Distributed Stochastic Proximal Algorithm With Random Reshuffling for Nonsmooth Finite-Sum Optimization

Xia Jiang\textsuperscript{1}, Xianlin Zeng\textsuperscript{2}, Member, IEEE, Jian Sun\textsuperscript{3}, Senior Member, IEEE, Jie Chen\textsuperscript{4}, Fellow, IEEE, and Lihua Xie\textsuperscript{5}, Fellow, IEEE

Abstract—The nonsmooth finite-sum minimization is a fundamental problem in machine learning. This article develops a distributed stochastic proximal-gradient algorithm with random reshuffling to solve the finite-sum minimization over time-varying multitagent networks. The objective function is a sum of differentiable convex functions and nonsmooth regularization. Each agent in the network updates local variables by local information exchange and cooperates to seek an optimal solution. We prove that local variable estimates generated by the proposed algorithm achieve consensus and are attracted to a neighborhood of the optimal solution with an $O((1/T) + (1/\sqrt{T}))$ convergence rate, where $T$ is the total number of iterations. Finally, some comparative simulations are provided to verify the convergence performance of the proposed algorithm.

Index Terms—Distributed optimization, proximal operator, random reshuffling (RR), stochastic algorithm, time-varying graphs.

I. INTRODUCTION

DISTRIBUTED nonsmooth finite-sum minimization is a basic problem of the popular supervised machine learning [1], [2], [3]. In machine learning communities, optimizing a training model with merely the average loss over a finite dataset usually leads to overfitting or poor generalization. Some nonsmooth regularizations are often included in the cost function to encode prior knowledge, which also introduces the challenge of nonsmoothness. What is more, in many practical network systems [4], [5], training data are naturally stored at different physical nodes, and it is expensive to collect data and train the model in one centralized node. Compared to the centralized setting, the distributed setting makes use of multicomputational sources to train the learning model in parallel, leading to potential speedup. However, since the data are distributed and the communication is limited, more involved approaches are needed to solve the minimization problem. Therefore, distributed finite-sum minimization has attracted much attention in machine learning, especially for large-scale applications with big data.

For large-scale nonsmooth finite-sum optimization problems, there have been many efficient centralized and distributed first-order algorithms, including subgradient-based methods [6], [7], [8] and proximal-gradient algorithms [9], [10], [11], [12], [13], [14]. Subgradient-based methods are very generic in optimization with the expense of slow convergence. To be specific, subgradient-based algorithms may increase the objective function of the optimization problem for small step sizes [10]. Proximal gradient algorithms are considered more stable than subgradient-based methods and often own better numerical performance than subgradient-based methods for convex optimization [15]. Hence, proximal gradient algorithms have attracted great interest in large-scale nonsmooth optimization problems. Particularly, training data are allocated to different computing nodes and the nonsmooth finite-sum optimization needs to be handled efficiently in a distributed manner. Many distributed deterministic proximal gradient algorithms have been proposed with guaranteed convergence for nonsmooth optimization over time-invariant and time-varying graphs [12], [13], [14]. These distributed works are developed under the primal-dual framework and are applicable for constrained nonsmooth optimization. However, these distributed algorithms focus on deterministic nonsmooth optimization and require computing the full gradients of all local functions at each iteration. The per iteration computational cost of a deterministic algorithm is much higher than that of a stochastic gradient method, which hinders the application of deterministic algorithms for nonsmooth optimization with large-scale data.

In large-scale machine learning settings, various first-order stochastic methods are leading algorithms due to their scalability and low computational requirements. Decentralized stochastic gradient descent (SGD) algorithms have gained a
lot of attention recently, especially for the traditional federated learning setting with a star-shaped network topology [16], [17], [18]. All of these methods are not fully distributed in the sense that they require a central parameter server. The potential bottleneck of the star-shaped network is a possible communication traffic jam on the central sever, and the performance will be significantly degraded when the network bandwidth is low. To consider a more general distributed network topology without a central server, many distributed stochastic gradient algorithms have been studied [19], [20], [21], [22], [23] for convex finite-sum optimization. To avoid the degenerated performance of algorithms with diminishing step sizes, some recent works [22], [23] have proposed consensus-based distributed SGD methods with nondiminishing (constant) step sizes for smooth convex optimization over time-invariant networks. With the variance reduction technique to handle the variance of the local stochastic gradients at each node, some distributed efficient stochastic algorithms with constant step sizes have been studied for strongly convex smooth optimization over time-invariant undirected networks [24] and time-invariant directed networks [25]. However, most of the existing distributed stochastic algorithms are designed for smooth optimization and are only applicable over time-invariant networks.

In practice, a communication network may be time-varying since network attacks may cause failures of communications and the connectivity of a network may change, especially for mobile networks [4], [26]. Therefore, it is necessary to study distributed algorithms over time-varying networks. Considering the unreliable network communication, some works have studied distributed asynchronous stochastic algorithms with diminishing step sizes for smooth convex optimization over multiagent networks [20], [21]. To avoid the degenerated performance of algorithms with diminishing step sizes, Nedic and Ozdaglar [7] developed a distributed gradient algorithm with a multistep consensus mapping for multiagent optimization. This multistep consensus mapping has also been widely applied in the follow-up works [27], [28] for multiagent consensus and nonsmooth optimization over time-varying networks. Different from these deterministic works, this article studies a distributed stochastic gradient algorithm with a sample-without-replacement heuristic for non-smooth finite-sum optimization over time-varying multiagent networks.

Random reshuffling (RR) is a simple, popular but elusive stochastic method for finite-sum minimization in machine learning. Contrasted with SGD, where the training data are sampled uniformly with replacement, the sampling-without-replacement RR method learns from each data point in each epoch and often owns better performance in many practical problems [29], [30], [31]. The convergence properties of SGD are well-understood with tight lower and upper bounds in many settings [32], [33]; however, the theoretical analysis of RR had not been studied until recent years. The sampling-without-replacement in RR introduces a significant complication that the gradients are now biased, which implies that a single iteration does not approximate a full gradient descent step. The incremental gradient (IG) algorithm is one special case of the shuffling algorithm. The IG generates a deterministic permutation before the start of epochs and reuses the permutation in all subsequent epochs. In contrast, in the RR, a new permutation is generated at the beginning of each epoch. The implementation of IG has the challenge of choosing a suitable permutation for cycling through the iterations [34], and the IG is susceptible to bad orderings compared to RR [35]. Therefore, it is necessary to study a RR algorithm for convex nonsmooth optimization. For strongly convex optimization, the seminal work [30] theoretically characterized various convergence rates for the RR method. One inspiring recent work by Mishchenko [36] has provided some involved yet insightful proofs for the convergence behavior of RR under weak assumptions. For smooth finite-sum minimization, Meng et al. [37] have studied decentralized stochastic gradient with shuffling and provided insights for practical data processing. What is more, the recent work [38] has studied centralized proximal gradient descent algorithms with RR (Prox-RR) for nonsmooth finite-sum optimization and extended the Prox-RR to federated learning. However, these works are not applicable to fully distributed settings, and the Prox-RR needs a strong convexity assumption, which hinders its application.

Inspired by the existing deterministic and stochastic algorithms, we develop a distributed stochastic proximal algorithm with random reshuffling (DPG-RR) for nonsmooth finite-sum optimization over time-varying multiagent networks, extending the RR to distributed nonsmooth convex optimization.

The contributions of this article are summarized as follows.

1) For nonsmooth convex optimization, we propose a distributed stochastic algorithm DPG-RR over time-varying networks. Although there are a large number of works on proximal SGD [39], [40], few works have studied how to extend RR to solve convex optimization with nonsmooth regularization. This article extends the recent centralized proximal algorithm Prox-RR in [38] for strongly convex nonsmooth optimization to general convex optimization and distributed settings. Another very related decentralized work is [37], which has provided important insights into distributed SGD with data shuffling under master–slave frameworks. This article extends [37] to fully distributed settings and makes it applicable for nonsmooth optimization. To our best knowledge, this is the first attempt to study fully distributed stochastic proximal algorithms with RR over time-varying networks.

2) The proposed algorithm DPG-RR needs fewer proximal operators compared with proximal SGD and is applicable for time-varying networks. To be specific, for the nonsmooth optimization with $n$ local functions, the standard proximal SGD applies the proximal operator after each stochastic gradient step [40] and needs $n$ proximal evaluations. In contrast, the proposed algorithm DPG-RR only operates one single proximal evaluation at each iteration. In addition, with the design of multistep consensus mapping, this proposed algorithm enables local variable estimates closer to each other, which is crucial for distributed nonsmooth optimization over time-varying networks. Compared with the existing IG
algorithm in [41], the proposed RR algorithm in this article requires a weaker assumption, where the cost function does not need to be strongly convex.

3) This work provides a complete and rigorous theoretical convergence analysis for the proposed DPG-RR over time-varying multiagent networks. Due to the sampling-without-replacement mechanism of RR, the local stochastic gradients are biased estimators of the full gradient. In addition, because of scattered information in the distributed setting, there are differences between local and global objective functions and, thus, local and global gradients. To handle the differences between local and global gradients, this article proves the summability of the error sequences of the estimated gradients. Then, we prove the convergence of the transformed stochastic algorithm by some novel proof techniques inspired by Mishchenko et al. [36].

The remainder of this article is organized as follows. The preliminary mathematical notations and proximal operator are introduced in Section II. The problem description and the design of a distributed solver are provided in Section III. The convergence performance of the proposed algorithm is analyzed in Section IV. Numerical simulations are provided in Section V, and the conclusion is made in Section VI.

II. NOTATIONS AND PRELIMINARIES

A. Notations

We denote $\mathbb{R}$ as the set of real numbers, $\mathbb{N}$ the set of natural numbers, $\mathbb{R}^n$ the set of $n$-dimensional real column vectors, and $\mathbb{R}^{n \times m}$ the set of $n \times m$ real matrices, respectively. We denote $\nu^T$ as the transpose of vector $\nu$. In addition, $\| \cdot \|$ denotes the Euclidean norm and $\langle \cdot, \cdot \rangle$ the inner product, which is defined by $(a, b) = a^Tb$. The vectors in this article are column vectors unless otherwise stated. The notation $[n]$ denotes the set $\{1, \ldots, n\}$. For a differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\nabla f(x)$ denotes the gradient of function $f$ with respect to $x$. For a convex nonsmooth function $h : \mathbb{R}^n \rightarrow \mathbb{R}$, $\partial h(x)$ denotes the subdifferential of function $h$ at $x$. The $\epsilon$-subdifferential of a nonsmooth convex function $h$ at $x$, denoted by $\partial_\epsilon h(x)$, is the set of vectors $y$ such that, for all $q$

$$h(q) - h(x) \geq y^T (q - x) - \epsilon.$$  \hspace{1cm} (1)

B. Proximal Operator

For a proper nondifferentiable convex function $h : \mathbb{R}^n \rightarrow (-\infty, \infty]$ and a scalar $\alpha > 0$, the proximal operator is defined as

$$\text{prox}_{\alpha h}(x) = \text{argmin}_{z \in \mathbb{R}^n} h(z) + \frac{1}{2\alpha} \| z - x \|^2. \hspace{1cm} (2)$$

The minimum is attained at a unique point $y = \text{prox}_{\alpha h}(x)$, which means that the proximal operator is a single-valued map. In addition, it follows from the optimality condition for convex optimization that

$$0 \in \partial h(y) + \frac{1}{\alpha} (y - x). \hspace{1cm} (3)$$

The following proposition presents some properties of the proximal operator.

**Proposition 1** [42]: Let $h : \mathbb{R}^n \rightarrow (-\infty, \infty]$ be a closed proper convex function. For a scalar $\alpha > 0$ and $x \in \mathbb{R}^n$, let $y = \text{prox}_{\alpha h}(x)$. For $x, \tilde{x} \in \mathbb{R}^n$, $\| \text{prox}_{\alpha h}(x) - \text{prox}_{\alpha h}(\tilde{x}) \| \leq \| x - \tilde{x} \|$, which is also called the nonexpansiveness of proximal operator.

When there exists error $\epsilon$ in the computation of the proximal operator, we denote the inexact proximal operator by $\text{prox}_{\alpha h}^\epsilon (\cdot)$, which is defined as

$$\text{prox}_{\alpha h}^\epsilon (x) \triangleq \begin{cases} \tilde{x} & \text{if } \frac{1}{2\alpha} \| \tilde{x} - x \|^2 + h(\tilde{x}) \leq \epsilon + \min_{z \in \mathbb{R}^n} \left\{ \frac{1}{2\alpha} \| z - x \|^2 + h(z) \right\} \\ x & \text{otherwise} \end{cases} \hspace{1cm} (4)$$

III. PROBLEM DESCRIPTION AND ALGORITHM DESIGN

In this article, we aim to solve the following nonsmooth finite-sum optimization problem over a multiagent network:

$$\min_{x \in \mathbb{R}^d} F(x), \hspace{0.5cm} F(x) = f(x) + \phi(x) = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{n} f_{j,i}(x) + \phi(x) \hspace{1cm} (5)$$

where $x \in \mathbb{R}^d$ is the unknown decision variable, $m$ is the number of agents in the network, and $n$ is the number of local samples. The global sample size of the multiagent network is $N = mn$. The cost function $f(x)$ in (5) denotes the global cost of the multiagent network and is differentiable. The regularization $\phi$ is nonsmooth, which encodes some prior knowledge. In the multiagent network, each agent $j$ only knows $n$ local samples and the corresponding cost functions $f_{j,i}(\cdot)$ to update the local decision variable. In addition, the nonsmooth regularization $\phi$ is available to all agents. In this setting, we assume that each agent uses local cost functions and communications with its neighbors to obtain the same optimal solution $x_*$ of problem (5).

In real-world applications of distributed settings, it is difficult to keep all communication links between agents connected and stable due to the existence of networked attacks and limited bandwidth. We consider a general time-varying multiagent communication network $G(t) = ([m], \mathcal{E}(t), A(t))$, where $\mathcal{E}(t)$ denotes the set of communication edges at time $t$. At each time $t$, agent $j$ can only communicate with agent $i$ if and only if $(j, i) \in \mathcal{E}(t)$. In addition, the neighbors of agent $j$ at time $t$ are the agents that agent $j$ can communicate with at time $t$. The adjacent matrix $\hat{A}(t)$ of the network is a square $m \times m$ matrix, whose elements indicate the weights of the edges in $\mathcal{E}(t)$.

For the finite-sum optimization (5), we make the following assumption.

**Assumption 2**: The cost function $f$ and the regular function $\phi$ satisfy the following conditions.

1) Functions $f$ and $\phi$ are convex.

2) Local function $f_{j,i}$ has a Lipschitz-continuous gradient with constant $L > 0$, i.e., for every $x, y \in \mathbb{R}^d$

$$\| \nabla f_{j,i}(x) - \nabla f_{j,i}(y) \| \leq L \| x - y \| \hspace{1cm} (6)$$

while the regular function $\phi$ is nonsmooth.
3) There exists a scalar $G_\phi$ such that, for each subgradient $z \in \partial \phi(x)$, $\|z\| < G_\phi$ for every $x \in \mathbb{R}^d$.

4) There exists a scalar $G_f$ such that, for each agent $j$ and every $x \in \mathbb{R}^d$, $\|\nabla f_j(x)\| < G_f$.

5) Furthermore, $F$ is lower bounded by some $F_\epsilon \in \mathbb{R}$, and the optimization problem owns at least one optimal solution $x^\star$.

Assumption 2 is common in stochastic algorithm research [37], [43], and finite-sum minimization problems satisfying Assumption 2 arise in many applications. Here, we provide some real-world examples.

Example 1: In the binary classification problem [44], the logistic regression optimization aims to obtain a predictor $x$ to estimate the categorical variables of the testing data. When the large-scale training samples are allocated to $m$ different agents, the local cost function in (5) is a cross-entropy error function (CNF), $f_j(x) = \ln(1 + \exp(-J_j(a_{ij}, x)))$, where $a_{ij}$ and $f_{ij}$ denote the feature vector and categorical value of the $i$th training sample at agent $j$. Then, the gradient of the convex function $f_{ij}$ is bounded and satisfies the Lipschitz condition in (6).

For the communication topology, the time-varying multiagent network of the finite-sum optimization (5) satisfies the following assumption.

Assumption 3: Consider the undirected time-varying network $\mathcal{G}(t)$ with adjacent matrices $A(t) = [a_{ij}(t)]$, $i = 1, 2, \ldots$

1) For each $t$, the adjacent matrix $A(t)$ is doubly stochastic.

2) There exists a scalar $\eta \in (0, 1)$ such that $a_{ij}(t) \geq \eta$ for all $j \in [m]$. In addition, $a_{ij}(t) \geq \eta$ if $[i, j] \in E(t)$, and $a_{jj}(t) = 0$ otherwise.

3) The time-varying graph sequence $\mathcal{G}(t)$ is uniformly connected. That is, there exists an integer $B \geq 1$ such that agent $j$ sends its information to all other agents at least once every $B$ consecutive time slots.

Remark 4: In Assumption 3, 1) is common for undirected graphs and balanced directed graphs, 2) means that each agent gives nonnegligible weights to the local estimate and the estimates received from neighbors, and 3) implies that the time-varying network is able to transmit information among any pair of agents in bounded time. Assumption 3 is widely adopted in the existing literature [7], [28].

Next, we design a distributed proximal gradient algorithm with RR for each agent $j \in [m]$ in the multiagent network. At the beginning of each iteration $t$, we sample indices $\pi^0_j, \ldots, \pi^{n-1}_j$ without replacement from $\{1, \ldots, n\}$ such that the generated $\pi_j = [\pi^0_j, \ldots, \pi^{n-1}_j]$ is a random permutation of $\{1, \ldots, n\}$. Then, we process with $n$ inner iterations of the form

$$x_{j,t+1}^i = x_{j,t}^i - \gamma \nabla f_{j,\pi_i}(x_{j,t}^i), \quad i \in \{0, \ldots, n-1\}$$

(7)

where $\gamma$ is a constant step size, the superscript $i$ denotes the $i$th inner iteration, and the subscripts $j$ and $t$ denote the $j$th agent and the $t$th outer iteration, respectively.

Then, to achieve the consensus between different agents, a multistep consensus mapping is applied as

$$v_{j,t} = \sum_{i=1}^m \lambda_{j,\pi_i} x_{i,t}^n$$

(8)

where $\lambda_{j,\pi_i}$ is the $(j, i)$th element of matrix $\Phi(\mathbb{T}(t) + t, \mathbb{T}(t))$ for $j, i \in \{1, \ldots, m\}$. The notation $\mathbb{T}(t)$ is the total number of communication steps before iteration $t$, and $\Phi$ is a transition matrix, defined as

$$\Phi(t, s) = A(t)A(t-1), \ldots, A(s+1)A(s), \quad t > s \geq 0$$

(9)

where $A(t) = [a_{ij}(t)]_{i,j=1,\ldots,m}$ is the adjacent matrix of the multiagent network at iteration $t$. To be specific, using vector notations $v_t = [v_{j,1}, \ldots, v_{j,m}]$ and $x^n_t = [x^n_{j,1}, \ldots, x^n_{j,m}]$, we rewrite (8) as

$$v_t = \Phi(\mathbb{T}(t) + t_s, \mathbb{T}(t))x^n_t = A(\mathbb{T}(t) + t_s)A(\mathbb{T}(t) + t - 1, \ldots, A(\mathbb{T}(t))x^n_t$$

Agents perform $t$ communication steps at iteration $t$. At each communication step, agents exchange their estimates $x^n_{j,t}$ and linearly combine the received estimates using weights $A(t)$. This mapping (8) is referred to as a multistep consensus mapping because linear combinations of estimates bring the estimates of different agents close to each other.

Finally, for the nonsmooth regular function, we use the proximal operator

$$x_{j,t+1} = \text{prox}_{\gamma \partial \phi}(v_{j,t})$$

(10)

to obtain the local variable estimate $x_{j,t+1}$ of agent $j$ at the next iteration.

The proposed DPG-RR is formally summarized in Algorithm 1.

Remark 5: Compared with the standard proximal SGD, where the proximal operator is applied after each SGD, the proposed DPG-RR only operates one single proximal evaluation at each iteration, which is $(1/n)$ of the proximal evaluations in the proximal SGD. Therefore, the proposed

---

**Algorithm 1** DPG-RR for Agent $j \in [m]$

1: (S.1) Initialization:
2: Step-size $\gamma > 0$; initial vector $x_{j,0} \in \mathbb{R}^d$ for $j \in [m]$; number of epochs $T$.
3: (S.2) Iterations:
4: for $t = 0, 1, \ldots, T - 1$ do
5: Generate a random permutation $\pi_t = (\pi^0_t, \ldots, \pi^{n-1}_t)$
6: $x^0_{j,t} = x_{j,t}$
7: for $i = 0, 1, \ldots, n - 1$ do
8: $x_{j,t+1} = x^i_{j,t} - \gamma \nabla f_{j,\pi_i}(x^i_{j,t})$
9: end for
10: $v_{j,t} = \sum_{i=1}^m \lambda_{j,\pi_i} x^i_{t,n}$
11: $x_{j,t+1} = \text{prox}_{\gamma \partial \phi}(v_{j,t})$

---
DPG-RR owns a lower computational burden and is applicable for nonsmooth optimization where the proximal operator is expensive to calculate.

Remark 6: Compared with the fully random sampling procedure, which is used in the well-known SGD method, one main advantage of the RR procedure is its intrinsic ability to avoid cache misses when reading the data from memory, enabling a faster implementation. In addition, an RR algorithm usually converges in fewer iterations than distributed SGD [30], [38]. It is due to that the RR procedure learns from each sample in each epoch, while the SGD can miss learning from some samples in any given epoch. The gradient method with the incremental sampling procedure is a special case of the RR gradient method, which has been investigated in [36].

Remark 7: The work in [37] has studied the convergence rates for strongly convex, convex, and nonconvex optimization under different shuffling procedures. Meng et al. [37] have provided a unified analysis framework for three shuffling procedures, including global shuffling, local shuffling, and insufficient shuffling. Unlike the work [37], this article focuses on nonsmooth optimization with a local shuffling procedure, which is more applicable for the distributed setting. In addition, this proposed distributed DPG-RR owns a comparable convergence rate and allows nonsmooth functions and time-varying communications between different agents for convex finite-sum optimization.

The convergence performance of the proposed algorithm is discussed in the following theorem, and the proof is provided in Section IV.

Theorem 8: Suppose that Assumptions 2 and 3 hold, and the step size $\gamma = M/\sqrt{T}$, $M \leq \sqrt{6}/6L_n$. Then, Algorithm 1 possesses the properties.

1) The local variable estimates $x_{j,t}$ achieve consensus, and

$$\lim_{t \to \infty} \|x_{j,t} - \bar{x}_t\| = 0,$$

where $\bar{x}_t \triangleq (1/m) \sum_{j=1}^m x_{j,t}$.

2) In addition, $\mathbb{E}[F(\bar{x}_t) - F(x^*)] = O((1/T) + (1/\sqrt{T}))$,

where $\bar{x}_t \triangleq (1/mT) \sum_{t=1}^T \sum_{j=1}^m x_{j,t}$.

Remark 9: If the local sample size $n$ increases, the convergence rate of the proposed algorithm will become slower. It follows from the proof of Theorem 8 that the constant term in the convergence rate is proportional to the third-order polynomial of $n$. Since the proposed algorithm is distributed, we can reduce the effect of local sample size $n$ on the convergence rate by using more agents for large-scale problems, whereas more agents may require higher communication costs over multiagent networks. In addition, the convergence rate also depends on other factors, such as the Lipschitz constant of objective functions, the upper bounds of subgradients, the shuffling variance, and the bounded intercommunication interval.

Remark 10: Theorem 8 extends that of the centralized Prox-RR in [38] to distributed settings and nonsmooth convex optimization. Compared with the popular stochastic subgradient method with a diminishing step size for nonsmooth optimi-

zation, whose convergence rate is $O((\log(T))/\sqrt{T})$ [19], the proposed DPG-RR has a faster convergence rate.

IV. THEORETICAL ANALYSIS

In this section, we present theoretical proofs for the convergence performance of the proposed algorithm. The proof sketch includes three parts.

1) Transform the DPG-RR to an algorithm with some error sequences, and prove that the error sequences are summable.

2) Estimate the bound of the forward per-epoch deviation of the DPG-RR.

3) Prove the consensus property of local variables and the convergence performance in Theorem 8.

Each part is discussed in detail in the subsequent subsections. To present the transformation, we define

$$\bar{x}_t \triangleq \frac{1}{m} \sum_{j=1}^m x_{j,t} \in \mathbb{R}^d,$$

$$\bar{v}_t \triangleq \frac{1}{m} \sum_{j=1}^m v_{j,t} \in \mathbb{R}^d,$$

$$z_{t+1} \triangleq \text{prox}_{\gamma_D}(\bar{v}_t).$$

A. Transformation of DPG-RR and the Summability of Error Sequences

Proposition 11: Suppose that Assumptions 2 and 3 hold. The average variable of Algorithm 1 satisfies

$$\bar{x}_{t+1} \in \text{prox}_{\gamma_D,\phi}\left(\bar{x}_t - \gamma \left(\sum_{i=0}^{n-1} (\nabla f_i(x_i^e) + e_{i,t})\right)\right) \quad (11)$$

where $f_i(x) = (1/m) \sum_{j=1}^m f_{j,i}(x)$, and the error sequences $\{e_i^t\}_{i=0}^\infty$ and $\{e_{i,t+1}\}_{i=0}^\infty$ satisfy

$$\|e_i^t\| \leq \frac{L}{m} \sum_{j=1}^m \|x_{j,t} - \bar{x}_i^t\|^2 \quad (12a)$$

$$e_{i,t+1} \leq \frac{2G_D}{m} \sum_{j=1}^m \|v_{j,t} - \bar{v}_i\| + \frac{1}{2\gamma} \left(\frac{1}{m} \sum_{j=1}^m \|v_{j,t} - \bar{v}_i\|\right)^2. \quad (12b)$$

Proof: See Appendix B.

Then, inspired by [28, Sec. III-B], we discuss the summabilities of error sequences $\{\gamma \|e_i^t\|\}$ and $\{e_i\}$ in the following proposition.

Proposition 12: Under Assumptions 2 and 3, error sequences $\{e_i^t\}$ and $\{e_i\}$ satisfy $\|e_i^t\| \leq b_{e,t} + \gamma C_0, e_i \leq b_{e,t}$ and $\sqrt{e_i} \leq b_{\sigma,t}$, where $b_{e,t}, b_{\sigma,t},$ and $b_{\tau,t}$ are polynomial-geometric sequences and $C_0$ is a constant.

Proof: See Appendix C.

1 Let $\{a_i\}$ be a sequence of real numbers. The series $\sum_{i=1}^\infty a_i$ is summable if and only if the sequence $x_n \triangleq \sum_{i=1}^n a_i, n \in \mathbb{N}$ converges.
B. Boundness of the Forward Per-Epoch Deviation

Before presenting the boundness analysis, we introduce some necessary quantities for the RR technique.

**Definition 13** [36]: For any $i$, the quantity $D_f(x, y) \triangleq f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y \rangle$ is the Bregman divergence between $x$ and $y$ associated with function $f_i$.

If the function $f_i$ is $L$-smooth, then, for all $x, y \in \mathbb{R}^d$, $D_f(x, y) \leq (L/2)\|x - y\|^2$. The difference between the gradients of a convex and $L$-smooth function $f_i$ satisfies
\[
\|\nabla f_i(x) - \nabla f_i(y)\|^2 \leq 2LD_f(x, y). \tag{13}
\]

In addition, the forward per-epoch deviation of the proposed algorithm is introduced as follows.

**Definition 14**: The forward per-epoch deviation of DPG-RR for the $t$th epoch is defined as
\[
\mathcal{V}_t \triangleq \sum_{i=0}^{n-1} \|\bar{x}_i^t - \bar{x}_{t+1}\|^2. \tag{14}
\]

Now, we show that the forward per-epoch deviation $\mathcal{V}_t$ of DPG-RR is upper bounded in the following lemma.

**Lemma 15**: If Assumption 2 holds, then the forward per-epoch deviation $\mathcal{V}_t$ satisfies
\[
\mathbb{E}[\mathcal{V}_t] \leq 24\gamma^2 Ln^2 \sum_{i=0}^{n-1} \mathbb{E}\left[D_{f_i}(x_*, \bar{x}_i^t)\right] + 3\gamma^2 n^2 \bar{\sigma}_e^2 + 9\gamma^2 nG_f^2
\]
\[
+ 6\gamma^2 n^2 G_{\phi}^2 + 12\gamma^3 (\gamma^2 b_{e,t} + \gamma^4 C_0^2) + 4n\gamma b_{e,t} \tag{15}
\]
where $\mathcal{V}_t$ is defined in (14), $\bar{\sigma}_e^2 \triangleq (1/n)\sum_{i=1}^n \|\nabla f_i(x_*) - (1/n)\nabla \bar{f}(x_*^t)\|^2$, and $b_{e,t}$ and $b_{e,t}$ are defined in (53) and (54), shown at the top of the page 14, respectively.

**Proof**: See Appendix D. □

C. Proof of Theorem 8

**Proof**: Suppose Assumptions 2 and 3 hold. With the results in lemmas, we are ready to provide the theoretical analysis for the convergence performance of the proposed algorithm.

1) First, we prove the consensus property of the generated local variables, i.e., $\lim_{t \to \infty} x_{j,t} \to \bar{x}_t$ for all agent $j$. By (40) in Lemma 20, $\sum_{j=1}^m \|x_{j,t} - \bar{x}_t\| \leq 2m \Gamma \mathbb{E} \sum_{i=1}^m \|x_{i,t}^n - \bar{x}_i\|$ holds. In addition, with Lemmas 17 and 21, we obtain that
\[
\sum_{i=1}^m \|x_{i,t}^n - \bar{x}_i\| < \infty.
\]

Then, $\sum_{i=1}^m \sum_{j=1}^m \|x_{j,t} - \bar{x}_i\|$ is nonnegative and hence, $\lim_{t \to \infty} \sum_{j=1}^m \|x_{j,t} - \bar{x}_t\| = 0$

and hence, $\lim_{t \to \infty} \|x_{j,t} - \bar{x}_t\| = 0$ for each agent $j$.

Therefore, local variable estimates $x_{j,t}$ of all agents achieve consensus and converge to the average $\bar{x}_t$ as $t \to \infty$.

2) Next, we consider the convergence rate of the proposed algorithm. By Lemma 16 and (11), we have
\[
\bar{x}_t = \gamma \bar{g}_{t} + \bar{x}_{t+1} + \gamma \bar{e}_{t+1} + p_{t+1} + \gamma \bar{d}_{t+1} \tag{16}
\]
where $\bar{e}_{t+1} = \sum_{i=0}^{n-1} \bar{e}_i^t$, $\bar{d}_{t+1} \in \partial_{e,t}\phi(\bar{x}_{t+1})$, and $\bar{g}_t \triangleq \sum_{i=0}^{n-1} \nabla f_i^t(\bar{x}_i^t)$. Then, the square of the norm between $\bar{x}_t$ and the optimal solution $x_*$ satisfies
\[
\|\bar{x}_t - x_*\|^2
\]
\[
= \|\gamma \bar{g}_t + \gamma \bar{e}_t + p_{t+1} + \gamma \bar{d}_{t+1} - x_*\|^2
\]
\[
\geq \|\bar{x}_t - x_*\|^2 + 2\gamma \sum_{i=0}^{n-1} \langle \bar{x}_{t+1} - x_*, \nabla f_i^t(\bar{x}_i^t) \rangle
\]
\[
+ 2(\bar{x}_{t+1} - x_*, \gamma \bar{e}_{t+1} + p_{t+1} + \gamma \bar{d}_{t+1}). \tag{17}
\]

For the second term in (17), we have for any $i$
\[
\langle \bar{x}_{t+1} - x_*, \nabla f_i^t(\bar{x}_i^t) \rangle
\]
\[
= f_i^t(\bar{x}_{t+1}) - f_i^t(x_*) + f_i^t(x_*) - f_i^t(\bar{x}_i^t) - \langle \nabla f_i^t(\bar{x}_i^t), x_* - \bar{x}_i^t \rangle
\]
\[
- [f_i^t(\bar{x}_{t+1}) - f_i^t(\bar{x}_i^t)] - [f_i^t(\bar{x}_{t+1}) - f_i^t(\bar{x}_i^t)]
\]
\[
= [f_i^t(\bar{x}_{t+1}) - f_i^t(x_*)] + D_{f_i^t}(x_*, \bar{x}_i^t) - D_{f_i^t}(\bar{x}_{t+1}, \bar{x}_i^t). \tag{18}
\]

Summing $f_i^t(\bar{x}_{t+1}) - f_i^t(x_*)$ over $i$ from 0 to $n - 1$ gives
\[
\sum_{i=0}^{n-1} [f_i^t(\bar{x}_{t+1}) - f_i^t(x_*)] = f(\bar{x}_{t+1}) - f_* \tag{19}
\]

Next, we bound the third term in (18) with $L$-smoothness in Assumption 2.2) as
\[
D_{f_i^t}(\bar{x}_{t+1}, \bar{x}_i^t) \leq \frac{L}{2} \|\bar{x}_{t+1} - \bar{x}_i^t\|^2. \tag{20}
\]

By summing (20) over $i$ from 0 to $n - 1$, we obtain an upper bound of the forward deviation $\mathcal{V}_t$ by Lemma 15 that
\[
\sum_{i=0}^{n-1} \mathbb{E}\left[D_{f_i^t}(\bar{x}_{t+1}, \bar{x}_i^t)\right]
\]
\[
\leq \frac{L}{2} \mathbb{E}[\mathcal{V}_t]
\]
\[
\leq 12\gamma^2 Ln^2 \sum_{i=0}^{n-1} \mathbb{E}\left[D_{f_i^t}(x_*, \bar{x}_i^t)\right] + 3\gamma^2 n^2 \bar{\sigma}_e^2
\]
\[
+ 9\gamma^2 LnG_f^2 + 3\gamma^2 nL^2G_{\phi}^2 + 6 Ln^3(\gamma^2 b_{e,t} + \gamma^4 C_0^2)
\]
\[
+ 2Ln \gamma b_{e,t}. \tag{21}
\]

With the above inequality, we estimate the lower bound of the sum of the second and third terms in (18) as
\[
\sum_{i=0}^{n-1} \mathbb{E}\left[D_{f_i^t}(x_*, \bar{x}_i^t)\right] - 3\gamma^2 Ln^2 \bar{\sigma}_e^2
\]
\[
- 9\gamma^2 LnG_f^2 - 3\gamma^2 nL^2G_{\phi}^2 - 6 Ln^3(\gamma^2 b_{e,t} + \gamma^4 C_0^2)
\]
\[
- 2Ln \gamma b_{e,t}
\]
\[
\geq -3\gamma^2 Ln^2 \bar{\sigma}_e^2 - 9\gamma^2 LnG_f^2 - 3\gamma^2 nL^2G_{\phi}^2
\]
\[
- 6Ln^3(\gamma^2 b_{e,t} + \gamma^4 C_0^2) - 2Ln \gamma b_{e,t} \tag{21}
\]
where the last inequality holds since \((1 - 12 \gamma^2 L^2 n^2) \geq 0\) by 
\[\gamma \leq \sqrt{3/6 \ln n},\] and 
\[\sum_{i=0}^{T-1} D_t(x_i, \bar{x}_i)\) is nonnegative due to the convexity.

By rearranging (17), we have
\[\|\bar{x}_{t+1} - x_\epsilon\|^2 \leq \|\bar{x}_t - x_\epsilon\|^2 - 2\gamma (f(\bar{x}_t) - f_\epsilon)\]
\[\leq \|\bar{x}_t - x_\epsilon\|^2 - 2\gamma (f(\bar{x}_t) - f_\epsilon)\]
\[\leq \|\bar{x}_t - x_\epsilon\|^2 - 2\gamma (f(\bar{x}_t) - f_\epsilon)\]
\[+ 2\gamma (1 + 2 \gamma nL) \sum_{t=0}^{T-1} b_{e,t} + 2\gamma (1 + 2 \gamma nL) \sum_{t=0}^{T-1} \lambda_t \|x_{t+1} - x_\epsilon\| . \] (23)

Then, we need to estimate an upper bound of \(\mathbb{E}[\|\bar{x}_{t+1} - x_\epsilon\|^2]\). By the transformation of (23) and the fact that \(\mathbb{E}[\|\bar{x}_t - x_\epsilon\|^2] \leq \mathbb{E}[\|\bar{x}_t - x_\epsilon\|^2]\)
\[\mathbb{E}[\|\bar{x}_t - x_\epsilon\|^2] \leq S_T + 2\sum_{t=0}^{T-1} \lambda_t \|x_{t+1} - x_\epsilon\|\]
where \(S_T = \|\bar{x}_0 - x_\epsilon\|^2 + 9T \gamma^3 \ln G_\gamma^2 + 3T \gamma^3 \ln^2 \sigma^2 + 6T \gamma^3 nL G^2_\phi + 12n^3 \gamma^3 T C_0^3 + 12 \gamma^3 \ln^{T-1} \sum_{t=0}^{T-1} b_{e,t}^2 + 2 \gamma (1 + 2 \gamma nL) \sum_{t=0}^{T-1} b_{e,t}\). It follows from Lemma 19 that
\[\mathbb{E}[\|\bar{x}_t - x_\epsilon\|^2] \leq \sum_{t=0}^{T-1} (\gamma n b_{e,t} + n \gamma^2 C_0 + \sqrt{2 \gamma} b_{e,t})\]
where the sequences \(b_{e,t}, b_{e,t},\) and \(b_{e,t}\) are polynomial-geometric sequences and are summable by Lemma 17.

By the summability of \(b_{e,t}, b_{e,t},\) and \(b_{e,t}\), there exist some scalars \(A\) and \(B\) such that
\[A_T = \sum_{t=0}^{T-1} (\gamma n b_{e,t} + n \gamma^2 C_0 + \sqrt{2 \gamma} b_{e,t}) \leq A < \infty \]
where we use the step size range \(\gamma \leq 1/\sqrt{T}\), and
\[B_T = TD + 12 \gamma^3 (1 + 2 \gamma nL) \sum_{t=0}^{T-1} b_{e,t}^2 \]
\[\leq TD + B \]
where \(D \triangleq \gamma^3 (9 \ln G^2_\gamma + 3 \ln^2 \sigma^2 + 6 \ln G^2_\phi + 12 n^3 \gamma^3 T C_0^3).\) Then, \(\mathbb{E}[(\bar{x}_t - x_\epsilon)]\) satisfies
\[\mathbb{E}[\|\bar{x}_t - x_\epsilon\|^2] \leq A + (\|\bar{x}_0 - x_\epsilon\|^2 + (TD + B) + A^2)^{1/2}.
\]
Since \(\{A_t\}\) and \(\{B_t\}\) are increasing sequences, we have for \(t \leq T\)
\[\mathbb{E}[\|\bar{x}_t - x_\epsilon\|^2] \leq A_T + (\|\bar{x}_0 - x_\epsilon\|^2 + B_T + A_T^2)^{1/2} \]
\[\leq 2A_T + \|\bar{x}_0 - x_\epsilon\| + \sqrt{TD + B} \]
(26)

Now, we can bound the right-hand side of (23) with (26)
\[2 \gamma \sum_{t=1}^{T-1} \mathbb{E}[F(\bar{x}_t) - F_\epsilon] \]
\[\leq \|\bar{x}_0 - x_\epsilon\|^2 + B_T + 2A_T(2A_T + \|\bar{x}_0 - x_\epsilon\| + \sqrt{B_T}) \]
\[\leq \|\bar{x}_0 - x_\epsilon\|^2 + TD + B + 2A_T(2A_T + \|\bar{x}_0 - x_\epsilon\| + \sqrt{B_T}) \]
By dividing both sides by \(2 \gamma T\), we get
\[\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[F(\bar{x}_t) - F_\epsilon] \]
Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Finally, by the convexity of $F$, the average iterate $\bar{x}_T = (1/T) \sum_{t=1}^T \bar{x}_t$ satisfies

$$
\mathbb{E}[F(\bar{x}_T) - F_*] 
\leq \frac{1}{T} \sum_{t=1}^T \mathbb{E}[F(\bar{x}_t) - F_*] 
\leq \frac{\|\bar{x}_0 - x_*\|^2}{2\gamma T} + \frac{D}{2\gamma} + \frac{B}{2\gamma T} + \frac{A}{\gamma T} 
\times (2A + \|\bar{x}_0 - x_*\| + \sqrt{T\textbf{D} + B}).
$$

where $\gamma \leq T^{-1/2}$ and the last equality holds due to the fact that $\sqrt{\textbf{D}/\gamma} = \mathcal{O}(T^{-1/4})$ and $\textbf{D}/\gamma = \mathcal{O}(T^{-1})$.

\begin{equation}
\mathcal{O}\left(\frac{1}{T} + \frac{1}{T^{3/2}}\right) \quad (27)
\end{equation}

V. SIMULATION

In this section, we apply the proposed algorithm DPG-RR to optimize the black-box binary classification problem [44], which is to find the optimal predictor $x \in \mathbb{R}^n$ by solving

$$
\min_{x \in \mathbb{R}^n} F(x), \quad F(x) = \frac{1}{m} \sum_{j=1}^m f_j(x) + \lambda_1 \|x\|_1
$$

where $f_j(x) \triangleq \sum_{i=1}^n \ln(1 + \exp(-l_{j,i}(a_{j,i}, x)))$, $a_{j,i} \in \mathbb{R}^d$ is the feature vector of the $i$th local sample of agent $j$, $l_{j,i} \in [-1, 1]$ is the classification value of the $i$th local sample of agent $j$, and $\{a_{j,i}, l_{j,i}\}_{i=1}^n$ denotes the set of local training samples of agent $j$. In this experiment, we use the publicly available real datasets a9a and w8a. In addition, in our setting, there are $m = 10$ agents and $\lambda_1 = 5 \times 10^{-4}$. All comparative algorithms are initialized with the same value. Experiment codes and the adjacent matrices of ten-agent time-varying networks are provided at https://github.com/managerjiang/Dis-Prox-RR.

1) Time-Invariant Networks: For comparison, we apply some existing distributed stochastic algorithms, including GT-SAGA in [24] and DSGD in [22], and the proposed stochastic DPG-RR to solve (28) over the ten-agent connected network. Since the cost function in (28) is nonsmooth, we replace the gradients in GT-SAGA and DSGD with subgradients. The simulation results for a9a and w8a datasets are shown in Figs. 1 and 2. Figs. 1(b) and 2(b) show that the trajectories of objective function $F(x)$ generated by all algorithms converge quickly to the optimum $F_*$. To measure the consensus performance, we define one quantity $D(x)$

$$
D(x) \triangleq \sum_{i=1}^n \ln(1 + \exp(-l_{i,1}(a_{i,1}, x)))
$$

is an expansion of $\sum_{i=1}^n \ln(1 + \exp(-l_{i,1}(a_{i,1}, x)))$, $a_{i,1} \in \mathbb{R}^d$ is the feature vector of the $i$th local sample of agent $j$, $l_{i,1} \in [-1, 1]$ is the classification value of the $i$th local sample of agent $j$, and $\{a_{i,1}, l_{i,1}\}_{i=1}^n$ denotes the set of local training samples of agent $j$. In this experiment, we use the publicly available real datasets a9a and w8a. In addition, in our setting, there are $m = 10$ agents and $\lambda_1 = 5 \times 10^{-4}$. All comparative algorithms are initialized with the same value. Experiment codes and the adjacent matrices of ten-agent time-varying networks are provided at https://github.com/managerjiang/Dis-Prox-RR.

$^4$a9a and w8a are available in the website www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

$^5$D(x) is an expansion of $\sum_{i=1}^n \ln(1 + \exp(-l_{i,1}(a_{i,1}, x)))$, $a_{i,1} \in \mathbb{R}^d$ is the feature vector of the $i$th local sample of agent $j$, $l_{i,1} \in [-1, 1]$ is the classification value of the $i$th local sample of agent $j$, and $\{a_{i,1}, l_{i,1}\}_{i=1}^n$ denotes the set of local training samples of agent $j$. In this experiment, we use the publicly available real datasets a9a and w8a. In addition, in our setting, there are $m = 10$ agents and $\lambda_1 = 5 \times 10^{-4}$. All comparative algorithms are initialized with the same value. Experiment codes and the adjacent matrices of ten-agent time-varying networks are provided at https://github.com/managerjiang/Dis-Prox-RR.
TABLE I
RUNNING TIME OF DIFFERENT ALGORITHMS

| Algorithms | a9a (s) | w8a (s) |
|------------|--------|--------|
| DPG-RR     | 25.21  | 140.89 |
| Prox-G     | 55.50  | 161.44 |
| NEXT       | 80.28  | 264.78 |
| DGM        | 30.91  | 120.51 |

as $D(x) = \sum_{i=1}^{n} x_i' \sum_{j=1}^{n} a_{ij}(x_i - x_j)$, where $a_{ij}$ is the $(i, j)$th element of one doubly stochastic adjacent matrix. If all local variable estimates achieve consensus, then $D(x) = 0$. Figs. 1(a) and 2(a) represent that local variables generated by all algorithms achieve consensus. In addition, it is seen that the proposed algorithm DPG-RR owns a better consensus performance than GT-SAGA and DSGD, which verifies the effectiveness of the multistep consensus mapping in DPG-RR. Although all comparative algorithms have comparable convergence rates in practice, Lian et al. [22] and Xin et al. [24] do not provide the theoretical convergence analysis of GT-SAGA and DSGD for non-smooth optimization. In addition, these algorithms are not applicable for time-varying multiagent networks; hence, we further compare our proposed algorithm with some recent distributed deterministic algorithms for non-smooth optimization over time-varying networks.

2) Time-Varying Networks: We compare DPG-RR with the distributed deterministic proximal algorithm Prox-G in [28], the algorithm NEXT in [45], and the distributed subgradient method (DGM) in [7] to solve (28) over the time-varying graphs. DPG-RR and Prox-G adopt the same constant step size; NEXT and DGM have diminishing step sizes. All distributed algorithms are applied over periodically time-varying ten-agent networks satisfying Assumption 3 to solve the problem (28), which are shown in Fig. 3.

The simulation results for a9a and w8a datasets are shown in Figs. 4 and 5, respectively. It is seen from Figs. 4(a) and 5(a) that the trajectories generated by DPG-RR, Prox-G, NEXT, and DGM all converge to zero, implying that local variable estimates generated by all algorithms achieve consensus. In addition, the DPG-RR shows a better consensus numerical performance than NEXT and DGM algorithms. From Figs. 4(b) and 5(b), we can see that DPG-RR converges.
These numerical results are consistent with the theoretical algorithm shows a comparable convergence rate with DPG-

classical techniques. DPG-RR avoids the computation of full local gradients in Prox-G. The running time of different algorithms is shown in Table I. It is observed that the proposed stochastic DPG-RR generally costs less running time than the deterministic algorithms Prox-G and NEXT. To show the advantages of the RR sampling procedure, we replace the RR step of DPG-RR with a deterministic incre-
smooth vector, i.e., \( \bar{X} \). Then, for all \( i \in \{1, \ldots, n\} \), and let \( X_i \) be sampled uniformly without replacement from \( \{X_1, \ldots, X_n\} \) and \( \bar{X} \) be their average. Then, the sample average and variance are given by

\[
\mathbb{E}[\bar{X}_x] = \bar{X}, \quad \mathbb{E}[\|\bar{X}_x - \bar{X}\|^2] = \frac{n-k}{k(n-1)} \sigma^2.
\]

Finally, we provide a lemma on the boundness of one special nonnegative sequence, which is vital for the convergence analysis.

Lemma 19 [9, Lemma 1]: Assume that the nonnegative sequence \( \{u_T\} \) satisfies the following recursion for all \( k \geq 1 \)

\[
u_T \leq S_T + \sum_{i=1}^{T} \lambda_i u_i \quad (31)
\]

with \( \{S_T\} \) an increasing sequence, \( S_0 \geq u_0^2 \), and \( \lambda_i \geq 0 \) for all \( t \). Then, for all \( T \geq 1 \)

\[
u_T \leq \frac{1}{2} \sum_{i=1}^{T} \lambda_i + \left(S_T + \left(\frac{1}{2} \sum_{i=1}^{T} \lambda_i\right)^2\right)^{\frac{1}{2}}. \quad (32)
\]

B. Proof of Proposition 11

Proof: By taking the average of (7) over \( m \) agents

\[
\bar{x}_i^{t+1} = \bar{x}_i^t - \gamma \left( \frac{1}{m} \sum_{j=1}^{m} \nabla f_j(x_i^t) + \xi_i^t \right)
\]

\[
= \bar{x}_i^t - \gamma \left( \nabla f_{r_i^t}(\bar{x}_i^t) + e_i^t \right) \quad (33)
\]

A. Some Vital Lemmas

First, the following lemma characterizes the \( \varepsilon_t \)-subdifferential of nonsmooth function \( \phi \) at \( \bar{x}_i \) and \( \partial_{\bar{x}_i} \phi(\bar{x}_i) \).

Lemma 16 [9, Lemma 2]: If \( \bar{x}_i \) is an \( \varepsilon_t \)-optimal solution to (2) in the sense of (4) with \( y = \bar{x}_i - \gamma (\bar{g}_i + \varepsilon_t) \), then there exists \( p_i \in \mathbb{R}^d \) such that

\[
\|p_i\| \leq \sqrt{2\gamma \varepsilon_t} \quad (29)
\]

and \( (1/\gamma) (\bar{x}_{i-1} - \bar{x}_i - \gamma \bar{g}_i - \varepsilon_t - p_i) \in \partial_{\bar{x}_i} \phi(\bar{x}_i) \).

The next lemma shows that polynomial-geometric sequences are summable, which is vital for the analysis of error sequences introduced by transformation.

Lemma 17 [28, Proposition 3]: Let \( \zeta \in (0, 1) \), and let

\[
P_{k-N} = \{c_N k^N + \cdots + c_1 k + c_0 \mid c_j \in \mathbb{R}, j = 0, \ldots, N\}
\]

denote the set of all \( N \)th order polynomials of \( k \), where \( N \in \mathbb{N} \). Then, for every polynomial \( p_k \in P_{k-N} \)

\[
\sum_{k=1}^{\infty} p_k \zeta^k < \infty.
\]

The result of this lemma for \( P_{k-N} = \{k^N\} \) will be particularly useful for the analysis in the following. Hence, we define

\[
S_N \triangleq \sum_{k=1}^{\infty} k^N \zeta^k < \infty. \quad (30)
\]

The following lemma characterizes the variance of sampling without replacement, which is a key ingredient in our convergence result of DPG-RR.

Lemma 18 [36, Lemma 1]: Let \( X_1, \ldots, X_n \in \mathbb{R}^d \) be fixed vectors, \( \bar{X} \triangleq (1/n) \sum_{i=1}^{n} X_i \) be their average, and \( \sigma^2 \triangleq (1/n) \sum_{i=1}^{n} \|X_i - \bar{X}\|^2 \) be the population variance. Fix any \( k \in \{1, \ldots, n\} \); let \( X_{\pi_k}, \ldots, X_{\bar{X}} \) be sampled uniformly without replacement from \( \{X_1, \ldots, X_n\} \) and \( \bar{X} \) be their average. Then, the sample average and variance are given by

\[
\mathbb{E}[\bar{X}_x] = \bar{X}, \quad \mathbb{E}[\|\bar{X}_x - \bar{X}\|^2] = \frac{n-k}{k(n-1)} \sigma^2.
\]

VI. CONCLUSION

Making use of RR, this article has developed one distributed stochastic proximal-gradient algorithm for solving large-scale nonsmooth convex optimization over time-varying multigraphs. The proposed algorithm operates only one single proximal evaluation at each epoch, owning significant advantages in scenarios where the proximal mapping is computationally expensive. In addition, the proposed algorithm owns constant step sizes, overcoming the degraded performance of most stochastic algorithms with diminishing step sizes. One future research direction is to improve the convergence rate of DPG-RR by some accelerated methods, such as Nesterov’s acceleration technique or variance reduction technique. Another one is to extend the proposed algorithm for distributed nonsmooth nonconvex optimization, which is applicable to deep neural network models.

APPENDIX

A. Some Vital Lemmas

First, the following lemma characterizes the \( \varepsilon_t \)-subdifferential of nonsmooth function \( \phi \) at \( \bar{x}_i \) and \( \partial_{\bar{x}_i} \phi(\bar{x}_i) \).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where \( e^i_t = (1/m) \sum_{j=1}^m (\nabla f_j, x^i_t) - \nabla f_j, (x^i_t) \) and \( f^i_t(x) = (1/m) \sum_{j=1}^m f_j, x^i_t(x) \). Because of the Lipschitz-continuity of the gradient of \( f_j, x^i_t \), \( \| e^i_t \| \) satisfies

\[
\| e^i_t \| \leq \frac{L}{m} \sum_{j=1}^m \| x^i_t - x^j_t \|. \tag{34}
\]

Recall that \( z_{t+1} = \text{prox}_{\phi, \gamma}(\bar{v}_t) = \arg\min_i \{ \phi(x) + (1/2\gamma)\| x - \bar{v}_t \|^2 \} \), which is the result of the exact centralized proximal step. Then, we relate \( z_{t+1} \) and \( \bar{x}_{t+1} \) by formulating the latter as an inexact proximal step with error \( e_{t+1} \). A simple algebraic expansion gives

\[
\phi(\bar{x}_{t+1}) + \frac{1}{2\gamma} \| \bar{x}_{t+1} - \bar{v}_t \|^2 \\
\leq \phi(z_{t+1}) + G_\phi \| \bar{x}_{t+1} - z_{t+1} \| + \frac{1}{2\gamma} \times \\
\left( \| z_{t+1} - \bar{v}_t \|^2 + 2(\bar{x}_{t+1} - \bar{v}_t, \bar{x}_{t+1} - z_{t+1}) + \| \bar{x}_{t+1} - z_{t+1} \|^2 \right) \\
\leq \min_{z \in \mathbb{R}^n} \left\{ \phi(z) + \frac{1}{2\gamma} \| z - \bar{v}_t \|^2 \right\} + \frac{1}{2\gamma} \| \bar{x}_{t+1} - z_{t+1} \|^2 \\
+ \| z_{t+1} - \bar{v}_t \| (G_\phi + \frac{1}{\gamma} \| z_{t+1} - \bar{v}_t \|)
\]

where we have used the fact that \( \phi(\bar{x}_{t+1}) + (1/2\gamma)\| \bar{x}_{t+1} - \bar{v}_t \|^2 = \min_z \{ \phi(z) + (1/2\gamma)\| z - \bar{v}_t \|^2 \} \) and \( \| z_{t+1} - \bar{v}_t, \bar{x}_{t+1} - z_{t+1} \| = \| z_{t+1} - \bar{v}_t \|\| \bar{x}_{t+1} - z_{t+1} \| \) in the last inequality.

Therefore, we can write

\[
\bar{x}_{t+1} = \text{prox}_{G_\phi, \gamma}(\bar{v}_t) \tag{35}
\]

where \( e_{t+1} = \| \bar{x}_{t+1} - z_{t+1} \| (G_\phi + (1/\gamma))\| z_{t+1} - \bar{v}_t \| + (1/2\gamma)\| \bar{x}_{t+1} - z_{t+1} \|^2 \).

By definition, \( z_{t+1} \) satisfies \( (1/\gamma)(\bar{v}_t - z_{t+1}) \in \partial \phi(z_{t+1}) \), and therefore, its norm is bounded by \( e_{t+1} \).

As a result

\[
e_{t+1} \leq 2G_\phi \| \bar{x}_{t+1} - z_{t+1} \| + \frac{1}{2\gamma} \| \bar{x}_{t+1} - z_{t+1} \|^2. \tag{36}
\]

Combined with the nonexpansiveness of the proximal operator

\[
\| \bar{x}_{t+1} - z_{t+1} \| \leq \frac{1}{m} \sum_{j=1}^m \| \text{prox}_{\phi, \gamma}(v_{j,t}) - \text{prox}_{\phi, \gamma}(\bar{v}_t) \|
\]

\[
\leq \frac{1}{m} \sum_{j=1}^m \| v_{j,t} - \bar{v}_t \|. \tag{37}
\]

Finally, substituting (37) to (36), we obtain (12b). \(\square\)

### C. Proof of Proposition 12

We first define two useful quantities \( \Gamma \triangleq 2((1 + \eta^{-(m-1)B})(1 - \eta^{-(m-1)B})) \) and \( \Xi \triangleq (1 - \eta^{-(m-1)B})(1/(1-(m-1)B)) \). Then, we provide some properties of the local variables generated by the proposed algorithm in the following lemma.

**Lemma 20:** Under Assumptions 2 and 3, for each iteration \( t \geq 2 \)

\[
\sum_{j=1}^m \| x^0_{j,t} \| \leq \sum_{j=1}^m \| x^0_{j,t-1} \| + m\gamma (G_\phi + nG_f) \tag{38}
\]

\[
\sum_{j=1}^m \| x_{j,t+1} - x_{j,t} \| \leq m\gamma \sum_{j=1}^m (\gamma + nG_f) \tag{39}
\]

\[
\sum_{j=1}^m \| x_{j,t} - \bar{x}_t \| \leq m\gamma \sum_{j=1}^m \| x^0_{j,t-1} \|. \tag{40}
\]

**Proof:** By (10), there exists \( z_{j,t+1} \in \partial \phi(x_{j,t+1}) \) such that

\[
x_{j,t+1} = x_{j,t} - \gamma z_{j,t+1}. \tag{41}
\]

Since function \( \phi \) has bounded subgradients

\[
\| x_{j,t+1} - x_{j,t} \| \leq \gamma G_\phi. \tag{42}
\]

1) By the algorithm 1, we have

\[
x^0_{j,t} = x^0_{j,t} - \gamma \sum_{i=0}^{n-1} \nabla f_j, x^i_{j,t}. \tag{43}
\]

Taking the norm of the above equality and summing over \( j \)

\[
\sum_{j=1}^m \| x^0_{j,t} \| = \sum_{j=1}^m \| x^0_{j,t} \| - \gamma \sum_{i=0}^{n-1} \nabla f_j, x^i_{j,t} \| \leq m\gamma \sum_{j=1}^m \| x^0_{j,t} \| + mnG_f. \tag{44}
\]

By (42), \( \| x_{j,t+1} \| - \| v_{j,t} \| \leq \gamma G_\phi \) holds. Since \( v_{j,t} \) is a convex combination of \( \{ x^0_{j,t} \}_{t=1}^m \) and \( l = 1 \)

\[
\sum_{j=1}^m \| v_{j,t} \| \leq \sum_{j=1}^m \| x^0_{j,t} \|. \tag{45}
\]

Then, substituting the fact that \( \| x_{j,t+1} \| - \| v_{j,t} \| \leq \gamma G_\phi \) and (44) to (43), we obtain

\[
\sum_{j=1}^m \| x^0_{j,t} \| \leq \sum_{j=1}^m \| x^0_{j,t-1} \| + m\gamma (G_\phi + nG_f). \tag{46}
\]

2) By (41) and the proposed algorithm

\[
x_{j,t+1} = x_{j,t} - \gamma z_{j,t+1} \tag{47}
\]

\[
= \sum_{l=1}^m \lambda_{j,l} x^0_{j,l} - \gamma z_{j,t+1} \tag{48}
\]

\[
= \sum_{l=1}^m \lambda_{j,l} \left( x^0_{j,l} - \gamma \sum_{i=0}^{n-1} \nabla f_j, x^i_{j,l} \right) - \gamma z_{j,t+1}. \tag{49}
\]

Then, subtracting \( x_{j,t} \) from both sides of (45) and taking norm, we obtain

\[
\sum_{j=1}^m \| x_{j,t+1} - x_{j,t} \| \leq m\gamma \sum_{j=1}^m \| x^0_{j,t-1} \| + m\gamma (G_\phi + nG_f) \tag{50}
\]

\[
= \sum_{j=1}^m \| x_{j,t} - x_{j,t} \| + m\gamma (G_\phi + nG_f). \tag{51}
\]
For the first term in (46), by the nonexpansiveness of the proximal operator
\[\|x_{i,t} - x_{j,t}\| \leq \|v_{t-1} - v_{j,t-1}\|.\]
In addition, by [27, Proposition 1], the bound of the distance between iterates \(v_{j,t}\) and \(\bar{v}_t\) satisfies
\[\|v_{j,t} - \bar{v}_t\| = \left\| \sum_{l=1}^{m} \lambda_{j,t} x_{l,t}^n - \frac{1}{m} x_{l,t}^n \right\| \leq \sum_{l=1}^{m} \lambda_{j,t} - \frac{1}{m} \left\| x_{l,t}^n \right\| \leq \Gamma \Xi \sum_{l=1}^{m} \|x_{l,t}^n\|. \tag{47}\]

Then, by transformation
\[\sum_{j=1}^{m} \sum_{l=1}^{m} \lambda_{j,t} \|v_{t-1} - v_{j,t-1}\| \leq \sum_{j=1}^{m} \sum_{l=1}^{m} \lambda_{j,t} (\|v_{t-1} - \bar{v}_t\| + \|v_{j,t-1} - \bar{v}_t\|) \leq 2m \Gamma \Xi^{t-1} \sum_{l=1}^{m} \|x_{l,t-1}\|. \tag{48}\]
Substituting (48) into (46)
\[\sum_{j=1}^{m} \|x_{j,t+1} - x_{j,t}\| \leq 2m \Gamma \Xi^{t-1} \sum_{l=1}^{m} \|x_{l,t-1}\| + m \gamma (G \phi + n G_f).\]

3) By the definition of \(\bar{\bar{x}}_t = (1/m) \sum_{p=1}^{m} x_{p,t} + \sum_{j=1}^{m} \|x_{j,t} - \bar{\bar{x}}_t\|\) satisfies
\[\sum_{j=1}^{m} \left\| x_{j,t} - \frac{1}{m} \sum_{p=1}^{m} x_{p,t} \right\| = \sum_{j=1}^{m} \left\| x_{j,t} - x_{p,t} \right\| \leq \frac{1}{m} \sum_{p=1}^{m} \left\| v_{j,t-1} - v_{p,t-1} \right\| \leq \frac{1}{m} \sum_{j=1}^{m} \sum_{p=1}^{m} (\|v_{j,t-1} - \bar{v}_t\| + \|v_{p,t-1} - \bar{v}_t\|) \leq 2m \Gamma \Xi \sum_{l=1}^{m} \|x_{l,t-1}\| \tag{47}\]
where the first inequality follows from Proposition 1, and the last inequality follows from (47).

The following lemma proves that the sequence \(\sum_{j=1}^{m} \|x_{j,t}\|^2\) is upper bounded by a polynomial-geometric sequence, which is vital in the discussions of the summability of the error sequences.

\[\text{Lemma 21:} \text{Under Assumptions 2 and 3, there exist nonnegative scalars } C_q = C_q(x_{1,1}^n, \ldots, x_{m,1}^n), C_1^q = C_1^q(m, \Gamma, C_q, C_q^2), \text{ and } C_q^2 = C_q^2(m, \Gamma, C_q, C_q^2, C_q) \text{ such that, for iteration } t \geq 1 \]
\[\sum_{j=1}^{m} \|x_{j,t}\|^2 \leq C_q + C_1^q t + C_q^2 t^2.\]

\[\text{Proof:} \text{This proof is inspired by [28, Lemma 1]. We proceed by induction on } t. \text{ First, we show that the result holds for } t = 1 \text{ by choosing } C_q = \sum_{j=1}^{m} \|x_{j,1}\|. \text{ It suffices to show that } \sum_{j=1}^{m} \|x_{j,t}\|^2 \text{ is bounded given the initial points } x_{j,0}.\]
Indeed, by (38)
\[\sum_{j=1}^{m} \|x_{j,1}\|^2 \leq \sum_{j=1}^{m} \|x_{j,0}\|^2 + m \gamma (G \phi + n G_f) < \infty.\]
Therefore, \(C_q = \sum_{j=1}^{m} \|x_{j,1}\|^2 < \infty\) is a valid choice.

Now, suppose the result in Lemma 21 holds for some positive integer \(t \geq 1\). We show that it also holds for \(t + 1\). We transform (38) in Lemma 20 to
\[\sum_{j=1}^{m} \|x_{j,t+1} - x_{j,t}\|^2 \leq 2m \Gamma \sum_{p=1}^{m} \Xi^p \sum_{j=1}^{m} \|x_{j,p}\|^2 + (t - 1) m \gamma (G \phi + n G_f). \tag{50}\]
Then, substituting the induction hypothesis for \(t \) into (50), we have
\[\sum_{j=1}^{m} \|x_{j,t+1} - x_{j,t}\| \leq 2m \Gamma \sum_{p=1}^{m} \Xi^p (C_q + C_1^q t + C_q^2 t^2) + (t - 1) m \gamma (G \phi + n G_f).\]
By Lemma 17 and (30), there exist constants \(S_{\Xi_1}^\Xi, S_{\Xi_2}^\Xi, \) and \(S_{\Xi_3}^\Xi\) such that
\[\sum_{p=1}^{\infty} \Xi^p (C_q + C_1^q t + C_q^2 t^2) \leq C_q S_{\Xi_1}^\Xi + C_1^q S_{\Xi_2}^\Xi + C_q^2 S_{\Xi_3}^\Xi.\]
By induction hypothesis and (49)
\[\sum_{j=1}^{m} \|x_{j,t+1}\|^2 \leq C_q + C_1^q t + C_q^2 t^2 + m \gamma (G \phi + n G_f) + 2m \Gamma (C_q S_{\Xi_1}^\Xi + C_1^q S_{\Xi_2}^\Xi + C_q^2 S_{\Xi_3}^\Xi). \tag{51}\]
Take the coefficients \(C_q, C_1^q, \) and \(C_q^2\) as
\[C_q = \sum_{j=1}^{m} \|x_{j,1}\|^2, \quad C_1^q = \frac{2m \Gamma C_q S_{\Xi_1}^\Xi + (2m \Gamma S_{\Xi_2}^\Xi - 1) C_q^2}{2m \Gamma S_{\Xi_3}^\Xi - 1}, \quad C_q^2 = \frac{C_q}{C_q^2}.\]
\[ C_q^2 = \frac{\gamma m}{2} (G_{\phi} + nG_f). \]

Then, comparing coefficients, we see that the right-hand side of (51) has an upper bound \( C_q + C_q^4(t+1) + C_q^2 (t+1)^2 \), which implies that the induction hypothesis holds for \( t+1 \). \( \square \)

**Proof of Proposition 12:**

Proof: By the results in Lemmas 20 and 21, we develop the theoretical proof for Proposition 12 as follows.

1) By (40) in Lemma 20

\[
\|x_{j,t} - \bar{x}_t\| \leq 2\Gamma \Xi^{-1} \sum_{i=1}^{m} \|x_{i,t-1}\|. \tag{52}
\]

Then, by Lemma 21, and (12a) and (52), there exist \( C_q \), \( C_q^4 \), \( C_q^2 > 0 \) such that

\[
\|e_t^j\| \leq 2\Gamma \Xi^{-1} (C_q + C_q^4 (t+1) + C_q^2 (t+1)^2) + \gamma C_0
\]

where \( C_0 = 2\Gamma n G_f \) and \( b_{x,t} \) is a polynomial-geometric sequence.

2) It follows from (12b) and (47), and Lemma 21 that (54), as shown at the top of the next page. Using the fact that \((a+b)^{1/2} \leq \sqrt{a} + \sqrt{b}\) for all nonnegative real numbers \( a \) and \( b \) (55), as shown at the top of the next page, where \( C(t) = (C_q + C_q^4 t + C_q^2 t^2) \).

\[ \]

**D. Proof of Lemma 15**

Proof: By Lemma 16 and (11), we have

\[
\tilde{x}_t - \gamma \tilde{g}_t - \tilde{x}_{t+1} = -\gamma \sum_{i=0}^{n-1} e^i_t = -\gamma \tilde{d}_{t+1} \tag{56}
\]

where \( \|p_{t+1}\| \leq (2\gamma e_{t+1})^{1/2}, \tilde{d}_{t+1} \in \tilde{c}_{t+1}, \tilde{g}(\tilde{x}_{t+1}), \tilde{g}_t \triangleq \sum_{i=0}^{n-1} \nabla f(x^i_{t+1}), \text{ and } f(x) = (1/m) \sum_{i=0}^{n-1} f_j(x_i). \]

In addition, by (33), \( \tilde{x}_t^k = \tilde{x}_t - \gamma \sum_{i=0}^{k-1} \nabla f(x^i_t) - \gamma \sum_{i=0}^{k-1} e^i_t \).

Then, combining (56) with (33)

\[
\tilde{x}_t^k - \tilde{x}_{t+1} = \tilde{x}_t - \gamma \sum_{i=0}^{n-1} \nabla f(x^i_t) - \gamma \sum_{i=0}^{k-1} e^i_t
\]

\[
- \left( \tilde{x}_t - \gamma \sum_{i=0}^{n-1} \nabla f(x^i_t) - \gamma \sum_{i=0}^{k-1} e^i_t - p_{t+1} - \gamma \tilde{d}_{t+1} \right)
\]

\[
= \gamma \sum_{i=k}^{n-1} \nabla f(x^i_t) + \gamma \sum_{i=0}^{k-1} e^i_t + p_{t+1} + \gamma \tilde{d}_{t+1}.
\]

Taking the norm of \( \tilde{x}_t^k - \tilde{x}_{t+1} \), we have

\[
\left\| \tilde{x}_t^k - \tilde{x}_{t+1} \right\|^2 \leq 6\gamma^2 \left\| \sum_{i=k}^{n-1} \nabla f(x^i_t) \right\|^2 + 6\gamma^2 \left\| \tilde{d}_{t+1} \right\|^2
\]

\[
+ 2 \left\| p_{t+1} \right\|^2
\]

\[
\leq 6\gamma^2 \left( 2 \sum_{i=k}^{n-1} \left( \left( \tilde{x}_t^i + \tilde{d}_{t+1} \right) + \left( \tilde{x}_t^i + p_{t+1} + \gamma \tilde{d}_{t+1} \right) \right) \right)^2
\]

\[\]

Next, consider the term \( \sum_{i=k}^{n-1} \nabla f(x^i_t)^2 \) in (57). By Lemma 18 with \( \tilde{X}_t = (1/(n-k)) \sum_{i=k}^{n-1} \nabla f(x^i_t) \) and \( \tilde{X}_t = (1/n) \tilde{X}_t \)

\[\]

}\]

where \( \sigma^2 \triangleq (1/n) \sum_{i=0}^{n} \|\nabla f(x^i_t) - (1/n) \tilde{X}_t \|^2 \) is the population variance at the optimum \( x^* \). Summing this over \( k \) from 0 to \( n-1 \)

\[
\leq \left\| \sum_{i=k}^{n-1} \nabla f(x^i_t) \right\|^2
\]

\[
= (n-k)^2 \left\| \sum_{i=k}^{n-1} \nabla f(x^i_t) \right\|^2
\]

\[
= (n-k)^2 \left[ \sum_{i=k}^{n-1} \nabla f(x^i_t) \right]
\]

\[
= (n-k)^2 \left[ \left\| \nabla f(x^* \right\|^2 + \left\| \tilde{X}_t - \tilde{X}_t^2 \right\|^2 \right]
\]

\[
= (n-k)^2 \left[ \frac{1}{n} \left\| \nabla f(x^* \right\|^2 + \frac{k}{n-k} \frac{\sigma^2}{n-k} \right]
\]

\[\]

where the inequality holds due to \( n \geq 2, n+1 < 2n \), and Assumption 2.4.

Thus, summing (57) over \( k \) gives

\[
\sum_{k=0}^{n-1} \left\| \tilde{x}_t^k - \tilde{x}_{t+1} \right\|^2
\]

\[\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\[ \epsilon_{t+1} \leq 2G_0 \Gamma \sum_{i=1}^{m} \left( \gamma_t^2 \| x_{t,i}^n \|^2 + \frac{1}{2\gamma_t} \left( \Gamma \Xi_t^m \sum_{i=1}^{m} \| x_{t,i}^n \|^2 \right)^2 \right) \leq 2G_0 \Gamma \left( C_q + C_q^1 t + C_q^2 t^2 \right) + \frac{1}{2\gamma_t} \left( \Gamma \Xi_t^m \left( C_q + C_q^1 t + C_q^2 t^2 \right)^2 \right) \]

where we use (58) and Assumption 2.3). □

REFERENCES

[1] F. Fützeleer and J. Malick, “Nonsmoothness in machine learning: Specific structure, proximal identification, and applications,” Set-Valued Varia-

[2] W. Tao, Z. Pan, G. Wu, and Q. Tao, “The strength of Nesterov’s extrapola-

[3] W. Zhong and J. Kwok, “Accelerated stochastic gradient method for

[4] M. Schmidt, N. L. Roux, and F. Bach, “Convergence rates of inexact

[5] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-

[6] K. Wang, Z. Fu, Q. Xu, D. Chen, L. Wang, and W. Yu, “Distributed fixed

[7] A. Astorino and A. Fuduli, “Nonsmooth optimization techniques for

[8] A. H. Zaini and L. Xie, “Distributed drone traffic coordination using

[9] M. Schmidt, N. L. Roux, and F. Bach, “Convergence rates of inexact

[10] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu,

[11] J. Lei, H.-F. Chen, and H.-T. Fang, “Asymptotic properties of primal-

[12] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-

[13] W. Zhong and J. Kwok, “Accelerated stochastic gradient method for

[14] F. Iutzeler and J. Malick, “Nonsmoothness in machine learning: Specific

[15] L. Vandenberghe, Lecture Notes for EE 236C, UCLA. Los Angeles, CA,

[16] F. Sattler, K.-R. Müller, and W. Samek, “Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[17] W. Wu, X. Jing, W. Du, and G. Chen, “Learning dynamics of gradient descent optimization in deep neural networks,” Sci. China Inf. Sci., vol. 64, no. 5, May 2021, Art. no. 150102.

[18] W. Li, Z. Wu, T. Chen, L. Li, and Q. Ling, “Communication-cen-

[19] O. Shamir and T. Zhang, “Stochastic gradient descent for non-smooth

[20] P. Bianchi, G. Forti, and W. Haemem, “Performance of a distributed stochastic approximation algorithm,” IEEE Trans. Inf. Theory, vol. 59, no. 11, pp. 7407–7418, Nov. 2013.

[21] J. Lei, H.-F. Chen, and H.-T. Fang, “Asymptotic properties of prinal-

[22] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu,

[23] M. Gürbüzbalaban, A. Ozdaglar, and P. A. Parrilo, “Why random

[24] F. Sattler, K.-R. Müller, and W. Samek, “Clustered federated learn-

[25] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, “Push-SAGA: A

[26] X.-F. Wang, A. R. Teel, K.-Z. Liu, and X.-M. Sun, “Stability

[27] Z. Li, B. Liu, and Z. Ding, “Consensus-based cooperative algorithms for training over distributed data sets using stochastic gradients,” IEEE Trans. Neural Netw. Learn. Syst., early access, Apr. 16, 2021, doi: 10.1109/TNNLS.2021.3071058.

[28] R. Xin, U. A. Khan, and S. Kar, “Variance-reduced decentralized stochastic optimization with accelerated convergence,” IEEE Trans. Signal Process., vol. 68, pp. 6255–6271, 2020, doi: 10.1109/TSP.2020.3031071.

[29] M. I. Qureshi, R. Xin, S. Kar, and U. A. Khan, “Push-SAGA: A decentralized stochastic algorithm with variance reduction over directed graphs,” IEEE Control Syst. Lett., vol. 6, pp. 1202–1207, 2022.

[30] F. Sattler, K.-R. Müller, and W. Samek, “Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[31] W. Wu, X. Jing, W. Du, and G. Chen, “Learning dynamics of gradient descent optimization in deep neural networks,” Sci. China Inf. Sci., vol. 64, no. 5, May 2021, Art. no. 150102.
[32] A. Rakhlin, O. Shamir, and K. Sridharan, “Making gradient descent optimal for strongly convex stochastic optimization,” in Proc. 29th Int. Conf. Int. Conf. Mach. Learn. Madison, WI, USA: Omnipress, 2012, pp. 1571–1578.

[33] Y. Drori and O. Shamir, “The complexity of finding stationary points with stochastic gradient descent,” in Proc. 37th Int. Conf. Mach. Learn. (Proceedings of Machine Learning Research), vol. 119. Vienna, Austria: PMLR, Jul. 2020, pp. 2658–2667. [Online]. Available: https://proceedings.mlr.press/v119/drori20a.html

[34] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for non-differentiable optimization,” SIAM J. Optim., vol. 12, no. 1, pp. 109–138, 2001.

[35] D. P. Bertsekas, Optimization for Machine Learning. Cambridge, MA, USA: MIT Press, 2011, ch. 4.

[36] K. Mishchenko, A. Khaled, and P. Richtarik, “Random reshuffling: Simple analysis with vast improvements,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33. Red Hook, NY, USA: Curran Associates, 2020, pp. 17309–17320. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/6cdec690ccbc2fb4499ccee236117111cd4-Paper.pdf

[37] M. Qi, W. Chen, Y. Wang, Z.-M. Ma, and T.-Y. Liu, “Convergence analysis of distributed stochastic gradient descent with shuffling,” Neurocomputing, vol. 337, pp. 46–57, Apr. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219300578

[38] K. Mishchenko, A. Khaled, and P. Richtarik, “Proximal and federated random reshuffling,” in Proc. 39th Int. Conf. Mach. Learn., in Proceedings of Machine Learning Research, vol. 162. Baltimore, MD, USA: PMLR, Jul. 2022, pp. 15718–15749.

[39] L. Xiao and T. Zhang, “A proximal stochastic gradient method with progressive variance reduction,” SIAM J. Optim., vol. 24, no. 4, pp. 2057–2075, 2014, doi: 10.1137/140961791.

[40] Z. Li and J. Li, “A simple proximal stochastic gradient method for nonsmooth nonconvex optimization,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31. Red Hook, NY, USA: Curran Associates, 2018, pp. 1–11. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/e727fa59d5defc3ebf5d395017662132-Paper.pdf

[41] N. D. Vanli, M. Gursoy, and A. Oradgar, “Global convergence rate of incremental aggregated gradient methods for nonsmooth problems,” in Proc. IEEE 55th Conf. Decis. Control (CDC), Dec. 2016, pp. 173–178.

[42] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” SIAM J. Imaging Sci., vol. 2, no. 1, pp. 183–202, 2009.

[43] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of FedAvg on Non-IID data,” in Proc. Int. Conf. Learn. Represent., 2020, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=HkxAnViDS

[44] S. Abu-Mostafa, M. Magdon-Ismail, and H. Lin, Learning From Data. New York, NY, USA: Cambridge University Press, 2020, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=HkxAnViDS

[45] P. Di Lorenzo and G. Scutari, “NEXT: In-network nonconvex optimization.” IEEE Trans. Signal Inf. Process. Netw., vol. 2, no. 2, pp. 120–136, Jun. 2016.

Xia Jiang received the bachelor’s degree in control science and engineering from Shandong University, Jinan, China, in 2017. She is currently pursuing the Ph.D. degree in control science and engineering with the School of Automation, Beijing Institute of Technology, Beijing, China, and the Beijing Institute of Technology Chongqing Innovation Center, Chongqing, China. She is visiting the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where she is currently a Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Her current research interests include distributed optimization, machine learning, and distributed computation of semidefinite problems.

Xianlin Zeng (Member, IEEE) received the B.S. and M.S. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2009 and 2011, respectively, and the Ph.D. degree in mechanical engineering from Texas Tech University, Lubbock, TX, USA, in 2015. He is currently an Associate Professor with the Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing, China. His current research interests include distributed optimization, distributed control, and distributed computation of network systems.

Jian Sun (Senior Member, IEEE) received the bachelor’s degree from the Department of Automation and Electric Engineering, Jilin Institute of Technology, Changchun, China, in 2001, the master’s degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences (CAS), Changchun, in 2004, and the Ph.D. degree from the Institute of Automation, CAS, Beijing, China, in 2007. He was a Research Fellow with the Faculty of Advanced Technology, University of Glamorgan, Pontypridd, U.K., from 2008 to 2009. He was a Post-Doctoral Research Fellow with the Beijing Institute of Technology, Beijing, from 2007 to 2010. In 2010, he joined the School of Automation, Beijing Institute of Technology, where he has been a Professor since 2013. Since 2019, he has also been with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing, China. His current research interests include networked control systems, time-delay systems, and the security of cyber-physical systems. Dr. Sun is also an Editorial Board Member of the Journal of Systems Science and Complexity and Acta Automatica Sinica.

Jie Chen (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from the Beijing Institute of Technology, Beijing, China, in 1986, 1996, and 2001, respectively. From 1989 to 1990, he was a Visiting Scholar with California State University, Long Beach, CA, USA. From 1996 to 1997, he was a Research Fellow with the School of Engineering, University of Birmingham, Birmingham, U.K. He is currently a Professor with the School of Automation, Beijing Institute of Technology, where he is also the Director of the Key Laboratory of Intelligent Control and Decision of Complex Systems. He is also the President of Tongji University, Shanghai, China. His research interests include complex systems, multiaagent systems, subjective optimization and decision, and constrained nonlinear control. Prof. Chen is also a fellow of the International Federation of Automatic Control (IFAC) and a member of the Chinese Academy of Engineering. He has served on the editorial boards of several journals, including the IEEE TRANSACTIONS ON CYBERNETICS, International Journal of Robust and Nonlinear Control, and Science China Information Sciences. He is also the Editor-in-Chief of Unmanned Systems and the Journal of Systems Science and Complexity.