IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Variance-reduced Shuffling Gradient Descent with Momentum for Finite-sum Minimization

Xia Jiang, Xianlin Zeng, *Member, IEEE,* Lihua Xie, *Fellow, IEEE,* and Jian Sun, *Member, IEEE*

***Abstract*— Finite-sum minimization is a fundamental optimization problem in signal processing and machine learning. This paper proposes a variance-reduced shuffling gradient descent with Nesterov's momentum for smooth convex finite-sum optimization. We integrate an explicit variance reduction into the shuffling gradient descent to deal with the variance introduced by shuffling gradients. The proposed algorithm with a unified shuffling scheme converges at a rate of $\mathcal{O}(\frac{1}{T})$, where T is the number of epochs. The convergence rate independent of gradient variance is better than most existing shuffling gradient algorithms for convex optimization. Finally, numerical simulations demonstrate the convergence performance of the proposed algorithm.**

I. Introduction

Finite-sum minimization is a fundamental problem with many practical applications in signal processing and machine learning [1], [2]. With the rapid growth of data in recent years, the deterministic gradient descent methods based on full gradients have become inefficient in solving finite-sum optimization problems. Therefore, various first-order stochastic methods are leading algorithms for finite-sum minimization due to their scalability and low computational requirements [3]–[5]. Stochastic gradient descent (SGD) is a well-known first-order algorithm where the actual full gradient is replaced by a gradient estimate calculated from randomly sampled data. SGD owns conditionally unbiased gradients by uniformly independent sampling and achieves the convergence rate of $\mathcal{O}(1/\sqrt{T})$ [6], [7], where T is the number of epochs. Inspired by the Nesterov's momentum technique, researchers have made efforts to integrate the momentum step into the stochastic gradient descent. With the strong growth condition that the variance converges to zero, some existing works [8], [9] proved a faster convergence rate of SGD with Nesterov's momentum. However, the vanishing variance assumption is necessary for these works to develop a better convergence rate of SGD with momentum than the traditional SGD.

This work was partly supported by the National Natural Science Foundation of China under Grants 61925303, 62088101, 62073035, 62173034, the Natural Science Foundation of Chongqing under Grant 2021ZX4100027. *(Corresponding author: Xianlin Zeng.)*

X. Jiang and J. Sun are with the National Key Lab of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing 10081, China, and also with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401120, China (e-mail: jiang-xia@bit.edu.cn; sunjian@bit.edu.cn).

X. Zeng is with the National Key Lab of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing 10081, China (e-mail: xianlin.zeng@bit.edu.cn).

L. Xie is with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (e-mail: elhxie@ntu.edu.sg).

Shuffling gradient descent is a popular but elusive stochastic algorithm for finite-sum minimization problems, where the components are sampled without replacement, differently from the traditional sampling-with-replacement SGD. Shuffling gradient algorithms with deterministic or random shuffling samples often perform better than SGD in many practical problems [10], [11]. Therefore, the theoretical research of shuffling gradient descent has recently attracted much attention. The main difficulty in analyzing shuffling gradient methods is that the sampling without replacement makes the conditional gradients biased. Facing this challenge, some inspiring works [12], [13] have provided some involved yet insightful proofs for the $\mathcal{O}(1/T^{2/3})$ convergence rate of shuffling gradient descent algorithms under weak assumptions. To further improve the convergence rate of shuffling gradient algorithms, the recent work [14] integrated the Nesterov's momentum step into shuffling gradient descent and obtained an $\mathcal{O}(1/T)$ convergence rate, in which there still exists the effect of gradient variance.

More recently, the variance reduction technique has become another important way to improve the convergence rate of first-order stochastic algorithms. Many works have shown that SGD can converge much faster if one makes a better choice of the stochastic gradient so that its variance reduces as the iteration increases [15], [16]. Many variance reduction techniques have been proposed for strongly convex optimization, such as SAGA [17], SVRG [18], and SARAH [19]. Furthermore, [20] has incorporated Nesterov's momentum trick into a variance-reduction-based algorithm and sped it up. However, all these variance reduction works are based on the SGD algorithms. Limited works such as [21], [22] have applied variance reduction techniques to the shuffling gradient descent algorithms.

Motivated by the popular practical applications of shuffling algorithms, we studied a new shuffling gradient descent algorithm with variance reduction and momentum steps for finite-sum convex optimization. The contributions of this paper are summarized as follows.

- This paper develops a variance-reduced shuffling gradient descent with momentum to obtain a solution for smooth convex optimization. We relax the vanishing gradient variance assumption in most existing SGD algorithms with Nesterov's momentum for convex optimization [8], [23]. In addition, we integrate an explicit variance reduction step into the shuffling gradient algorithm to eliminate the effect of gradient variance and obtain a better convergence rate.
- We provide a rigorous and complete proof for the

proposed variance-reduced shuffling gradient algorithm with momentum. We first provide an equivalent reformulation of the original finite-sum minimization problem. Then, we establish a concise analysis using some proper auxiliary variables and the backward per-iteration deviation at each epoch.

- For the unified shuffling scheme (either deterministic or random shuffling), the proposed algorithm achieves an improved convergence rate of $\mathcal{O}(\frac{1}{T})$ in terms of the number of epochs, better than the $\mathcal{O}(\frac{1}{T^{2/3}})$ convergence rate of shuffling gradient algorithms without momentum [24]. Compared with the accelerated shuffling work [14], the proposed algorithm achieves a faster convergence rate independent of gradient variance, at the cost of computing the full gradient once at each epoch.

The remainder of the paper is organized as follows. The problem description and variance-reduced shuffling gradient algorithm are developed in section II. The convergence analysis of the proposed algorithm is provided in section III. The efficiency of the proposed algorithm is verified by simulations in Section IV and the conclusion is made in section V.

Mathematical notations: We denote $\mathbb{R}$ as the set of real numbers, $\mathbb{R}^n$ as the set of n-dimensional real column vectors, $\mathbb{R}^{n\times m}$ as the set of n-by-m real matrices, and $\mathbb{N}_+$ as the set of positive integers. All vectors in the paper are column vectors, unless otherwise noted. The notation $\mathbf{1}$ denotes an all-1 vector with the corresponding dimension. The notation $[n]$ denotes the set $\{0,\cdots,n-1\}$. For a real vector v, $\|v\|$ is the Euclidean norm. For a differentiable function $f(x)$, its gradient vector is represented by $\nabla f(x)$.

II. Problem Description and Algorithm Design

In this paper, we study the following finite-sum convex optimization problem

$$\min_{x\in\mathbb{R}^d} f(x),\ f(x)=\frac{1}{n}\sum_{i\in[n]} f_i(x), \tag{1}$$

where $x\in\mathbb{R}^d$ is the variable to be determined and f_i for $i\in[n]$ is a scalar function. This standard finite-sum problem arises in many signal processing and machine learning tasks [25]–[27], where the number of components n is large such that deterministic algorithms relying on full gradients are usually inefficient for this problem.

We design a new stochastic gradient descent using the unified shuffling scheme (either deterministic or random shuffling). With an explicit variance reduction and Nesterov's momentum, we propose a variance-reduced shuffling gradient algorithm with momentum in Algorithm 1 to reduce the effect of biased gradients and obtain a faster convergence rate.

In the proposed algorithm, an epoch is one complete pass through all training data. At the beginning of each epoch, the data samples are shuffled to obtain a new random or deterministic permutation of the index set $[n]$. Then, consecutive gradient descents are executed with the shuffled permutation in the inner iterations. The step-size η_k/n in each gradient descent satisfies $0<\eta_k<1/L$, and the proposed algorithm uses a gradient estimator g_k^i instead of the stochastic gradient ∇f_i to reduce the gradient variance. The gradient estimator takes an explicit variance reduction, first developed in [18] for strongly-convex optimization. At the end of each epoch, we adopt Nesterov's momentum step using the final variable of inner iterations.

Algorithm 1 Variance-reduced shuffling gradient algorithm with momentum (VRSGM)

Initialization: Set $y_0=x_0$, the number of epochs T.
for $k=1,\cdots,T$ **do**
  Generate any permutation $\pi_k=(\pi_k^0,\cdots,\pi_k^{n-1})$ of $[n]$ (either deterministic or random)
  $y_k^0=y_{k-1}$
  for $i=0,\cdots,n-1$ **do**
    $g_k^i=\nabla f_{\pi_k^i}(y_k^i)-\nabla f_{\pi_k^i}(y_{k-1})+\nabla f(y_{k-1})$
    $y_k^{i+1}=y_k^i-\frac{\eta_k}{n}g_k^i$
  end for
  $x_k=y_k^n$
  $y_k=x_k+\frac{k-1}{k+2}(x_k-x_{k-1})$
end for

Remark 2.1: Compared with the existing Nesterov accelerated SGD algorithms [8], [9], the proposed algorithm adopts an explicit variance reduction step to remove the vanishing variance assumption, which is generally not satisfied in practice. In addition, the introduced variance reduction step eliminates the effect of gradient variance on the convergence rate of the algorithm, different from the accelerated shuffling gradient work [14].

Remark 2.2: Different from the unbiased gradient estimator in SGD, the gradient in the shuffling gradient algorithm is biased, so the intuition of applying the momentum step in each inner iteration may not be superior due to the possible error accumulation [14].

III. Theoretical analysis

This section provides the convergence analysis for Algorithm 1 under the following basic assumptions of objective function.

Assumption 3.1: Each objective function f_i is convex and L-smooth, i.e., $f_i(x)+\langle\nabla f_i(x),y-x\rangle\le f_i(y)$ and $\|\nabla f_i(x)-\nabla f_i(y)\|\le L\|x-y\|$ for all $x,y\in\mathbb{R}^d$ and $i\in[n]$.

The L-smoothness of function f_i also implies that the function f_i satisfies

$$f_i(x)\le f_i(y)+\langle\nabla f_i(y),x-y\rangle+\frac{L}{2}\|x-y\|^2,\ i\in[n].$$

For concise proof, we define $v_k\triangleq\frac{k+1}{2}x_k-\frac{k-1}{2}x_{k-1}$ and *the backward per-iteration deviation* of VRSGM at epoch k as $B_k=\frac{1}{n}\sum_{i\in[n]}\|y_k^i-y_k^0\|^2$ for all $k\in\mathbb{N}_+$. Define x_* as a solution to Problem (1) and $f^*\triangleq f(x_*)$.

At first, we estimate an upper bound of the optimality gap $f(x_T)-f^*$ for $T\in\mathbb{N}_+$.

Lemma 3.1: Suppose Assumption 3.1 holds. Let $0 < \eta_k \leq \frac{1}{L}$, $i \in [n]$ and $\{\epsilon_k\}_{k=1}^T$ be a positive sequence. Then, we have

$$\begin{aligned} &T(T+2)[f(x_T) - f^*] \\ \leq &\sum_{k=1}^{T} \frac{L^2\eta_k(k+1)^2}{2\epsilon_k} B_k - \sum_{k=1}^{T}[f(x_k) - f^*] \\ &+ \sum_{k=1}^{T} \frac{2}{\eta_k}\|v_{k-1} - x_*\|^2 - \sum_{k=1}^{T} \frac{2}{\eta_k}(1-\epsilon_k)\|v_k - x_*\|^2. \end{aligned} \tag{2}$$

Proof: Let $a_{i,k-1} \triangleq -\nabla f_{\pi_k^i}(y_{k-1}) + \nabla f(y_{k-1})$. The gradient estimate in Line 6 of VRSGM can be written as

$$g_k^i = \nabla f_{\pi_k^i}(y_k^i) + a_{i,k-1}. \tag{3}$$

Clearly, $\sum_{i\in[n]} a_{i,k-1} = \sum_{i\in[n]} \big(- \nabla f_{\pi_k^i}(y_{k-1}) + \nabla f(y_{k-1})\big) = 0$. Adding this zero to $f(\cdot)$ in problem (1) yields that

$$f(x_k) = \frac{1}{n}\sum_{i\in[n]} (f_i(x_k) + \langle a_{i,k-1}, x_k\rangle) = \frac{1}{n}\sum_{i\in[n]} \tilde{f}_i(x_k), \tag{4}$$

where $\tilde{f}_i(x_k) \triangleq f_i(x_k) + \langle a_{i,k-1}, x_k\rangle$ for all $i \in [n]$. Note that $\nabla\tilde{f}_i(x_k) = \nabla f_i(x_k) + a_{i,k-1}$ for all $i \in [n]$. It follows from (3) that

$$g_k^i = \nabla\tilde{f}_{\pi_k^i}(y_k^i), \quad \forall i \in [n]. \tag{5}$$

Following from the fact that $y_k^{i+1} = y_k^i - \frac{\eta_k}{n}\nabla\tilde{f}_{\pi_k^i}(y_k^i)$, we have $y_k^i = y_k^0 - \sum_{j=0}^{i-1}\eta_k^j\nabla\tilde{f}_{\pi_k^j}(y_k^j)$. Note that $y_k^0 = y_{k-1}$ and $y_k^n = x_k$. We obtain

$$\begin{aligned} x_k &= y_{k-1} - \frac{\eta_k}{n}\sum_{j\in[n]} \nabla\tilde{f}_{\pi_k^j}(y_k^j) \\ &= y_{k-1} - \frac{\eta_k}{n}\sum_{j\in[n]} \nabla f_{\pi_k^j}(y_k^j). \end{aligned} \tag{6}$$

Following from the definition of $\nabla\tilde{f}_i(x_k)$ and $\sum_{i\in[n]} a_{i,k-1} = 0$, we have the fact that $\sum_{i\in[n]}\nabla f_i(x_i) = \sum_{i\in[n]}\nabla\tilde{f}_i(x_i)$ holds for any given argument x_i. From this fact, the last equality of (6) holds. In addition, because our problem assumptions are the same as those of [14] and the variable updating (6) is the same as the variable updating in the proof of Lemma 1 of [14], we can follow similar derivations of Lemma 1 of [14] to obtain the desired result. ∎

For convenience, we define $v_0 = x_0$ and $\theta_k = \frac{2}{k+2} \in (0,1)$ for $k \geq 1$ with $\theta_0 = 1$. Recall that $v_k \triangleq \frac{k+1}{2}x_k - \frac{k-1}{2}x_{k-1}$. By the y_k updating in line 10 of Algorithm 1, we have that y_k is a convex combination of v_k and x_k,

$$\begin{aligned} y_k =& x_k + \frac{k-1}{k+2}(x_k - x_{k-1}) \\ =& \frac{2}{k+2}\Big(\frac{k+1}{2}x_k - \frac{k-1}{2}x_{k-1}\Big) + \Big(1 - \frac{2}{k+2}\Big)x_k \\ =& \theta_k v_k + (1-\theta_k)x_k. \end{aligned} \tag{7}$$

Using the above quantities and (7), we establish an upper bound of B_k in the following lemma.

Lemma 3.2: Suppose Assumption 3.1 holds. Let $\eta_k \leq \frac{1}{2L}$ and $k \in \mathbb{N}_+$. Then, for all $k \in \mathbb{N}_+$, B_k satisfies

$$B_k \leq 8\eta_k^2 L\Big(\frac{k-1}{k+1}(f(x_{k-1}) - f^*) + \frac{2}{k+1}(f(v_{k-1}) - f^*)\Big).$$

Proof: By the proposed algorithm and the L-smoothness of objective functions,

$$\begin{aligned} &\|y_k^i - y_k^0\|^2 \\ =&\frac{\eta_k^2}{n^2}\Big\|\sum_{j=0}^{i-1}\nabla f_{\pi_k^j}(y_k^j) - \sum_{j=0}^{i-1}\nabla f_{\pi_k^j}(y_{k-1}) + \sum_{j=0}^{i-1}\nabla f(y_{k-1})\Big\|^2 \\ \leq&\frac{2i\eta_k^2L^2}{n^2}\sum_{j=0}^{i-1}\|y_k^j - y_{k-1}\|^2 + \frac{2\eta_k^2 i}{n^2}\sum_{j=0}^{i-1}\Big\|\nabla f(y_{k-1})\Big\|^2 \\ \leq&\frac{2i\eta_k^2L^2}{n^2}\sum_{j=0}^{i-1}\|y_k^j - y_k^0\|^2 + \frac{2\eta_k^2}{n}\sum_{j=0}^{i-1}\Big\|\nabla f(y_{k-1})\Big\|^2 \\ \leq&\frac{2i\eta_k^2L^2}{n^2}\sum_{j=0}^{n-1}\|y_k^j - y_k^0\|^2 + \frac{2\eta_k^2}{n}\sum_{j=0}^{n-1}\Big\|\nabla f(y_{k-1})\Big\|^2 \\ =&\frac{2i\eta_k^2L^2}{n}B_k + \frac{2\eta_k^2}{n}\sum_{j=0}^{n-1}\Big\|\nabla f(y_{k-1})\Big\|^2, \end{aligned}$$

where the first inequality holds due to the AM-QM inequality (i.e. $\|\sum_{j=1}^n x_j\|^2 \leq n\sum_{j=1}^n\|x_j\|^2$, $\forall x_j \in \mathbb{R}^d$) and L-smoothness of f_i, and $B_k = \frac{1}{n}\sum_{i=0}^{n-1}\|y_k^i - y_k^0\|^2$. Summing up the previous expression from $i = 0$ to $i = n-1$ yields

$$\begin{aligned} nB_k =& \sum_{i=0}^{n-1}\|y_k^i - y_k^0\|^2 \\ \leq&\sum_{i=0}^{n-1}\frac{2i\eta_k^2L^2}{n}B_k + \sum_{i=0}^{n-1}\frac{2\eta_k^2}{n}\sum_{j=0}^{n-1}\Big\|\nabla f(y_{k-1})\Big\|^2 \\ \leq&\frac{2\eta_k^2L^2}{n}\frac{n^2+n}{2}B_k + \sum_{i=0}^{n-1}\frac{2\eta_k^2}{n}\sum_{j=0}^{n-1}\Big\|\nabla f(y_{k-1})\Big\|^2 \\ \leq&2\eta_k^2L^2nB_k + \sum_{i=0}^{n-1}\frac{2\eta_k^2}{n}\sum_{j=0}^{n-1}\Big\|\nabla f(y_{k-1})\Big\|^2 \\ \leq&\frac{1}{2}nB_k + \sum_{i=0}^{n-1}\frac{2\eta_k^2}{n}\sum_{j=0}^{n-1}\Big\|\nabla f(y_{k-1})\Big\|^2, \end{aligned}$$

where the third inequality holds due to $\frac{n^2+n}{2} \leq n^2$ and the fourth inequality holds due to $\eta_k \leq \frac{1}{2L}$ and $\eta_k^2L^2 \leq \frac{1}{4}$.

By subtracting $\frac{nB_k}{2}$ and multiplying $\frac{2}{n}$ from the both sides of the above inequality, we have

$$B_k \leq 4\eta_k^2\|\nabla f(y_{k-1})\|^2. \tag{8}$$

Because of the convexity and L-smoothness of f, $\|\nabla f(y_{k-1})\|^2 \leq 2L(f(y_{k-1}) - f^*)$. Substituting this inequality into (8) gives

$$B_k \leq 8\eta_k^2L(f(y_{k-1}) - f^*). \tag{9}$$

Due to the fact that $y_k = \theta_k v_k + (1-\theta_k)x_k$,

$$\begin{aligned} &f(y_k) - f^* \\ =&f(\theta_k v_k + (1-\theta_k)x_k) - f^* \\ \le&\theta_k f(v_k) + (1-\theta_k)f(x_k) - f^* \\ =&\theta_k(f(v_k) - f^*) + (1-\theta_k)\big(f(x_k) - f^*\big). \end{aligned} \tag{10}$$

Then, following from (9) and (10), B_k satisfies

$$\begin{aligned} B_k \le&8\eta_k^2 L(f(y_{k-1}) - f^*) \\ \le&8\eta_k^2 L\Big((1-\theta_{k-1})(f(x_{k-1}) - f^*) + \theta_{k-1}\big(f(v_{k-1}) - f^*\big)\Big), \end{aligned}$$

where $\theta_{k-1} = \frac{2}{k+1}$ and $1-\theta_{k-1} = \frac{k-1}{k+1}$. ∎

Now, we are ready to prove the convergence rate of Algorithm 1 using properties of the optimality gap in Lemma 3.1 and B_k in Lemma 3.2.

Theorem 3.1: Suppose Assumption 3.1 holds. Let $\eta_k = \frac{h\alpha^k}{L} < \frac{1}{2L}$, $\alpha = 1 + \frac{1}{T}$ and $h = \frac{4}{5\sqrt{e^3}(T+1)} > 0$ for $k, T \in \mathbb{N}_+, T \ge 2$ and $k \le T$. Then,

$$f(x_T) - f^* = \mathcal{O}\Big(\frac{L\|x_0 - x_*\|^2}{T}\Big), \quad T \ge 2. \tag{11}$$

Proof:

By Lemma 3.1, we have

$$\begin{aligned} &T(T+2)(f(x_T) - f^*) \\ \le&\sum_{k=1}^{T} \frac{L^2\eta_k(k+1)^2}{2\epsilon_k} B_k - \sum_{k=1}^{T}(f(x_k) - f^*) \\ &+ \sum_{k=1}^{T}\frac{2}{\eta_k}\|v_{k-1} - x_*\|^2 - \sum_{k=1}^{T}\frac{2}{\eta_k}(1-\epsilon_k)\|v_k - x_*\|^2 \\ \le&\sum_{k=1}^{T}\frac{4L^3\eta_k^3(k+1)^2}{\epsilon_k}\Big[\frac{k-1}{k+1}\big(f(x_{k-1}) - f^*\big) \\ &+ \frac{2}{k+1}\big(f(v_{k-1}) - f^*\big)\Big] - \sum_{k=1}^{T}\big(f(x_k) - f^*\big) \\ &+ \sum_{k=1}^{T}\frac{2}{\eta_k}\|v_{k-1} - x_*\|^2 - \sum_{k=1}^{T}\frac{2}{\eta_k}(1-\epsilon_k)\|v_k - x_*\|^2 \\ \le&\sum_{k=1}^{T}\frac{4L^3\eta_k^3(k+1)^2}{\epsilon_k}\big(f(x_{k-1}) - f^* + f(v_{k-1}) - f^*\big) \\ &- \sum_{k=1}^{T}(f(x_{k-1}) - f^*) + \sum_{k=1}^{T}(f(x_{k-1}) - f(x_k)) \\ &+ \sum_{k=1}^{T}\frac{2}{\eta_k}\|v_{k-1} - x_*\|^2 - \sum_{k=1}^{T}\frac{2}{\eta_k}(1-\epsilon_k)\|v_k - x_*\|^2, \end{aligned} \tag{12}$$

where the second inequality holds by Lemma 3.2, and the third inequality holds due to $\frac{2}{k+1} \le 1$ and $\frac{k-1}{k+1} \le 1$.

Without loss of generality, we define $\epsilon_k \triangleq 1 - \frac{1}{\alpha} - \frac{he}{4} > 0$. Then, the coefficient of the first term in (12) satisfies

$$\begin{aligned} \frac{4L^3\eta_k^3(k+1)^2}{\epsilon_k} &= \frac{4h^3\alpha^{3k}(k+1)^2}{\epsilon_k} \\ &\le 4h^3\alpha^{3k}(T+1)^2\epsilon_k^{-1} \\ &\le 16h^3e^3(T+1)^2\frac{\alpha}{4\alpha - 4 - \alpha he}. \end{aligned} \tag{13}$$

It follows from $\alpha = 1 + \frac{1}{T} \le \frac{3}{2}$ that $h\big(16h^2e^3(T+1)^2\alpha + \alpha e\big) \le h\big(24h^2e^3(T+1)^2 + \frac{3e}{2}\big)$. Recall that $h^2 = \frac{16}{25e^3(T+1)^2}$. Then, we have

$$\begin{aligned} &h\big(16h^2e^3(T+1)^2\alpha + \alpha e\big) \\ &\quad\le \frac{4}{5\sqrt{e^3}(T+1)}\Big(\frac{24e^3(T+1)^2 \times 16}{25e^3(T+1)^2} + \frac{3e}{2}\Big) \\ &\quad< \frac{4}{5\sqrt{e^3}T}(16 + \frac{3e}{2}). \end{aligned}$$

Because $16 + \frac{3e}{2} < 5\sqrt{e^3}$, it follows that $16h^3e^3(T+1)^2\alpha < 4\alpha - 4 - \alpha he$. Then, it follows from (13) that

$$\frac{4L^3\eta_k^3(k+1)^2}{\epsilon_k} < 1, \ \forall k \in [T]. \tag{14}$$

With the inequalities (12) and (14), we have

$$\begin{aligned} &T(T+2)(f(x_T) - f^*) \\ \le&\sum_{k=1}^{T}\Big[\frac{4L^3\eta_k^3(k+1)^2}{\epsilon_k} - 1\Big](f(x_{k-1}) - f^*) \\ &+ f(x_0) - f(x_T) + \sum_{k=1}^{T}\big(f(v_{k-1}) - f^*\big) \\ &+ \sum_{k=1}^{T}\frac{2}{\eta_k}\|v_{k-1} - x_*\|^2 - \sum_{k=1}^{T}\frac{2}{\eta_k}(1-\epsilon_k)\|v_k - x_*\|^2 \\ \le&f(x_0) - f(x_T) + \sum_{k=1}^{T}\Big[f(v_{k-1}) - f(v_k)\Big] + \sum_{k=1}^{T}\frac{L}{2}\|v_k - x_*\|^2 \\ &+ \sum_{k=1}^{T}\frac{2}{\eta_k}\|v_{k-1} - x_*\|^2 - \sum_{k=1}^{T}\frac{2}{\eta_k}(1-\epsilon_k)\|v_k - x_*\|^2 \\ \le&\frac{L}{2}\|x_0 - x_*\|^2 + f(v_0) - f^* + \sum_{k=1}^{T}\frac{L}{2}\|v_k - x_*\|^2 \\ &+ \sum_{k=1}^{T}\frac{2}{\eta_k}\|v_{k-1} - x_*\|^2 - \sum_{k=1}^{T}\frac{2}{\eta_k}(1-\epsilon_k)\|v_k - x_*\|^2 \\ =&\frac{L}{2}\|x_0 - x_*\|^2 + f(v_0) - f^* + \sum_{k=1}^{T}\frac{2L}{h\alpha^k}\|v_{k-1} - x_*\|^2 \\ &- \sum_{k=1}^{T}\Big[\frac{2L}{h\alpha^k}(1-\epsilon_k) - \frac{L}{2}\Big]\|v_k - x_*\|^2, \end{aligned} \tag{15}$$

where the second inequality holds due to $f(v_k) - f^* \le \frac{L}{2}\|v_k - x_*\|^2$, the third inequality holds due to $f(x_0) - f(x_T) \le f(x_0) - f^* \le \frac{L}{2}\|x_0 - x_*\|^2$ and $f(v_0) - f(v_T) \le f(v_0) - f^*$.

Clearly, $\alpha^k \le \alpha^T = (1 + \frac{1}{T})^T \le e$. Hence, $\epsilon_k = 1 - \frac{1}{\alpha} - \frac{he}{4} \le 1 - \frac{1}{\alpha} - \frac{h\alpha^k}{4}$ and $1 - \epsilon_k \ge \frac{1}{\alpha} + \frac{h\alpha^k}{4} = (\frac{2}{h\alpha^{k+1}} + \frac{1}{2})\frac{h\alpha^k}{2}$.

It follows that

$$\frac{2L}{h\alpha^k}(1-\epsilon_k)-\frac{L}{2}\geq\frac{2L}{h\alpha^{k+1}}, \tag{16}$$

and, recalling (15), we have,

$$\begin{aligned}
&T(T+2)(f(x_T)-f^*)\\
\leq&\frac{L}{2}\|x_0-x_*\|^2+f(v_0)-f^*+\sum_{k=1}^{T}\frac{2L}{h\alpha^k}\|v_{k-1}-x_*\|^2\\
&-\sum_{k=1}^{T}\frac{2L}{h\alpha^{k+1}}\|v_k-x_*\|^2\\
\leq&\frac{L}{2}\|x_0-x_*\|^2+f(v_0)-f^*+\frac{2L}{h\alpha}\|v_0-x_*\|^2\\
\leq&\frac{L}{2}\|x_0-x_*\|^2+\Big(\frac{L}{2}+\frac{2L}{h\alpha}\Big)\|v_0-x_*\|^2\\
=&\frac{L}{2}\|x_0-x_*\|^2+\frac{(h\alpha+4)L}{2h\alpha}\|v_0-x_*\|^2,
\end{aligned} \tag{17}$$

where the last inequality holds due to $f(v_0)-f^*\leq\frac{L}{2}\|v_0-x_*\|^2$.

Dividing the both sides of (17) by $T(T+2)$ yields

$$\begin{aligned}
&f(x_T)-f^*\\
\leq&\frac{L}{2T(T+2)}\|x_0-x_*\|^2+\frac{(h\alpha+4)L}{2h\alpha T(T+2)}\|v_0-x_*\|^2\\
=&\frac{L}{2T(T+2)}\|x_0-x_*\|^2+\frac{(1+4/h\alpha)L}{2T(T+2)}\|x_0-x_*\|^2\\
=&\frac{2L+5\sqrt{e^3}TL}{2T(T+2)}\|x_0-x_*\|^2\\
=&\mathcal{O}\Big(\frac{L\|x_0-x_*\|^2}{T}\Big),
\end{aligned} \tag{18}$$

where the first and second equalities hold due to $x_0=v_0$ and $\frac{4}{h\alpha}=5\sqrt{e^3}T$, respectively. ■

Remark 3.1: Although the step-size $\frac{\eta_k}{n}$ is small for large n, the convergence rate in (18) is independent of n due to the n consecutive gradient descents in the inner iterations of the proposed algorithm.

Remark 3.2: With the help of variance reduction and momentum step, the proposed algorithm has a faster convergence rate than the $\mathcal{O}(\frac{1}{T^{2/3}})$ rate of traditional shuffling gradient algorithm [24]. The accelerated shuffling gradient work [14] shows that random shuffling has a faster decay rate of gradient variance than deterministic shuffling. In contrast, the proposed shuffling gradient algorithm possesses the same convergence rate in both shuffling settings, because the proposed algorithm achieves a convergence result without the effect of gradient variance. To be specific, because it is not affected by the gradient variance σ, the proposed algorithm has a better convergence rate of $\mathcal{O}(\frac{L\|x_0-x_*\|^2}{T})$ than the accelerated shuffling gradient in [14], which owns a convergence rate of $\mathcal{O}(\frac{\sigma^2/L+L\|x_0-x_*\|^2}{T})$. In addition, due to using the momentum design, the proposed algorithm has a convergence rate of the last iterate $f(x_T)$, which is better in practice than that of the average iterate $f(\tilde{x}_T)$, where $\tilde{x}_T=\frac{1}{T}\sum_{k=1}^{T}x_k$, in the previous work [21].

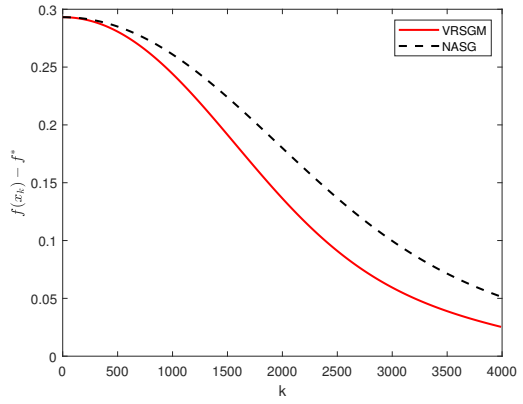

Fig. 1. The trajectories of $f(x_k)-f^*$ over a9a dataset.

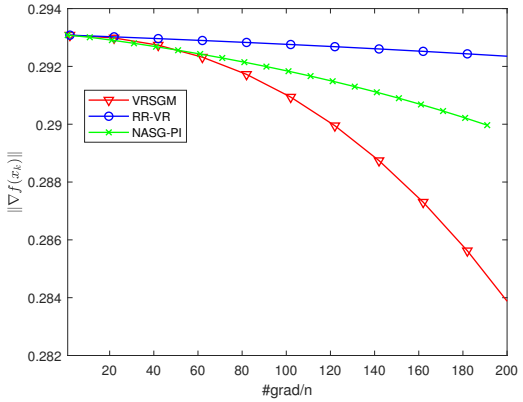

Fig. 2. The norm of gradients generated by different algorithms.

Remark 3.3: To discuss the trade-off between variance reduction and the computational cost clearly, we provide the gradient computation complexity of the proposed algorithm, i.e., the gradient computational cost required to achieve an ϵ-accurate solution. The gradient computation complexity of the proposed algorithm is $\mathcal{O}(n\epsilon^{-1})$, which is the same as that of the algorithm in [14]. We assign a computational cost c for evaluating ∇f_i to compare the specific number of gradient evaluations. For the proposed VRSGM algorithm, the computational cost is $\mathcal{O}(2ncL\|x_0-x_*\|^2\epsilon^{-1})$. For the algorithm in [14], the computational cost is $\mathcal{O}(nc(\sigma_*^2/L+L\|x_0-x_*\|^2)\epsilon^{-1})$, where $\sigma_*^2\triangleq\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x_*)\|^2$ is the gradient variance. By comparison, we obtain that if the gradient variance σ_*^2 satisfies $\sigma_*^2\geq L^2\|x_0-x_*\|^2$, the proposed VRSGM has a lower computational cost.

IV. Simulation

In this section, we apply the proposed VRSGM method to the logistic regression problem, which has the form (1) with

$$f(x)=\frac{1}{n}\sum_{i=1}^{n}\ln\Big(1+\exp\big(-l_i\langle a_i,x\rangle\big)\Big), \tag{19}$$

where $\{(a_i,l_i)\}_{i=1}^n$ is a set of samples, $a_i\in\mathbb{R}^d$ is the feature vector of the ith sample, $l_i\in\{-1,1\}$ is the classification

value of the ith sample. For comparison, we also apply some other stochastic algorithms, i.e. the random reshuffling with variance reduction (RR-VR) in [21] and Nesterov accelerated shuffling gradient descent (NASG) and NASG that applies Nesterov's momentum in each inner iteration (NASG-PI) in [14], to solve (19). We conduct the numerical experiments over the public a9a dataset. We apply the random reshuffling scheme to all stochastic algorithms and take the same initialization value.

We show the trajectories of $f(x_k) - f^*$ in terms of epoch k in Fig. 1. The convergent trajectory of VRSGM demonstrates that the proposed VRSGM algorithm has a sublinear convergence rate, verifying the theoretical analysis in Theorem 3.1. In addition, the proposed VRSGM converges faster than NASG in terms of epoch k, which also confirms the discussion about the convergence rates of these two algorithms in Remark 3.2.

We show the convergence results in terms of gradient evaluations of different comparative algorithms in Fig. 2. Since VRSGM takes twice as many gradient evaluations per epoch as NASG, NASG has better convergence performance. For simplicity, we omit the numerical trajectory of NASG algorithm. Fig. 2 indicates that the proposed VRSGM algorithm converges faster than others, demonstrating the proposed algorithm's convergence performance. In addition, the proposed VRSGM algorithm having a momentum step converges faster than RR-VR [21], which verifies the discussion that using the momentum step can help improve the convergence result at the end of Remark 3.2.

V. Conclusion

Combining the explicit variance reduction and Nesterov's momentum, this paper has developed a variance-reduced shuffling gradient algorithm for convex finite-sum optimization. With the unified shuffling scheme, the shuffling gradient descent methods own conditionally biased gradients. This paper provides the convergence analysis using an equivalent problem reformulation and backward per-iteration deviation to handle biased gradients. The main result provides a convergence rate of $\mathcal{O}(\frac{1}{T})$, which works for the last returned iterate rather than the average iterate. One future research direction is to reduce the computational cost of the explicit variance reduction in the proposed algorithm by some improved variance reduction technique, such as [19], [22].

References

[1] A. Agrawal, S. Barratt, and S. Boyd, "Learning convex optimization models," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 8, pp. 1355–1364, 2021.

[2] L. Chen, B. Xin, and J. Chen, "Interactive multiobjective evolutionary algorithm based on decomposition and compression," *Sci. China Inf. Sci.*, vol. 64, no. 10, pp. 202 201–, 2021.

[3] G. Lan, *First-order and Stochastic Optimization Methods for Machine Learning*. Springer Nature Switzerland AG: Springer Cham, 2020.

[4] X. Yi, S. Zhang, T. Yang, T. Chai, and K. H. Johansson, "A primal-dual sgd algorithm for distributed nonconvex optimization," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 5, pp. 812–833, 2022.

[5] C. Liu, K. H. Johansson, and Y. Shi, "Private stochastic dual averaging for decentralized empirical risk minimization," *IFAC-PapersOnLine*, vol. 55, no. 13, pp. 43–48, 2022, 9th IFAC Conference on Networked Systems NECSYS 2022.

[6] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, p. 400–407, 1951.

[7] S. Zhao, Y. Xie, and W. Li, "On the convergence and improvement of stochastic normalized gradient descent," *Sci. China Inf. Sci.*, vol. 64, no. 1, p. 132103, 2021.

[8] S. Vaswani, F. Bach, and M. Schmidt, "Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron," in *Proc. 22th Int. Conf. Artif. Intell. and Statist.*, ser. Proceedings of Machine Learning Research, vol. 89. PMLR, 16–18 Apr 2019, pp. 1195–1204.

[9] G. Lan, "An optimal method for stochastic composite optimization," *Math. Program.*, vol. 133, no. 1, pp. 365–397, 2012.

[10] K. Ahn, C. Yun, and S. Sra, "SGD with shuffling: optimal rates without component convexity and large epoch requirements," in *Advances in Neural Inf. Process. Syst.* Curran Associates, Inc., 2020.

[11] M. Gurbuzbalaban, A. Ozdaglar, and P. A. Parrilo, "Why random reshuffling beats stochastic gradient descent," *Math. Program.*, vol. 186, pp. 49–84, 2021.

[12] K. Mishchenko, A. Khaled, and P. Richtarik, "Random reshuffling: Simple analysis with vast improvements," in *Advances in Neural Inf. Process. Syst.*, vol. 33. Curran Associates, Inc., 2020, pp. 17 309–17 320.

[13] I. Safran and O. Shamir, "How good is SGD with random shuffling?" in *Proc. 33th Conf. Learn. Theory*, ser. Proceedings of Machine Learning Research, vol. 125. PMLR, 09–12 Jul 2020, pp. 3250–3284.

[14] T. H. Tran, K. Scheinberg, and L. M. Nguyen, "Nesterov accelerated shuffling gradient method for convex optimization," in *Proc. 39th Int. Conf. Mach. Learn.*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 21 703–21 732.

[15] J. Lei, P. Yi, J. Chen, and Y. Hong, "A communication-efficient linearly convergent algorithm with variance reduction for distributed stochastic optimization," in *Eur. Control Conf. (ECC)*, 2020, pp. 1250–1255.

[16] C. Changyou, W. Wenlin, Z. Yizhe, Q. Su, and L. Carin, "A convergence analysis for a class of practical variance-reduction stochastic gradient MCMC," *Sci. China Inf. Sci.*, vol. 62, no. 1, p. 12101, 2018.

[17] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Inf. Process. Syst.*, vol. 27. Curran Associates, Inc., 2014.

[18] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Inf. Process. Syst.*, vol. 26. Curran Associates, Inc., 2013.

[19] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, "SARAH: A novel method for machine learning problems using stochastic recursive gradient," in *Proc. 34th Int. Conf. Mach. Learn.*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 2613–2621.

[20] Z. Allen-Zhu, "Katyusha: The first direct acceleration of stochastic gradient methods," in *Proc. 49th Annu. ACM SIGACT Symp. on Theory Comput.*, ser. STOC 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 1200–1205.

[21] G. Malinovsky, A. Sailanbayev, and P. Richtárik, "Random reshuffling with variance reduction: New analysis and better rates," *CoRR*, vol. abs/2104.09342, 2021. [Online]. Available: https://arxiv.org/abs/2104.09342

[22] B. Ying, K. Yuan, and A. H. Sayed, "Variance-reduced stochastic learning under random reshuffling," *IEEE Trans. Signal Process.*, vol. 68, pp. 1390–1408, 2020.

[23] K. Yuan, B. Ying, and A. H. Sayed, "On the influence of momentum acceleration on online learning," *J. Mach. Learn. Res.*, vol. 17, no. 192, pp. 1–66, 2016.

[24] K. Mishchenko, A. Khaled, and P. Richtarik, "Random reshuffling: Simple analysis with vast improvements," in *Advances in Neural Inf. Process. Syst.*, vol. 33. Curran Associates, Inc., 2020, pp. 17 309–17 320.

[25] X. Li, M. Meng, and L. Xie, "A linearly convergent algorithm for multi-agent quasi-nonexpansive operators in real hilbert spaces," in *2020 59th IEEE Conf. Decis. Control (CDC)*, 2020, pp. 4903–4908.

[26] X. Yi, S. Zhang, T. Yang, and K. H. Johansson, "Zeroth-order algorithms for stochastic distributed nonconvex optimization," *Automatica*, vol. 142, p. 110353, 2022.

[27] J. Chen, J. Sun, and G. Wang, "From unmanned systems to autonomous intelligent systems," *Engineering*, vol. 12, pp. 16–19, 2022.