j=1}^{n} g_j'(x_0^t)$$

and

$$\mathbf{E}\left[\|y_0^t - g(x_0^t)\|^2\right] = 0\,, \qquad \mathbf{E}\left[\|z_0^t - g'(x_0^k)\|^2\right] = 0\,. \tag{45}$$

As a result, the initial variances in Lemma 1 diminishes and (30) reduces to

$$\begin{cases}\mathbf{E}[\|y_i^t - g(x_i^t)\|^2] \le \sum_{r=1}^{i}\frac{\ell_g^2}{S_t}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2],\\ \mathbf{E}[\|z_i^t - g'(x_i^t)\|^2] \le \sum_{r=1}^{i}\frac{L_g^2}{S_t}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2].\end{cases} \tag{46}$$

In addition, combining (35) and (45), we have

$$\mathbf{E}[\|\tilde{\nabla}F(x_i^t) - F'(x)\|^2] \le \frac{G_0}{S_t}\sum_{r=1}^{i}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2]. \tag{47}$$

Note that Lemma 4 is still true.

B.1 Proof of Theorem 3

Proof. The proof follows similar steps as those in the proof of Theorem 1. So we only note down the significantly different steps here.

Specifically, following the proof of Theorem 1 in Section A.1, by applying (46) instead of (30), we get the following result instead of inequality (41),

$$\frac{\eta}{8}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \le \mathbf{E}[\Phi(x_0^t)] - \mathbf{E}[\Phi(x_\tau^t)].$$

Summing this up apply the random selection rule of $\bar{x}$ gives

$$\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] = \frac{1}{\tau T}\sum_{t=1}^{T}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \le \frac{8(\Phi(x_0^1) - \Phi^*)}{\tau T\eta} = \frac{8(\Phi(x_0^1) - \Phi^*)}{\sqrt{n}T\eta}.$$

Therefore, we have to set $T = O(\frac{1}{\sqrt{n}\epsilon})$ to get an ϵ-solution. Note that the sample complexity per epoch is $n + \tau_t S_t = 2n$, the total sample complexity will be $O(n + \sqrt{n}\epsilon^{-1})$. □

B.2 Proof of Theorem 4

Proof. If $T \le T_0$, then the result is exactly what we proved from Theorem 2. Therefore, the first bound in (23) is already guaranteed.

If $T > T_0$, when $1 \le t \le T_0$, then everything still runs identically to that described in Theorem 2. Consequently, the following bound is effective

$$\frac{\eta}{8}\sum_{t=1}^{T_0}\sum_{i=0}^{\tau_t-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \le \Phi(x_0^1) - \mathbf{E}[\Phi(x_0^{T_0+1})] + \sum_{t=1}^{T_0}\frac{3\sigma_0^2\eta}{4B_t}\tau_t. \tag{48}$$

When $T_0 + 1 \le t \le T$, the following bound becomes effective,

$$\frac{\eta}{8}\sum_{t=T_0+1}^{T}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \le \mathbf{E}[\Phi(x_0^{T_0+1})] - \Phi^*.$$

Therefore, we have

$$\mathbf{E}[\|\mathcal{G}(\bar{x})\|^2] = \frac{1}{\sum_{t=1}^{T}\tau_t}\sum_{t=1}^{T}\sum_{i=0}^{\tau-1}\mathbf{E}[\|\mathcal{G}(x_i^t)\|^2] \le \frac{8(\Phi(x_0^1) - \Phi^*)}{\eta\sum_{t=1}^{T}\tau_t} + 6\sigma_0^2\cdot\frac{\sum_{t=1}^{T_0}\tau_t/B_t}{\sum_{t=1}^{T}\tau_t}.$$

Note that

$$\sum_{t=1}^{T_0}\tau_t/B_t \le \sum_{t=1}^{T_0}\frac{1}{at+b} \le \frac{1}{a+b} + \frac{1}{a}\ln\left(\frac{aT_0+b}{a+b}\right) = O(\ln n),$$

and

$$\sum_{t=1}^{T}\tau_t \ge (T-T_0)\sqrt{n} + \sum_{t=1}^{T_0}(at+b) = \sqrt{n}(T-T_0) + \frac{a}{2}T_0^2 + (\frac{a}{2}+b)T_0 = O(\sqrt{n}(T-T_0+1)).$$

With the above two bounds, we have proved the second result in (23).

For any $\epsilon > 0$, if $\epsilon \ge O(1/T_0^2) = O(n^{-1})$. In this case, the algorithm will spend most epochs in the adaptive phase, whose sample complexity is $\tilde{O}(\epsilon^{-3/2})$. if $\epsilon = o(n^{-1})$, we need $T > T_0$. By (23), we know $\sqrt{n}(T - T_0 + 1) = \tilde{O}(\epsilon^{-1})$, this means that the total sample complexity will be

$$\sum_{t=1}^{T}\big(B_t + 2\tau_t S_t\big) \le 3\sum_{t=1}^{T_0}(at+b+1)^2 + 3(T-T_0)n = \tilde{O}(n^{3/2} + \sqrt{n}\epsilon^{-1}) = \tilde{O}(\sqrt{n}\epsilon^{-1}).$$

When $\epsilon \ge O(n^{-1})$, we have $\epsilon^{-3/2} \le \sqrt{n}\epsilon^{-1}$. When $\epsilon = o(n^{-1})$, we have $\epsilon^{-3/2} > \sqrt{n}\epsilon^{-1}$. Combining the two cases together gives the sample complexity of $\tilde{O}(\min\{\sqrt{n}\epsilon^{-1}, \epsilon^{-3/2}\})$. □

C Convergence analysis under gradient-dominant condition

C.1 Proof of Theorem 5

Proof. For the ease of notation, let us only focus in one run of Algorithm 1. And denote the input point as x_0^1 and the output point $\bar{x}$. Note that in this case $\Phi(x) = F(x)$. By (42) and (24), we have

$$\mathbf{E}[F(\bar{x}) - F^*] \leq \nu \mathbf{E}[\|F'(\bar{x})\|^2] \leq \frac{8\nu(F(x_0^1) - F^*)}{\tau T \eta} + \frac{6\nu\sigma_0^2}{B}$$

By the selection of $T = \lceil \frac{16\nu\sqrt{\epsilon}}{\eta} \rceil$, $\tau = 1/\sqrt{\epsilon}$ and $B = 1 + 12\nu\sigma_0^2/\epsilon$, we have

$$\mathbf{E}[F(\bar{x}) - F^*] \leq \frac{1}{2}(F(x_0^1) - F^*) + \frac{1}{2}\epsilon, \tag{49}$$

which is (25).

Suppose we periodically restart the Algorithm 1 after every Tepochs, and set the outputs to be $\bar{x}^k$, where $k = 1, 2, \ldots$ denotes the number of restarts. We use the output of the kth period $\bar{x}^k$ as the initial point to start the next period, which produces $\bar{x}^{k+1}$. As a result, the above inequality translates to

$$\mathbf{E}[F(\bar{x}^{k+1}) - F^*] \leq \frac{1}{2}(\mathbf{E}[F(\bar{x}^k)] - F^*) + \frac{1}{2}\epsilon.$$

Equivalently,

$$\mathbf{E}[F(\bar{x}^k) - F^*] - \epsilon \leq \frac{1}{2}\left(\mathbf{E}[F(\bar{x}^{k-1}) - F^*] - \epsilon\right),$$

which leads to

$$\mathbf{E}[F(\bar{x}^k) - F^*] \leq \frac{1}{2^k}\left(\mathbf{E}[F(\bar{x}^0) - F^*] - \epsilon\right) + \epsilon.$$

Therefore, the expected optimality gap converges linearly to a ϵ-ball around 0. □

Next we discuss the sample complexity with different parameter settings.

- If we choose $\tau = S = 1/\sqrt{\epsilon}$, $B_t = 12\nu\sigma_0^2/\epsilon$, and $T = \lceil \frac{16\nu\sqrt{\epsilon}}{\eta} \rceil$, then the total sample complexity is

  $$T(B + 2\tau S)\ln\frac{1}{\epsilon} = \frac{16\nu\sqrt{\epsilon}}{\eta}\left(\frac{12\nu\sigma_0^2}{\epsilon} + \frac{1}{\sqrt{\epsilon}}\frac{1}{\sqrt{\epsilon}}\right)\ln\frac{1}{\epsilon} = O\left((\nu^2\sigma_0^2\epsilon^{-1/2} + \nu\epsilon^{-1/2})\ln\epsilon^{-1}\right)$$

  However, the above derivation needs to assume $\frac{16\nu\sqrt{\epsilon}}{\eta} \geq 1$ or at least $O(1)$, which means $\epsilon > (\eta/\nu)^2$. If this condition is not satisfied, then we have $T = 1$ and the complexity is

  $$O\big((\nu\sigma_0^2\epsilon^{-1} + \epsilon^{-1})\ln\epsilon^{-1}\big).$$

  Notice that the second term does not depend on ν or the conditions number.
- If we choose $\tau = S = 1$, $B_t = 1 + 12\nu\sigma_0^2/\epsilon$, and $T = \lceil \frac{16\nu}{\eta} \rceil$, the we also have

  $$\mathbf{E}[F(\bar{x}) - F^*] \leq \frac{1}{2}(F(x_0^1) - F^*) + \frac{1}{2}\epsilon,$$

  and the total sample complexity is

  $$T(B + 2\tau S)\ln\frac{1}{\epsilon} = \frac{16\nu}{\eta}\left(\frac{12\nu\sigma_0^2}{\epsilon} + 2\right)\ln\frac{1}{\epsilon} = O\left(\nu^2\sigma_0^2\epsilon^{-1/2} + \nu\right)\ln\epsilon^{-1}$$

  Defining the condition number $\kappa = L_F\nu = O(\nu/\eta)$, the above complexity becomes

  $$T(B + 2\tau S)\ln\frac{1}{\epsilon} = O\left(\kappa^2\sigma_0^2\epsilon^{-1} + \kappa\right)\ln\epsilon^{-1}$$

  Thus when $\sigma = 0$, we have $O\big(\kappa\ln\epsilon^{-1}\big)$ for deterministic optimization.

C.2 Proof of Theorem 6

The proof is very similar to the previous one. It actually becomes simpler by noticing that in the finite-sum case, the terms involving σ_0^2 disappear:

$$\mathbf{E}[F(\bar{x}) - F^*] \leq \nu \mathbf{E}[\|F'(\bar{x})\|^2] \leq \frac{8\nu(F(x_0^1) - F^*)}{\tau T \eta}.$$

By choosing $T = \left\lceil \frac{16\nu}{\eta\sqrt{n}} \right\rceil$, $\tau = S = \sqrt{n}$. we again obtain (49). In this case, we have $B = n$ and

$$T(B + 2\tau S)\epsilon^{-1} = \left\lceil \frac{16\nu}{\eta\sqrt{n}} \right\rceil \left(n + 2\sqrt{n}\sqrt{n}\right) \ln \epsilon^{-1} = O\left(n + \nu\sqrt{n}\right) \ln \epsilon^{-1}.$$

D Convergence analysis under optimally strong convexity

In order to prove Theorems 7 and 8, we first state Lemma 3 in [34] in our notations.

Lemma 5 (Lemma 3 in [34])**.** *Let $\Phi(x) = F(x) + r(x)$, where $F'(x)$ is L_F-Lipschitz continuous, and $F(x)$ and $r(x)$ are convex. For any $x \in \mathrm{dom}(r)$, and any $v \in \mathbf{R}^d$, define*

$$x^+ := \mathrm{Prox}_{\eta r(\cdot)}(x - \eta v),\ \mathcal{G} := \frac{1}{\eta}(x - x^+),\ \textit{and}\ \Delta := v - F'(x),$$

where η is a step size satisfying $0 < \eta \leq 1/L_F$. Then for any $y \in \mathbf{R}^d$,

$$\Phi(y) \geq \Phi(x^+) + \mathcal{G}^T(y - x) + \frac{\eta}{2}\|\mathcal{G}\|^2 + \Delta^T(x^+ - y).$$

D.1 Proof of Theorem 7

Proof. For the ease of notation, let us only focus in one run of Algorithm 1. And denote the input point as x_0^1 and the output point $\bar{x}$. If we set $x = x_i^t$, $y = x^*$, $v = \tilde{\nabla} F(x_i^t)$, $x^+ = x_{i+1}^t$ and $\mathcal{G} = \tilde{\mathcal{G}}(x_i^t)$, we get the following useful inequality,

$$\langle \tilde{\mathcal{G}}(x_i^t), x^* - x_i^t \rangle \leq \Phi(x^*) - \Phi(x_{i+1}^t) - \frac{\eta}{2}\|\tilde{\mathcal{G}}(x_i^t)\|^2 - \langle F'(x_i^t) - \tilde{\nabla} F(x_i^t), x^* - x_{i+1}^t \rangle.$$

As a result we have the following inequality,

$$\begin{aligned}
& \|x_{i+1}^t - x^*\|^2 \\
= \ & \|x_i^t - x^*\|^2 + \eta^2\|\tilde{\mathcal{G}}(x_i^t)\|^2 + 2\eta\langle \tilde{\mathcal{G}}(x_i^t), x^* - x_i^t \rangle \\
\leq \ & \|x_i^t - x^*\|^2 + \eta^2\|\tilde{\mathcal{G}}(x_i^t)\|^2 - 2\eta(\Phi(x_{i+1}^t) - \Phi(x^*)) - \eta^2\|\tilde{\mathcal{G}}(x_i^t)\|^2 \\
& -2\eta\langle F'(x_i^t) - \tilde{\nabla} F(x_i^t), x^* - x_{i+1}^t \rangle \\
\leq \ & \|x_i^t - x^*\|^2 - 2\eta(\Phi(x_{i+1}^t) - \Phi(x^*)) + \frac{2\eta}{\mu}\|F'(x_i^t) - \tilde{\nabla} F(x_i^t)\|^2 + \frac{\eta\mu}{2}\|x_{i+1}^t - x^*\|^2 \\
\leq \ & \|x_i^t - x^*\|^2 - \eta(\Phi(x_{i+1}^t) - \Phi(x^*)) + \frac{2\eta}{\mu}\|F'(x_i^t) - \tilde{\nabla} F(x_i^t)\|^2. \qquad (50)
\end{aligned}$$

Note that the inequality (50) is originally obtained in [35]. Adding $2\mu\cdot$(50) to (39), we get

$$\begin{aligned}
2\mu\eta\mathbf{E}[\Phi(x_{i+1}^t) - \Phi^*] \ \leq \ & \mathbf{E}[\Phi(x_i^t) + 2\mu\|x_i^t - x^*\|^2] - \mathbf{E}[\Phi(x_{i+1}^t) + 2\mu\|x_{i+1}^t - x^*\|^2] \\
& -(\frac{1}{2\eta} - \frac{L_F}{2})\mathbf{E}[\|x_{i+1}^t - x_i^t\|^2] + \frac{9}{2}\eta\mathbf{E}[\|\tilde{\nabla} F(x_i^t) - F'(x_i^t)\|^2]. \qquad (51)
\end{aligned}$$

By (51) and (33), we have

$$\begin{aligned}
2\mu\eta\sum_{i=0}^{\tau_t - 1}\mathbf{E}[\Phi(x_{i+1}^t) - \Phi^*] \ \leq \ & \mathbf{E}[\Phi(x_{\tau_t}^t) + 2\mu\|x_{\tau_t}^t - x^*\|^2] - \mathbf{E}[\Phi(x_0^t) + 2\mu\|x_0^t - x^*\|^2] \\
& -(\frac{1}{2\eta} - \frac{L_F}{2} - \tau_t\frac{9G_0\eta}{2S_t})\sum_{r=1}^{\tau_t}\mathbf{E}[\|x_r^t - x_{r-1}^t\|^2] + \tau_t\frac{9\sigma_0^2\eta}{2B_t}.
\end{aligned}$$

According to the selection of τ_t, S_t, B_t and η, we know that the coefficient $(\frac{1}{2\eta} - \frac{L_F}{2} - \tau_t \frac{9G_0\eta}{2S_t}) \geq 0$. Consequently,

$$2\mu\eta \sum_{i=0}^{\tau_t - 1} \mathbf{E}[\Phi(x_{i+1}^t) - \Phi^*] \quad \leq \quad \mathbf{E}[\Phi(x_{\tau_t}^t) + 2\mu\|x_{\tau_t}^t - x^*\|^2] - \mathbf{E}[\Phi(x_0^t) + 2\mu\|x_0^t - x^*\|^2] + \tau_t \frac{9\sigma_0^2\eta}{2B_t}.$$

Summing this up and apply the random selection rule of $\bar{x}$ gives

$$\begin{aligned}\mathbf{E}[\Phi(\bar{x}) - \Phi^*] \quad &\leq \quad \frac{1}{2\mu\eta\tau T}\mathbf{E}[\Phi(x_0^1) - \Phi^* + 2\mu\|x_0^1 - x^*\|^2] + \frac{9\sigma_0^2}{4\mu B_t} \\ &\leq \quad \frac{5}{2\mu\eta\tau T}\mathbf{E}[\Phi(x_0^1) - \Phi^*] + \frac{9\sigma_0^2}{4\mu B_t}\,.\end{aligned}$$

If we choose $T = \lceil \frac{5\sqrt{\epsilon}}{\mu\eta} \rceil$, $\tau = S = \frac{1}{\sqrt{\epsilon}}$ and $B_t = 1 + \frac{9\sigma_0^2}{2\mu\epsilon}$, then $\frac{5}{2\mu\eta\tau T} \leq \frac{1}{2}$ and we obtain

$$\mathbf{E}[\Phi(\bar{x}) - \Phi^*] \;\leq\; \frac{1}{2}\mathbf{E}[\Phi(x_0^1) - \Phi^*] + \frac{1}{2}\epsilon\,.$$

This proves the inequality (28). The rest of the proof will mimic that of Theorem 5. □

Discussions on sample complexity:

- If we choose $\tau = S = 1/\sqrt{\epsilon}$, $B_t = 1 + \frac{9\sigma_0^2}{2\mu\epsilon}$, and $T = \lceil \frac{5\sqrt{\epsilon}}{\mu\eta} \rceil$, then the sample complexity is

  $$T(B + 2\tau S)\ln\frac{1}{\epsilon} = \frac{5\sqrt{\epsilon}}{\mu\eta}\left(\frac{9\sigma_0^2}{2\mu\epsilon} + \frac{1}{\sqrt{\epsilon}}\frac{1}{\sqrt{\epsilon}}\right)\ln\frac{1}{\epsilon} = O\left((\mu^{-2}\sigma_0^2\epsilon^{-1/2} + \mu^{-1}\epsilon^{-1/2})\ln\epsilon^{-1}\right)\,.$$

  The above derivation needs to assume $\frac{5\sqrt{\epsilon}}{\mu\eta} \geq 1$ or at least $O(1)$, which means $\epsilon > (\eta\mu)^2$. If this condition is not satisfied, then we have $T = 1$ and the complexity is

  $$O\big((\mu^{-1}\sigma_0^2\epsilon^{-1} + \epsilon^{-1})\ln\epsilon^{-1}\big).$$

- If we choose $\tau = S = 1$, $B_t = 1 + \frac{9\sigma_0^2}{\mu\epsilon}$, and $T = \lceil \frac{5}{\mu\eta} \rceil$, the we also have

  $$\mathbf{E}[F(\bar{x}) - F^*] \leq \frac{1}{2}(F(x_0^1) - F^*) + \frac{1}{2}\epsilon,$$

  and the total sample complexity is

  $$T(B + 2\tau S)\ln\frac{1}{\epsilon} = \frac{5}{\mu\eta}\left(\frac{9\sigma_0^2}{\mu\epsilon} + 2\right)\ln\frac{1}{\epsilon} = O\left(\mu^{-2}\sigma_0^2\epsilon^{-1} + \mu^{-1}\right)\ln\epsilon^{-1}$$

  Defining the condition number $\kappa = L_F\nu = O(1/(\mu\eta))$, the above complexity becomes

  $$T(B + 2\tau S)\ln\frac{1}{\epsilon} = O\left(\kappa^2\sigma_0^2\epsilon^{-1} + \kappa\right)\ln\epsilon^{-1}$$

  Thus when $\sigma = 0$, we have $O\big(\kappa\ln\epsilon^{-1}\big)$ for deterministic optimization.

D.2 Proof of Theorem 8

The proof is very similar to the previous one. It actually becomes simpler by noticing that in the finite-sum case, the terms involving σ_0^2 disappear.

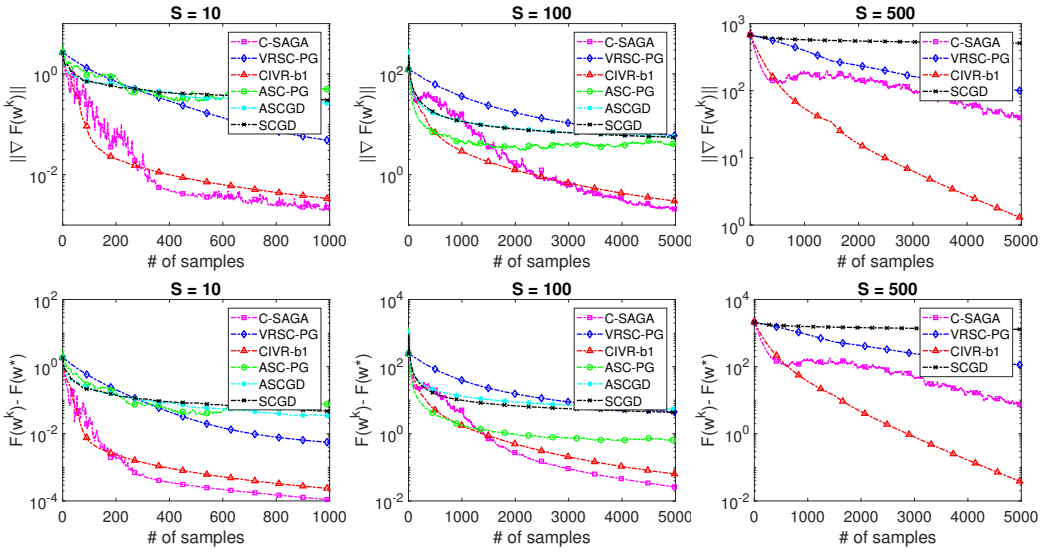

Figure 2: Experiments on policy evaluation for MDP for cases with $S = 10$, $S = 100$ and $S = 500$.

E Numerical experiments on policy evaluation for MDP

Here we provide additional numerical experiments on the policy evaluation problem for MDP.

Let $\mathcal{S} = \{1, ..., S\}$ be the state space of some Markov decision process. Suppose a reward of $R_{i,j}$ is received after transitioning from state i to state j. Let $P^\pi \in \mathbf{R}^{S\times S}$ be the transition probability matrix under some fixed policy π. Then the evaluation of the value function $V^\pi : \mathcal{S} \to \mathbf{R}$ under such policy is equivalent to solving the following Bellman equation:

$$V^\pi(i) = \sum_{j=1}^{S} P^\pi_{i,j}(R_{i,j} + \gamma V^\pi(j)) = \mathbf{E}_{j|i}[R_{i,j} + \gamma V^\pi(j)].$$

Following the suggestion of [5, 32], we apply the linear function approximation $V^\pi(i) \approx \langle \Psi_i, w^* \rangle$ for a given set of feature vectors Ψ_i. and would like to compute the optimal vector w^*. This can be formulated as the following problem

$$\underset{w}{\text{minimize}} \;\; F(w) \triangleq \sum_{i=1}^{S} \left(\langle \Psi_i, w \rangle - \sum_{j=1}^{S} P^\pi_{i,j}(R_{i,j} + \gamma \langle \Psi_j, w \rangle) \right)^2.$$

Let's denote

$$q^\pi_i(w) \triangleq \sum_{j=1}^{S} P^\pi_{i,j}(R_{i,j} + \gamma \langle \Psi_j, w \rangle) = \mathbf{E}_{j|i}[R_{i,j} + \gamma \langle \Psi_j, w \rangle].$$

Then by defining

$$g(w) = \left[\langle \Psi_1, w \rangle, ..., \langle \Psi_S, w \rangle, q^\pi_i(w), ..., q^\pi_S(w) \right]^T$$

and

$$f(y_1, ..., y_S, z_1, ..., z_S) = \|y - z\|^2 = \sum_{i=1}^{S} (y_i - z_i)^2,$$

the Least squares problem is transformed into the form of (2).

For this problem, we test the SCGD [31], the ASCGD [31], the ASC-PG [32], the VRSC-PG [11], C-SAGA [35] and our CIVR algorithms. In Section 5, we already tested the algorithms under their standard batch sizes, e.g. $\lceil n^{2/3} \rceil$ and $\lceil \sqrt{n} \rceil$. However, small constant batch sizes are often preferred in practice. Therefore, we would like to set the batch size to $s = 1$ for all algorithms. For this special case, we denote the CIVR as the CIVR-b1. To balance the sample complexity between the initial full batch sampling and the later subsampling with $s = 1$, we set the epoch length for VRSC-PG and CIVR-b1 to be S.

Note that the last S components of g are all independent expectations, therefore the variance reduction technique of VRSC-PG [11], C-SAGA [35] and CIVR-b1 applied to each of these components. In the experiments, P^π, Φ and R^π are generated randomly.

Similar to the experiments performed in Section 5, the step sizes are chosen from $\{0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$ by experiments for VRSC-PG, C-SAGA as well as for CIVR-b1. For $S = 10$, $\eta = 0.1$ works best for both C-SAGA and CIVR-b1, while $\eta = 0.01$ works best for VRSC-PG; For $S = 100$, $\eta = 0.001$ works best for both C-SAGA and CIVR-b1, while $\eta = 0.0001$ works best for VRSC-PG. For $S = 500$, $\eta = 0.0001$ works best for all three of them.

When $S = 10$ and $S = 100$, we choose $\alpha_k = 0.01k^{-3/4}$ and $\beta_k = 0.1k^{-1/2}$ for SCGD, $\alpha_k = 0.01k^{-5/7}$ and $\beta_k = 0.1k^{-4/7}$ for ASCGD and $\alpha_k = 0.01k^{-1/2}$ and $\beta_k = 0.1k^{-1}$ for ASC-PG. When $S = 500$, we choose $\alpha_k = 0.0001k^{-3/4}$ and $\beta_k = 0.001k^{-1/2}$ for SCGD while ASCGD and ASC-PG fail to converge under various trials of parameters. The meaning of these step size parameters can be found in [32] and [31].

Figure 2 shows three experiments with sizes $S = 10$, $S = 100$ and $S = 500$ respectively. We can see that both C-SAGA and CIVR-b1 preform much better than other algorithms in our setting. CIVR-b1 has more smooth and stable trajectory than C-SAGA.