IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 22, 2021, accepted October 6, 2021, date of publication October 14, 2021, date of current version February 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3120017

A Flexible Stochastic Multi-Agent ADMM Method for Large-Scale Distributed Optimization

LIN WU[1,2], YONGBIN WANG[1,2], AND TUO SHI[3,4]

[1]Key Laboratory of Convergent Media and Intelligent Technology, Ministry of Education, Beijing 100024, China
[2]School of Computer and Cyberspace Security, Communication University of China, Beijing 100024, China
[3]Beijing Police College, Beijing 102202, China
[4]Institute of Scientific and Technical Information of China, Beijing 100038, China

Corresponding author: Yongbin Wang (ybwang@cuc.edu.cn)

This work was supported in part by the Research on the Technology Support for Convergent Media and Service Pattern through the National Key Research and Development Program and under Grant 2019YFB1406201.

ABSTRACT While applying stochastic alternating direction method of multiplier (ADMM) methods has become enormously potential in distributed applications, improving the algorithmic flexibility can bring huge benefits. In this paper, we propose a novel stochastic optimization method based on distributed ADMM method, called Flex-SADMM. Specifically, we incorporate the variance reduced first-order information and the approximated second-order information for solving the subproblem of ADMM, which targets at the stable convergence and improving the accuracy of the search direction. Moreover, different from most ADMM based methods that require each computation node to perform the update in each iteration, we only require each computation node updates within a bounded iteration interval, this has significantly improved the flexibility. We further provide the theoretical results to guarantee the convergence of Flex-SADMM in the nonconvex optimization problems. These results show that our proposed method can successfully overcome the above challenges while the computational complexity is maintained low. In the empirical study, we have verified the effectiveness and the improved flexibility of our proposed method.

INDEX TERMS Distributed optimization, ADMM, variance reduction, Hessian approximation, flexibility.

I. INTRODUCTION

Mathematical optimization has been widely applied in many applications, e.g., computational biology [1], [2], wireless communication [3]–[7] and machine learning [8]–[10]. The stochastic optimization is based on the theories of the deterministic optimization. It can be traced back to the epochal work [11], and its main ingredient is to use a mini-batch of data points that are randomly sampled from the full dataset to approximate the exact gradient for the search direction, which is known as the stochastic gradient descent (SGD).

However, it will have large variance which leads to a slow convergence [12]. Thus, there are several methods targeting at controlling and reducing the variance during mini-batch optimization. Particularly, the variance reduction techniques [13] have been developed to solve the above issue and greatly accelerate the convergence of SGD. Exemplar algorithms are stochastic variance reduction gradient method (SVRG) [14], stochastic average gradient method (SAG) [15] and its extension SAGA [16]. In particular, SAG and SAGA utilize full gradient without direct calculation but being instead updated with the newest information at each iteration. They have the fast convergence speed and the low computation level as SGD. Furthermore, utilizing second-order information, e.g., the Hessian matrix, is more accurate and can lead to faster convergence speed. To be precise, various recursive update schemes for the approximated Hessian matrix have been proposed including symmetric rank-one (SR1) update [17]–[19] and rank-two variants such as the Davidon-Fletcher-Powell (DFP) scheme [20], [21] and the BFGS update scheme [22], [23].

In particular, [24] has studied the online BFGS and online limited memory BFGS (oLBFGS) under stochastic regime. The main ingredient is adapting the deterministic BFGS and limited memory BFGS (LBFGS) to the stochastic version using the stochastic gradient. The convergence properties of oLBFGS have been further studied under the strong convexity assumption in [25]. [26] has proposed a novel stochastic

The associate editor coordinating the review of this manuscript and approving it for publication was Fung Po Tso.

quasi-Newton method for solving the support vector machine (SVM) problems. It aims to approximate the diagonal elements via the term-by-term equalities. Since it only involves scalar multiplications - the approximated diagonal elements of the rescaling matrix, this method is quite computationally efficient. However, direct applying stochastic gradient for the Hessian approximations may bring noise, which may affect the robustness of the convergence. Hence, it is a challenge for controlling the quality of the curvature approximations in stochastic optimization.

The above optimization methods are implemented in a single computational node. However, in the big data era, solving a large-scale optimization problem in machine learning with a single computation node is generally prohibitive. This is due to the reason that the quantities of realistic data for training a machine learning model can be very large and typically range from 1TB to 1PB. As a result, distributed optimization has become a crucial technique and it has been of importance to develop algorithms that are both efficient and scalable enough to process huge datasets in a parallelized or fully decentralized fashion. The alternating direction method of multipliers (ADMM) is a natural fit for the distributed convex optimization, and in particular to large-scale machine learning problems. ADMM method is simple and its main ingredient is to decompose the problems into subproblems, which can be individually solved in parallel, thus it turns out to be a natural fit in a wide class of large-scale applications, e.g., machine learning with large-scale data-distributed systems [27], distributed compresive sensing [28], [29], and massive MIMO wireless communication systems [30], [31]. Moreover, since it solves optimization problems via augmented Lagragian function, which penalizes the contraint equality with a quadratic term, it is advantageous in numerical stabability, and hence it has attracted a considerable amount of attention in both practical and theoretical aspects over decades [32]–[36].

Specifically, [32] has provided an extensive review of ADMM framework. The convergence analysis for a class of strongly convex problems with a linear equality constraint has been studied. The comprehensive work [36] has further studied the convergence rate of ADMM method with multiple separable blocks of variables in strongly convex optimization problems. It has established the global R-linear convergence rate with the sufficient small dual stepsize and the assumption that certain error bound condition holds true. In [33], a decentralized ADMM (DADMM) method is proposed for further computational efficiency. The main ingredient is to take the network topology of agents into full consideration and then each agent can individually update its variable such that it communicates with its neighbors only. [35] has further proposed a linearized version of DADMM (DLM). It combines the computational efficiency of distributed gradient method and the rapid convergence rate of ADMM. With the same assumption of strong convexity in the objective function, the linear convergence rate of both DADMM and DLM is further demonstrated. In [34], ADMM method has been investigated for solving a class of nonconvex problems with multiple separable blocks of variables, where the penalty parameter is assumed to be sufficiently large such that the subproblem is strongly convex and can be conveniently solved.

The above ADMM methods have been studied under the deterministic regime, and have assumed full accessibility to the whole distributed dataset. However, in realistic applications, the computational workload may still be heavy when the distributed dataset is large. A more natural way of handling such problem is to utilize the sampled mini-batch of dataset instead of the full distributed dataset for unbiased estimation of gradient, which leads to the incorporation of stochastic settings into ADMM. In [37], a stochastic ADMM method has been proposed for solving a class of nonsmooth convex problems. Moreover, the convergence rates for both convex and strongly convex functions has been studied under specific assumptions. [38] have investigated the well-known SVRG strategy applied in stochastic ADMM. [39] has extensively studied stochastic ADMM combined with various variance reduction strategies for nonconvex problems. Specifically, SVRG, stochastic average gradient (SAG) and the extension of SAG (SAGA) have been incorporated into ADMM method, which has resulted in SVRG-ADMM, SAG-ADMM and SAGA-ADMM respectively. All the resultant ADMM methods have been demonstrated to converge to a ϵ-stationary point in expectation. In [38], SVRG-ADMM has been further investigated for solving convex optimization problems with composite objective functions, which has achieved the convergence rates of $O(\log S/S)$ for strongly convex and Lipschitz smooth objective functions, and $O(1/\sqrt{S})$ for convex objective functions without Lipschitz smoothness. In [40], a novel stochastic ADMM has been proposed, its main ingredient is to combine classical stochastic ADMM with gradient free and variance reduction strategy.

However, all the above methods cannot be directly applied in a system with multiple agents that only a part of agents are available for updating variables while other agents remain silent at each iteration, e.g., federated learning applications. Although [36] has improved the flexibility of ADMM, the subproblem may exists large variance when the stochastic gradient is applied. Therefore, we propose a novel ADMM method, which aims to improve the flexibility of the classical stochastic ADMM method. Precisely, the novel method only requires each agent update its variables at least once periodically. While the flexibility is improved, we incorporate the variance reduction technique. Thus, our proposed method divides the procedure of classical stochastic ADMM method into two stages, where the variance reduced stochastic gradient is computed at the first stage, and then it is applied in iterations of second stage. It should be noted that when the periodicity is set to be one, i.e., all agents update their corresponding variables at each iteration, our proposed method

becomes SVRG-ADMM. Hence, our proposed method generalizes the current SVRG-ADMM. To summarize, our contributions are as follows:

- A novel stochastic ADMM method is proposed. As we have not assumed the convexity of the objective function f, our proposed method can be potentially applied to solve a specific class of nonconvex problems. Moreover, as the subproblem may be nonconvex, we incorporate Sd-REG-LBFGS to ensure the positive definiteness of the Hessian approximation. Furthermore, each agent is not required to update its variable at each iteration. Instead, we require that each agent updates its variable at least once periodically.
- We incorporate SVRG strategy into our proposed stochastic ADMM method. Thus, our proposed ADMM method is also divided into two stages as SVRG method. Although SVRG has been widely used, its application to the stochastic ADMM framework above is not straightforward since the direct combination may not ensure convergence. Considering that each agent updates its variable at least once periodically, we propose the periodicity to be the iteration number of the second stage of SVRG method.
- We have shown the convergence result of the novel method. We construct a real valued sequence, which is closely related to the augmented Larangian function (it may be viewed as a further regularized version of augmented Largrangian function). Instead of showing that the augmented Lagrangian function is decreasing at each iteration, we have shown that the constructed sequence is convergent. It lies threefold: (1). We have shown the sequence is monotonically decreasing at each iteration within the second stage; (2). We will show that the sequence is decreased at the start of the next second stage; (3). We have shown the sequence is bounded below. Thus the sequence is convergent. With the help of the convergence result, we further show that the proposed method is convergent to its KKT stationary point.

II. THE PROPOSED ALGORITHM

In this section, we start with the review of basic theory in ADMM. This part is based on [32]. Then, we will introduce the application of SRL in the subproblem of stochastic ADMM, which is quite straightforward. Specifically, suppose that there are K agents working in parallel (Here, each agent corresponds to a computation node [42]), and the large dataset $\mathcal{D}$ is divided into K subsets $\mathcal{D}_k$. Moreover, $\mathcal{D}_k$ is allocated to the kth agent. Consider the following consensus optimization problem with equality constraint:

$$\begin{aligned} \min_{x_k, k=0,\ldots,K;} \quad & \sum_{k=1}^{K} f_k\left(x_k; \mathcal{D}_k\right) + g\left(x_0\right), \\ \text{s.t.} \quad & x_k - x_0 = 0, k = 1, \ldots, K \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^d$, $f_k : \mathbb{R}^d \to \mathbb{R}$ is possibly a nonconvex function with respect to the kth agent, and $g : \mathbb{R}^d \to \mathbb{R}$ is assumed to be a convex function with respect to x_0. Moreover, the single function with respect to the data point $d_i \in \mathcal{D}_k$ is assumed to be $f_{k,i}(x_k; d_i)$. With these settings, we have $f_k(x_k) = \sum_{d_i \in \mathcal{D}_k} f_{k,i}(x_k; d_i)$. Here, we use the notation $f_k(x_k)$ for $f_k(x_k; \mathcal{D}_k)$ for simplicity, i.e., $f_k(x_k) := f_k(x_k; \mathcal{D}_k)$. The problem (1) can be solved through augmented Larangian function as follows:

$$\begin{aligned} L(\{x_k\}, \{\lambda_k\}, x_0) = & \sum_{k=1}^{K} f_k(x_k; \mathcal{D}_k) + g(x_0) \\ & + \sum_{k=1}^{K} \langle \lambda_k, x_k - x_0 \rangle \\ & + \sum_{k=1}^{K} \frac{\rho_k}{2} \|x_k - x_0\|^2 . \end{aligned} \tag{2}$$

In this section, we consider the large dataset cases and the subset $\mathcal{D}_k$ is also large. Hence, the traditional stochastic ADMM method exists the following three problems:

(1). In linearized ADMM, the subproblem $x_k^{t+1} = \operatorname{argmin}_{x_k} f_k(x_k^t) + \langle \nabla f_k(x_k^t), x_k - x_k^t \rangle + \frac{\beta}{2} \left\| x_k - x_k^t \right\|^2 + \langle \lambda_k, x_k - x_0 \rangle + \frac{\rho_k}{2} \|x_k - x_0\|^2$ may be solved more accurately by incorporating Hessian approximation B_k, where β is the approximation parameter, t is the iteration number and ρ_k is the ADMM regularization parameter. However, if $f_k(\cdot)$ is nonconvex, the subproblem can be difficult to solve since the reduction in the objective function of the subproblem may not be ensured at each iteration.

(2). Moreover, directly sampling a small subset of $\mathcal{D}_k$ to form the stochastic gradient for optimization may lead to large variance, this may further slowdown the convergence speed.

(3). All the K agents are available to implement the update of each $x_k, k = 1, \ldots, K$ in parallel, traditional stochastic ADMM method can be employed. However, a more general case is when there are less than K agents available for the implementation of the update at each iteration and thus traditional stochastic ADMM cannot be directly applied.

A. SVRG STRATEGY

The reduction of the variance of stochastic gradient can be realized by two stage optimization. In the first stage, the full gradient is evaluated at the newly updated variable. Then it is stored and used for the calculation of stochastic gradient in next stage. Specifically, denote the iteration number as $(s+1, t+1)$, where $s+1$ is iteration number of the first stage, and $t+1$ is the second stage iteration number, then the stochastic gradient proposed with respect to the kth variable x_k in [14] is given as follows:

$$v_{k,t}^{s+1} = \frac{1}{b_k^t} \sum_{d_i \in \tilde{\mathcal{D}}_k^t} [\nabla f_{k,i}(x_{k,t}^s; d_i) - \nabla f_{k,i}(\tilde{x}_k^s; d_i)] + \nabla f_k(\tilde{x}_k^s). \tag{3}$$

where a subset of data points $\tilde{\mathcal{D}}_k^t$ are randomly sampled with the batch size $b_k^t \ll N_k$. It can be easily verified

that the stochastic gradient is unbiased, i.e., $\mathbb{E}[v_{k,t}^{s+1}|x_{k,t}^{s}] = \nabla f_k(x_{k,t}^{s+1})$. As the full gradient $\nabla f_k(\tilde{x}_k^s)$ is only calculated at the first stage and maintained for the second stage, SVRG strategy is computationally efficient. Moreover, it is verified in Lemma 1 that as the algorithm converges, the variance of $v_{k,t}^{s+1}$ will be progressively reduced to null. SVRG is first proposed in [14], and soon it has been widely applied [38]–[40] and shown effectiveness and fast convergence. Considering the above benefits, we incorporate SVRG method in developing our proposed stochastic ADMM method.

As not all agents update their corresponding variables at each iteration, developing an ADMM based algorithm that guarantees convergence can be difficult. However, in the convergence analysis, by requiring that each agent updates its variable at least once every specific period T, the stochastic algorithm is expected to convergence. Specifically, at each iteration $t+1$, define an index set $\mathcal{I}^{t+1} \subseteq \{0, 1, \ldots, K\}$, if an agent index $k \in \mathcal{I}^{t+1}$, then the agent is required to update the variable x_k, otherwise, the agent keeps the variable x_k using the previous iterate. For the requirement of convergence that each variable must be updated at least once for every T iterations, we have,

$$\bigcup_{t=1}^{T} \mathcal{I}_t^s = \{0, 1, \ldots, K\}. \tag{4}$$

B. HESSIAN APPROXIMATION SCHEME

For dealing the problem 1, let us consider the quadratic approximation of the objective function in ADMM subproblem, specifically, we have:

$$\begin{aligned} x_k^{t+1} = &\operatorname{argmin}_{x_k} f_k(x_k^t) + \langle \nabla f_k(x_k^t), x_k - x_k^t \rangle \\ &+ \frac{1}{2} \left\langle B_k^t \left(x_k - x_k^t\right), x_k - x_k^t \right\rangle \\ &+ \langle \lambda_k, x_k - x_0 \rangle + \frac{\rho_k}{2} \|x_k - x_0\|^2 . \end{aligned} \tag{5}$$

Popular recursive update schemes for the approximated Hessian matrix may be used, and these include symmetric rank-one (SR1) update [17]–[19] and rank-two variants such as the Davidon-Fletcher-Powell (DFP) scheme [20], [21] and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update scheme [22], [23]. In this subsection, we mainly consider BFGS method as it is one of the most popular quasi-Newton algorithms:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}, \tag{6}$$

where the correction pairs are $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ respectively. However, there exists another problem of numerical stability. Especially for non-convex objective functions, the positive definiteness of the Hessian update is difficult to maintain. Moreover, for small dataset, it may result in the ill-conditioning problem, which harms the convergence. To address these concerns, we adopt Sd-REG-LBFGS update scheme. The main ingredient is to incorporate both regularization scheme and damped parameter, where the former can ensure numerical stability and the latter can keep the positive definiteness of the Hessian approximation matrix. Specifically, Sd-REG-LBFGS updates the Hessian approximation matrix via:

$$B_{k+1} = B_k + \frac{\tilde{\hat{y}}_k \tilde{\hat{y}}_k^T}{s_k^T \tilde{\hat{y}}_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma \tag{7}$$

with the modified gradient difference:

$$\hat{y}_k = \bar{\theta}_k y_k + (1 - \bar{\theta}_k)(B_k + \delta I)s_k, \tag{8}$$

where δ is a given positive constant that satisfies specific condition (See Lemma 1), and the damped parameter is given as:

$$\bar{\theta}_k = \begin{cases} \dfrac{0.8 s_k^T (B_k + \delta I) s_k - \gamma s_k^T s_k}{s_k^T (B_k + \delta I) s_k - s_k^T y_k}, & \text{if } s_k^T y_k \le 0.2 s_k^T (B_k \\ 1, & +\delta I)s_k + \gamma s_k^T s_k, \\ & \text{otherwise.} \end{cases} \tag{9}$$

Moreover, the extensions to stochastic regime and limited memory version are straightforward.

Algorithm 1 Flex-SADMM

Input: initialize x_k for $k = 0, \ldots, K$ and memory size M. Choose the constant δ and γ satisfying $0.8\delta > \gamma$.

for $s = 0, 1, \ldots$ **do**
  Set $\tilde{x}_k \leftarrow x_k$, and calculate $\nabla f_k(x_k)$, for $k = 1, \ldots, K$.
  for $t = 0, 1, \ldots, T-1$ **do**
    if $0 \in I$ **then**
      $x_0 \leftarrow argmin_{x_0} L(\{x_k\}, \{\lambda_k\}, x_0)$
    else $x_0 \leftarrow x_0$
    end if
    if $k \in I$ **then**
      Update x_k via (5) and update λ_k via $\lambda_k \leftarrow \lambda_k + \rho_k (x_k - x_0)$
    else
      $x_k \leftarrow x_k$, $\lambda_k \leftarrow \lambda_k$
    end if
  end for
end for

C. FLEX-SADMM

According the above discussions, we propose a novel stochastic ADMM method that only requires each agent updates its corresponding variable at least once every T iterations. For the incorporation of SVRG, we divide the ADMM procedure into two stages, where the full gradient is calculated at the first stage, and the iteration number required for the second stage is exactly set to T. In this way, each agent can update its variable at least once during the second stage. According to the above discussions, we summarize our proposed stochastic ADMM as in Algorithm 1.

In Algorithm 1, since all agents shall update their corresponding variables at least once within the T iterations at the second stage, the full gradient $\nabla f_k(\tilde{x}_k^s)$ is calculated in

step 2 for all agents $k = 1, \ldots, K$. Then $v_{k,t}^{s+1}$ is further computed for $k \in \mathcal{I}$, which is used for updating the corresponding variable in step 10. It should be noted that the steps 2-14 can be carried out in parallel. Specifically, the chosen agents in the index set $\mathcal{I}$ can update the variables x_k and λ_k in a distributed manner. In step 10, the Hessian approximation by Sd-REG-LBFGS is maintained positive definite. As mentioned above, the strategy can avoid the ill-conditioning problem and thus our proposed algorithm can perform robustly under small samples. Moreover, the subproblem in step 10 is a quadratic convex problem and can be solved by directly nulling the first-order derivative.

Remark: Our proposed stochastic ADMM method can be potentially applied in a parameter server. Moreover, for a widely used distributed gradient descent method, there have been extensive research on both asychronous and synchronous algorithms. Synchronous algorithms require that the server waits for all the workers for their update, while asynchronous collects parts of the worker information. The distributed gradient descent (DGD) has relatively smaller fault tolerance while ADMM method is able to increase the fault tolerance. Thus, ADMM method can be adapt to the asynchronous algorithm. Since our proposed method only requires that each worker updates its variables at least once every T iterations, we have provided a framework of adaptation to asynchronous algorithms.

III. CONVERGENCE ANALYSIS

For notational simplicity, let us denote $x_{k,t}^{s+1}$ as x_k^t, the same strategy is used for other variables. Subsequently, the following lemma shows the variance of the stochastic gradient is progressively reduced to zero when the algorithm converges.

Lemma 1: The variance of the stochastic gradient is bounded above by the distance between current iterate x_k^t and $\tilde{x}_k$. i.e., $\mathbb{E}\left\|v_k^t - \nabla f_k(x_k^t)\right\|^2 \leq c_k^t \mathbb{E}\left\|x_k^t - \tilde{x}_k\right\|^2$.

Proof: See Appendix A.

Next, we shall focus on the change of augmented Lagrangian function for the update of each variable, then it follows the function changes of each iteration. As mentioned above, not only the change of the augmented Lagrangian function within the second stage shall be considered, the change between first stage shall also be evaluated. Note that the function $L(\{x_k\}, \{\lambda_k\}, x_0)$ is strongly convex with respect to x_0 with modulus $\gamma > 0$, i.e.,

$$\begin{aligned} L\left(\{x_k\}, \{\lambda_k\}, x_0\right) \geq\ & L\left(\{x_k\}, \{\lambda_k\}, x_0^*\right). \\ & + \left\langle \nabla_{x_0} L\left(\{x_k\}, \{\lambda_k\}, x_0^*\right), x_0 - x_0^* \right\rangle \\ & + \frac{\gamma}{2}\left\|x_0 - x_0^*\right\|^2. \end{aligned} \tag{10}$$

In particular, since x_0^* is a minimizer of $L(\{x_k\}, \{\lambda_k\}, x_0)$, then we have:

$$L\left(\{x_k\}, \{\lambda_k\}, x_0^*\right) \leq L\left(\{x_k\}, \{\lambda_k\}, x_0\right) - \frac{\gamma}{2}\left\|x_0 - x_0^*\right\|^2. \tag{11}$$

According to this, we have the following lemma which shows that each update of x_0 is expected to lead to the reduction in the augmented Largrangian function.

Lemma 2: If $0 \in \mathcal{I}_{t+1}^{s+1}$, then we have:

$$\begin{aligned} &\mathbb{E}\left[L\left(\left\{x_{k,t}^{s+1}\right\}, \left\{\lambda_{k,t}^{s+1}\right\}, x_{0,t+1}^{s+1}\right)\right. \\ &\left. -L\left(\left\{x_{k,t}^{s+1}\right\}, \left\{\lambda_{k,t}^{s+1}\right\}, x_{0,t}^{s+1}\right)\right] \leq -\frac{\gamma}{2}\mathbb{E}\left\|x_{0,t+1}^{s+1} - x_{0,t}^{s+1}\right\|. \end{aligned} \tag{12}$$

Proof: See Appendix B.

Next, for notational convenience, let us define the function $L_k(x_k, \lambda_k, x_0) := f_k(x_k) + \langle \lambda_k, x_k - z\rangle + \frac{\rho_k}{2}\|x_k - x_0\|^2$, which can be viewed intuitively as the individual augmented Lagrangian function corresponding to the kth agent and $L(\{x_k\}, \{\lambda_k\}, x_0) = \sum_{k=1}^{K} L_k(x_k, \lambda_k, x_0)$. Then, we have the following useful lemma to evaluate the change of $L_k(x_k, \lambda_k, x_0)$ for each update of λ_k at agent k.

Lemma 3: If $k \in \mathcal{I}$, then for $t = 1, \cdots, T-1$, it holds that

$$\begin{aligned} &\mathbb{E}[L_k(x_{k,t+1}^{s+1}, \lambda_{k,t+1}^{s+1}, x_{0,t+1}^{s+1}) - L_k(x_{k,t+1}^{s+1}, \lambda_{k,t}^{s+1}, x_{0,t+1}^{s+1})] \\ &\leq \frac{c_{k,t}^{s+1}}{\rho_k^2} \cdot \mathbb{E}\left\|x_{k,t}^{s+1} - \tilde{x}_k^s\right\|^2 + \frac{L_k^2 + \psi_{U_k}^2}{\rho_k^2} \cdot \mathbb{E}\left\|x_{k,t}^{s+1} - \tilde{x}_k^s\right\|^2 \\ &\quad + \frac{c_{k,t-1}^{s+1}}{\rho_k^2} \cdot \mathbb{E}\left\|x_{k,t-1}^{s+1} - \tilde{x}_k^s\right\|^2 + \frac{\psi_{U_k}^2}{\rho_k^2} \cdot \mathbb{E}\left\|x_{k,t+1}^{s+1} - x_{k,t}^{s+1}\right\|^2, \end{aligned} \tag{13}$$

for $t = 0$, the following holds:

$$\begin{aligned} &\mathbb{E}[L_k(x_{k,1}^{s+1}, \lambda_{k,1}^{s+1}, x_{0,1}^{s+1}) - L_k(x_{k,1}^{s+1}, \lambda_{k,0}^{s+1}, x_{0,1}^{s+1})] \\ &\leq \frac{L_k^2 + \psi_{U_k}^2}{\rho_k^2}\mathbb{E}\left\|x_{k,T}^s - x_{k,T-1}^s\right\|^2 + \frac{\psi_{U_k}^2}{\rho_k^2}\mathbb{E}\left\|x_{k,1}^{s+1} - \tilde{x}_k^s\right\|^2 \\ &\quad + \frac{c_{k,T-1}^s}{\rho_k^2} \cdot \mathbb{E}\left\|x_{k,T-1}^s - \tilde{x}_k^{s-1}\right\|^2. \end{aligned} \tag{14}$$

Proof: See Appendix C.

We continue to consider the update of the variable x_k at agent k,

Lemma 4: If $k \in \mathcal{I}$, it satisfies:

$$\begin{aligned} &\mathbb{E}[L_k(x_{k,t+1}^{s+1}, \lambda_{k,t}^{s+1}, x_{0,t+1}^{s+1}) - L_k(x_{k,t}^{s+1}, \lambda_{k,t}^{s+1}, x_{0,t+1}^{s+1})] \\ &\leq -\left(\frac{\rho_k - L_k}{2} + \psi_{L_k} - \frac{c_{k,t}^{s+1}}{\psi_{L_k} + \rho_k}\right)\mathbb{E}\left\|x_{k,t+1}^{s+1} - x_{k,t}^{s+1}\right\|. \end{aligned} \tag{15}$$

Proof: See Appendix D.

Moreover, with a sufficiently large parameter ρ_k, it can be seen that L_k is decreased for each update of x_k. Now, we are ready to construct the following sequence which is helpful for the establishment of the convergence result.

Lemma 5: Define the positive sequence $\mu_{k,t}^s$

$$\mu_{k,t}^s = \begin{cases} 3 + (1 + \dfrac{1}{\alpha})\dfrac{c_{k,t+1}^s}{c_{k,t}^s}\mu_{k,t+1}^s, & \text{if } 1 \leq t \leq T-1, \\ 1, & \text{if } t = T, \end{cases} \tag{16}$$

and the positive sequence $\{\mathcal{M}_{k,t}^s\}$

$$\mathcal{M}_{k,t}^s = \begin{cases} \frac{\rho_k - L_k}{2} + \psi_{L_k} - \frac{c_{k,t}^s}{\psi_{L_k}+\rho_k} \\ \quad - \frac{L_k^2 + 2\psi_{U_k}^2}{\rho_k^2} \\ -(1+\alpha)\,\mu_{t+1}^s \frac{c_{k,t+1}^s}{\rho_k^2}, & \text{if } 1 \le t \le T-1, \\ \frac{\rho_k - L_k}{2} + \psi_{L_k} - \frac{c_{k,0}^s}{\psi_{L_k}+\rho_k} \\ -\frac{c_{k,1}^s \mu_{k,1}^s + L_k^2 + 2\psi_{U_k}^2}{\rho_k^2}, & \text{if } t = T. \end{cases} \tag{17}$$

Furthermore let us define the sequence $\{\zeta_t^s\}$, where $\zeta_t^s = \sum_{k=1}^K \zeta_{k,t}^s$ and $\zeta_{k,t}^s$ is given as follows:

$$\begin{aligned} \zeta_{k,t}^s &= \mathbb{E}[L_k(x_{k,t}^s, \lambda_{k,t}^s, x_{0,t}^s) + \mu_{k,t}^s \cdot \frac{c_{k,t}^s}{\rho_k^2} \mathbb{E}\left\|x_{k,t}^s - \tilde{x}_k^s\right\|^2 \\ &\quad + \frac{c_{k,t-1}^s}{\rho_k^2} \mathbb{E}\left\|x_{k,t-1}^s - \tilde{x}_k^s\right\|^2 \\ &\quad + \frac{L_k^2 + \psi_{U_k}^2}{\rho_k^2} \mathbb{E}\left\|x_{k,t+1}^s - x_{k,t}^s\right\|^2, \end{aligned} \tag{18}$$

then the following holds:

$$\begin{aligned} &\zeta_{t+1}^{s+1} - \zeta_t^{s+1} \\ &= \begin{cases} -\sum_{k\neq 0, k\in\mathcal{I}_{t+1}^s} \mathcal{M}_{k,t}^s \mathbb{E}\left\|x_{k,t+1}^s - x_{k,t}^s\right\|^2 \\ -\frac{\gamma}{2} K \cdot \mathbb{I}(0 \in \mathcal{I}_{t+1}^s) \mathbb{E}\left\|z_{t+1}^s - z_t^s\right\|^2, \\ -\sum_{k=1}^K \frac{c_{k,t}^s}{\rho_k^2} \mathbb{E}\left\|x_{k,t}^s - \tilde{x}_k^s\right\|^2, & \text{if } 1 \le t \le T-1, \\ -\sum_{k\neq 0, k\in\mathcal{I}_1^s} \mathcal{M}_{k,0}^s \mathbb{E}\left\|x_{k,1}^s - x_{k,0}^s\right\|^2 \\ -\frac{\gamma}{2} K \cdot \mathbb{I}(0 \in \mathcal{I}_1^s) \mathbb{E}\left\|z_1^s - z_0^s\right\|^2, \\ -\sum_{k=1}^K \frac{c_{k,T}^s}{\rho_k^2} \mathbb{E}\left\|x_{k,T}^s - \tilde{x}_k^{s-1}\right\|^2, & \text{if } t = 0. \end{cases} \end{aligned} \tag{19}$$

Proof: See Appendix E.

It can be seen from (19) that the sequence $\{\zeta_t^s\}$ is monotonically decreasing. In fact, we will show the sequence is bounded below in the following theorem. Hence, the sequence is convergent. This is useful for the establishment of the convergence of our proposed algorithm to the KKT stationary point in expectation, which is described in detail in the following theorem.

Theorem 1: For each $k \in \{1, \ldots, K\}$ the sequence $\{x_{k,t}^s, \lambda_{k,t}^s, x_{0,t}^s\}$ generated by the proposed Algorithm 1 is expected to converge to a limit point $\{x_k^*, \lambda_k^*, x_0^*\}$ which satisfies:

$$\mathbb{E}\left\|\nabla f_k(x_k^*) + \lambda_k^*\right\| = 0; \tag{20}$$

$$\mathbb{E}\left\|x_k^* - x_0^*\right\| = 0; \tag{21}$$

$$\mathbb{E}[d(\textstyle\sum_{k=1}^K \lambda_k^*, \partial g(x_0^*))] = 0, \tag{22}$$

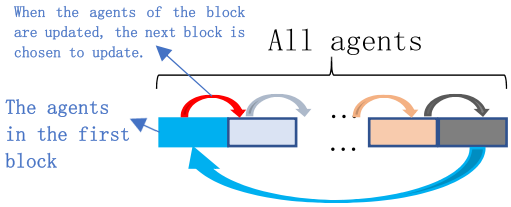

FIGURE 1. The arrangement of the available agents at each iteration in the second stage.

where the metric $d(x, \mathcal{Y})$ is the minimal distance between the vector x and the set $\mathcal{Y}$, i.e.,

$$d(x, \mathcal{Y}) = \min_{y\in\mathcal{Y}} \|x - y\|. \tag{23}$$

Moreover, given any sufficiently small positive value $\varepsilon > 0$, the iteration number S needed to achieve the following ε-stationary point

$$\mathbb{E}\left\|\nabla f_k(x_{k,t}^s) + \lambda_{k,t}^s\right\| \le \varepsilon; \tag{24}$$

$$\mathbb{E}\left\|x_{k,t}^s - x_{0,t}^s\right\| \le \varepsilon; \tag{25}$$

$$\mathbb{E}\left[d\left(\textstyle\sum_{k=1}^K \lambda_{k,t}^s, \partial g\left(x_{0,t}^s\right)\right)\right] \le \varepsilon. \tag{26}$$

satisfies $S \ge \frac{\eta(\zeta_0^2 - \zeta^*)}{\tau\varepsilon}$ for some positive constant $\eta > 0$ and $\tau > 0$.

Proof: See Appendix F.

IV. NUMERICAL EXPERIMENTS

We have studied the theoretical convergence properties of our proposed Flex-SADMM. For the convergence speed in the *Theorem 1*, we have shown that it is closely related to the first stage iteration number S. However, it is also related to the second stage iteration number T. Let us illustrate this briefly. For the convergence speed $O(1/S)$ in *Theorem 1*, it has been derived under the assumption that each agent k has updated its variables at least one time within the second stage. In fact, if we assume that each agent k is required to update its variables r times within the second stage, it can be straightforwardly derived that the convergence speed will be $O(\frac{1}{r\cdot S})$. Hence, in this section, we will study the effect of the update times r for each agent at the second stage. Here, five scenarios based on ADMM are applied for studying the effectiveness of using our proposed Hessian approximation scheme:

1). Flex-SADMM: it will be implemented according to Algorithm 1;
2). Flex-SADMM without BFGS: when the stochasticity is introduced in real applications, it will also bring stochasticity to the BFGS matrix, which will counteract or even degrade the performance that BFGS matrix has brought. Hence, it is necessary to conduct this ablation check the effect of BFGS;
3). Flex-SADMM without SVRG: to reflect the tremendous performance improvement brought by SVRG,

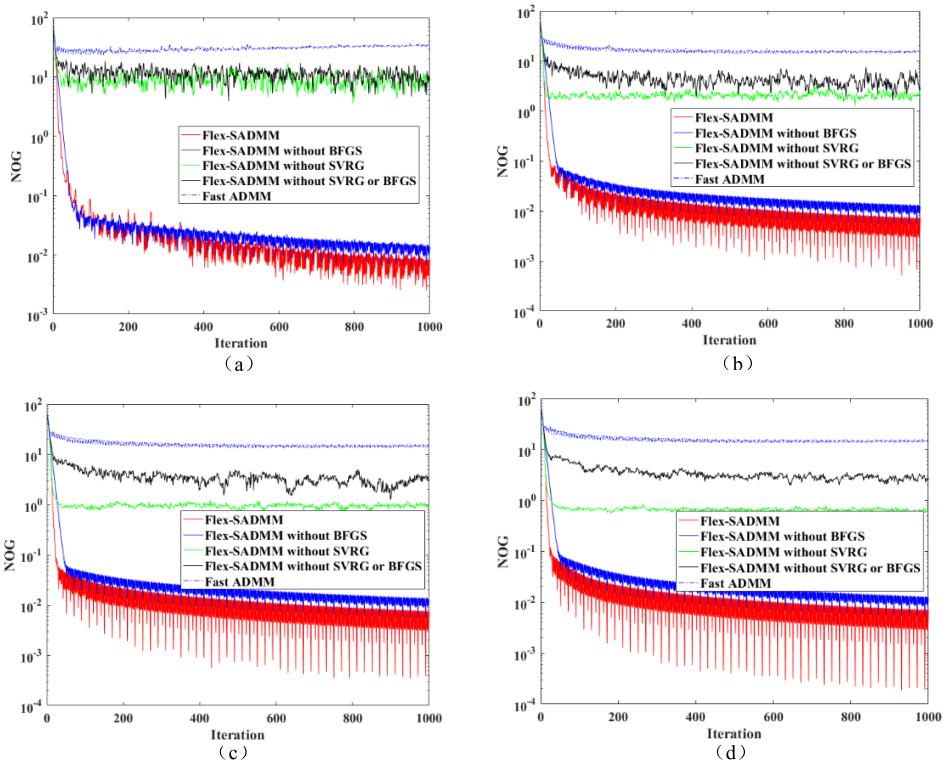

FIGURE 2. The effectiveness of different batch sizes in {1, 10, 50, 100}, to be specific, (a). $m = 1$; (b). $m = 10$; (c). $m = 50$; (d). $m = 100$.

we also implemented Flex-SADMM with no SVRG to show its importance and efficiency;

4). Flex-SADMM without SVRG or BFGS: it will be seen that SVRG has brought large performance improvement, which even overwhelms the BFGS. Hence, Flex-SADMM with no SVRG or BFGS is implemented.

5). Fast ADMM: the momentum based algorithm [43];

For the dataset, we choose *scene* dataset [41] (available at http://mulan.sourceforge.net /datasets-mlc.html), since it is simple to study the effect of the parameters. We choose this dataset since it is simple to study the effect of the parameters on our proposed algorithm. With the *Scene* data, we form a predictor of scene labels from image features. It contains 1,784 images in the training set and 446 images in the test set. There are 294 images features and up to 6 scene labels per image. We have regrouped the scene dataset to contain only two labels, with label = 1 containing 875 images for training, and the left is label = 0 group. Hence, this is a balanced binary classification problem. In relation with the problem (1), for the logistic regression problem.

Furthermore, we consider the above 5 scenarios applying to solve logistic regression for binary classification problem due to the simplicity of logistic regression. For the performance evaluation, we choose the norm of gradient (NOG) such that it can reflect how close the iteration point is to the optimal point, i.e.,

$$\text{NOG} = \left\| \frac{1}{N} \sum\nolimits_{n=1}^{N} [z_n - \sigma(\theta^T x_n)] x_n \right\| \tag{27}$$

where $\boldsymbol{\theta}$ is the optimization variable, z_n is the class label and $\boldsymbol{x_n}$ is the feature vector.

A. THE EFFECTIVENESS OF BATCH SIZE

We first show the effectiveness of the batch size in our proposed methods. Specifically, we consider the batch sizes $m \in \{1, 10, 50, 100\}$. The total iteration number is set to 1, 000 such that all algorithms is sufficient to converge. To be technically precise, the first stage iteration number is set to $S = 50$ and it follows that the second stage iteration number is $T = 20$. In particular, we arrange the available agents as one block, and each block has no common agent. Then for simplicity, one by one block is chosen at each iteration, while the unchosen agents remain silent, and the Figure 1 can illustrate this. Therefore, we assume there are $K = 10$ agents with available agents $AG = 5$ at each iteration. Subsequently, each agent will update its variables 10 times. According to *Theorem 1*, it is sufficient to ensure the convergence. Furthermore, For the Hessian approximation scheme with Sd-REG-LBFGS, the regularization parameters are set to $\gamma = 10^{-3}$ and $\delta = 1.25\gamma + 10^{-4}$ to satisfy the condition that $0.8\delta > \gamma$.

We first consider the batch size $m = 1$. It can be seen from Figure 2 that our proposed method performs nearly the same with the scenario 2). It seems that no Hessian approximation has lower computational cost while the performance is slightly better. However, without Hessian approximation in scenario 4), it performs worse than scenario 3). Hence, this performance improvement is brought by SVRG in large

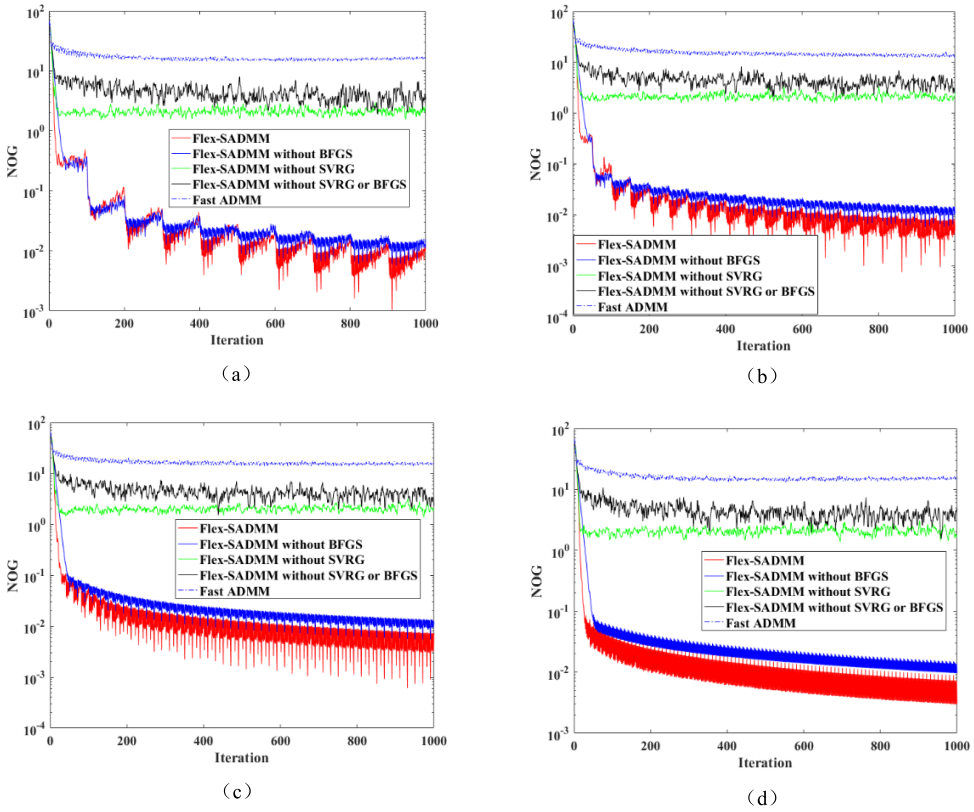

FIGURE 3. The effect of the iterations number in the first stage, to be specific, (a). $S = 10$; (b). $S = 20$; (c). $S = 50$; (d). $S = 100$.

proportion, and has overwhelmed the Hessian approximation, which can also bring significant performance improvement. But here, it should be noted that there also exists large variance in Hessian approximation matrix. This may counteract the potential performance improvement brought by the Hessian matrix to some extent. For larger batch sizes, our proposed method has shown a stable performance comparing to small batch sizes. Hence, we suggest choosing sufficient small batch size such that the computational cost is greatly reduced while the performance maintains nearly the same.

B. THE IMPACT OF FIRST STAGE

Recall that with the assumption that each agent should update its variables at least one time with the second stage, the convergence speed will become $O(1/S)$. Moreover, the number of the iteration in the first stage determines the times of calculating the full gradient. Therefore, with the above discussions, we continue to study the effect of the iteration number in first stage, namely, S. Here, the total iteration number is set to 1, 000 to ensure fair comparison. Hence, the iteration number in second stage is $1000/S$. We choose $S = [10, 20, 50, 100]$. Moreover, for simplicity, we arrange the available agents as one block, and each block has no common agent. According to this, we set the available agent number to be $AG = 5$. Moreover, the dataset is divided into $K = 10$. Subsequently, we have each agent will update its variables $r = [50, 25, 10, 5]$ times respectively.

Figure 3 shows the results of different iteration number settings in the first stage. It can be seen that convergence speed is nearly the same. This is because for the four scenarios of $r = [50, 25, 10, 5]$, since the convergence speed is $O(\frac{1}{r \cdot S})$ as aforementioned discussions, the four scenarios share nearly the same convergence speed. This matches theory well. In particular, the zigzags along the line shows the regularity, which is an interesting phenomenon. To be specific, taking Figure 3 (a) as the example, there are in general 10 zigzags and each is with smaller zigzags oscillating, we call the larger zigzag as global zigzag and the smaller as local zigzag. For the Figure 3 (a), the number of global zigzags equals to the first stage iteration number $S = 10$. This is due to the reason that the full gradient is calculated at each start of the first stage, and then it has been passed to the second stage. Hence, with the newly calculated full gradient, it will drop more obviously at the start of the second stage. For the local zigzags, it is obvious that it lies within second stage. It oscillates because only one block of agents updates its variables at each iteration while the others remain silent, and then next block is chosen for the updating. Hence, larger S will lead to dense zigzags and this matches well as the Figure 3 shows.

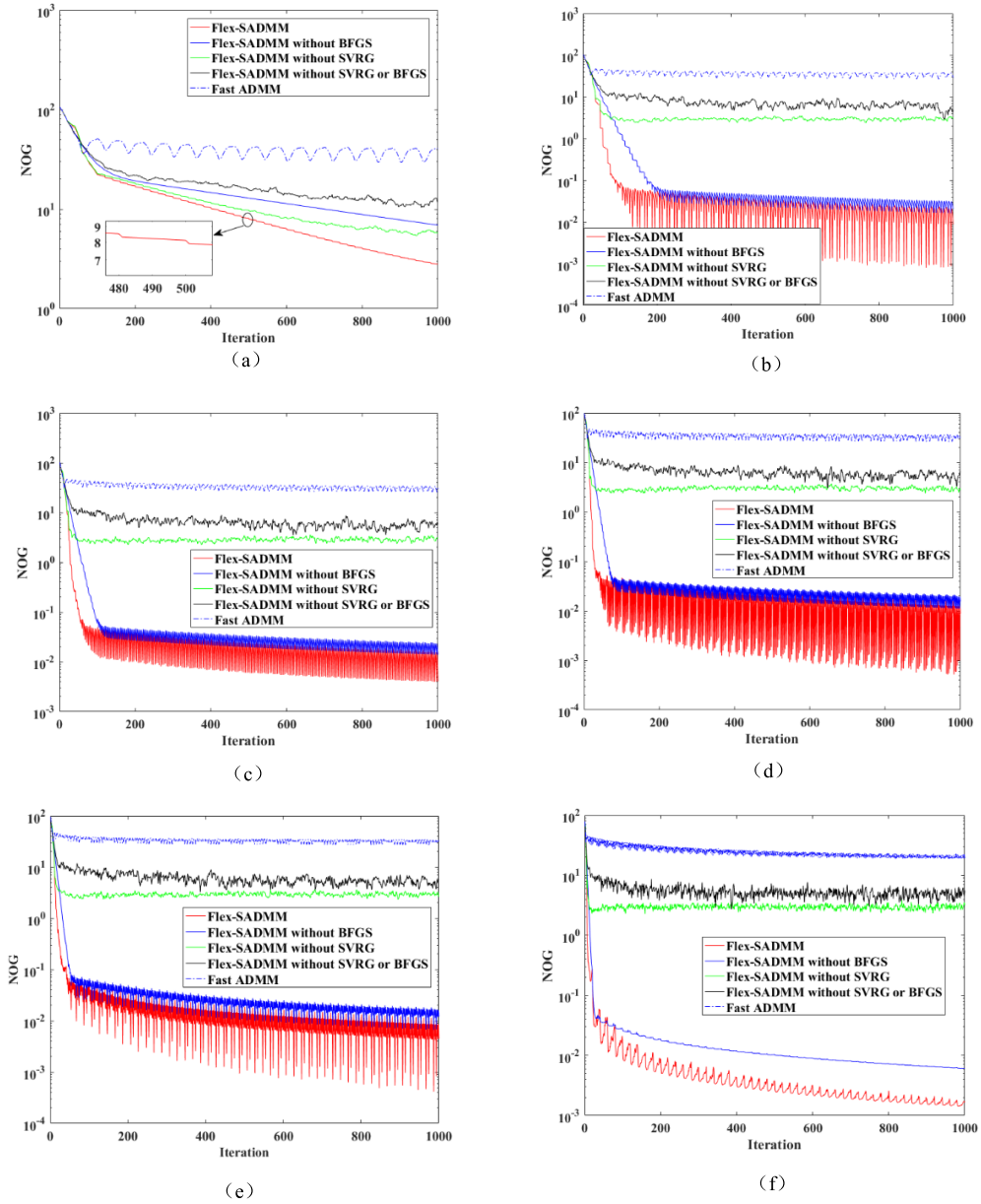

FIGURE 4. The effect of the number of available agents at each iteration, to be specific, (a). $AG = 1$; (b). $AG = 3$; (c). $AG = 5$; (d). $AG = 7$; (e). $AG = 9$; (f). $AG = 20$.

C. THE IMPACT OF AVAILABLE AGENTS

According to the above experimental results, we notice that the updating times of each agent within the second stage have a significant impact on the performance of our proposed method. Hence, we continue to study the impact of different number of the available agents at each iteration, which can determine the update times of each agent. Specifically, the total iteration number is set to 1, 000, and the first stage iteration number is set to $S = 50$, such that the update times of each agent can be varied more flexibly. For the data division, it will be set to $K = 20$. Furthermore, we set $AG = [1, 3, 5, 7, 9, 20]$. Subsequently, the corresponding update times for each agent are $[1, 3, 5, 7, 9, 20]$ respectively. It should be noted that when $AG = 20$, it means all agents update their variables at each iteration.

As Figure 4 illustrates, it can be seen that in general larger update times for each agent lead to a better performance of NOG and faster convergent speed. This matches well with the convergent theory $O(\frac{1}{r \cdot S})$. Note that the curves in Figure 4 (a) and (f) are much smoother. For Figure 4 (f), the reason is that all the agents update their variables and thus there will be no zigzags. In Figure 4 (a), it can be seen that at each start of the second stage, the curve drops more sharply than the cases in the iterations of the second stage. However, compare to other cases that multiple agents update their variables at each iteration, this decreasing is much slight.

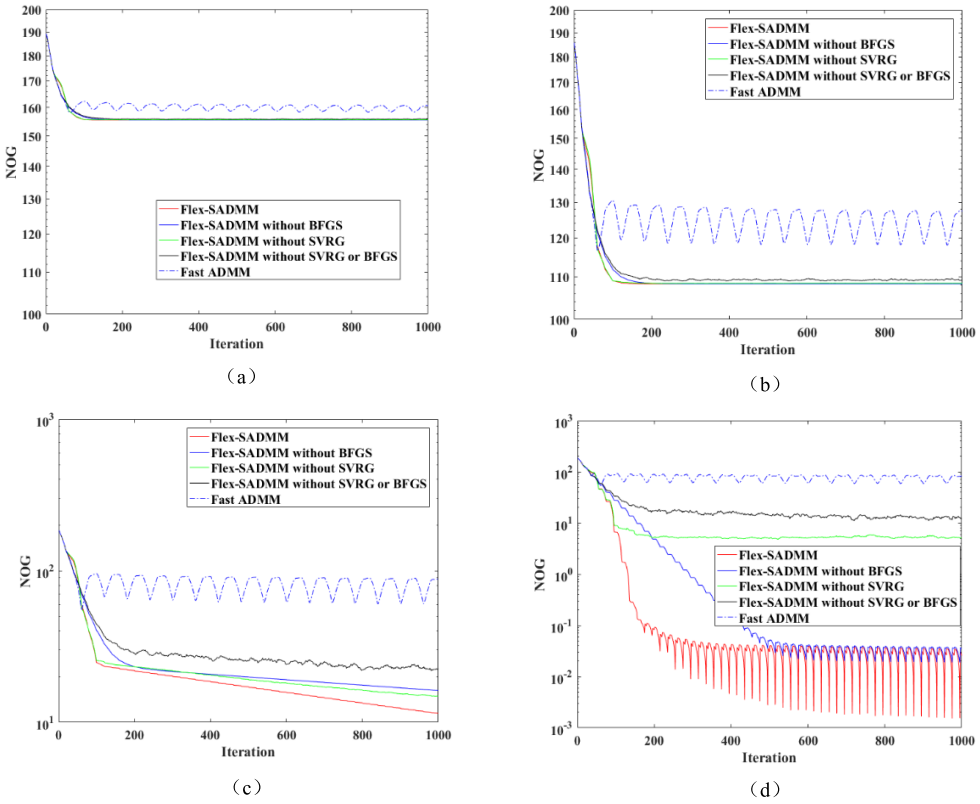

FIGURE 5. The study of the case when the condition that each agent should update its variables at least one time is not satisfied, to be specific, (a). $AG = 1$; (b). $AG = 2$; (c). $AG = 3$; (d). $AG = 4$.

This is because only one agent performs to decrease NOG. For other cases that have multiple agents available at each iteration, it can be seen that the performance improvement becomes smaller as the number of available agents increases. Hence, our proposed method has strong ability in fault tolerance, which also means that one can choose fewer available agents at each iteration for the update, because the very slight performance sacrificing is worth trading for saving hardware resources.

D. THE STUDY OF THE CONVERGENCE

Next, we continue to study when the condition that each agent updates its variables at least once is not satisfied. To achieve this scenario, the dataset is divided into $K = 60$ parts. The total iteration number is set to 1, 000 with $S = 50$. Hence, under the condition that adopts the available agent scenario according to Figure 1, there should be at least a block of 3 agents updating the variables at each iteration. To see if it is not satisfied, we choose to set to be $AG = [1, 2, 3, 4]$. Obviously, we have set three groups for studying, where the first group does not satisfy the condition, the second group just meets the condition and the third group just surpasses the condition. Specifically, $AG = [1, 2]$ does not satisfy the convergence condition. $AG = 3$ just meets the condition while $AG = 4$ just exceeds the condition.

It can be seen from Figure 5 that when the condition that each agent should update its variables at least one time (here, $AG \geq 3$) is not satisfied, all algorithms perform poorly. $AG = 3$ seems to be a water-shed, which means beyond $AG = 3$ it can reach a better and satisfactory performance, while below $AG = 3$, the performance is poor. Note that when $AG = 4$, the performances of all algorithms are substantially improved compared to $AG \leq 3$. Therefore, it matches *Theorem 1* well.

V. CONCLUSION

As there exists different issues for classical stochastic ADMM methods, we have proposed a novel stochastic ADMM method to address these concerns. Specifically, we first incorporate SVRG strategy and divide the ADMM procedure into two stages. At the second stage, the agents work for updating their corresponding variables in parallel. However, we only require each of them updates its variable at least once at the second stage. Furthermore, the convergence properties of our proposed method are comprehensively studied. Since we have not assumed the convexity of the objective function, our proposed method can be potentially applied in nonconvex problems. In the comparisons with other methods, our proposed method has shown to be advantageous especially in flexibility.

APPENDIX

A. PROOF OF LEMMA 1

Proof: Suppose at the current iteration t, we sample a subset of data points $\tilde{\mathcal{D}}_k^t$ with the batch size $b_k^t \ll N_k$ for the

stochastic gradient:

$$v_k^t = \frac{1}{b_k^t}\sum\nolimits_{d_i\in\tilde{\mathcal{D}}_k^t}[\nabla f_{k,i}(x_k^t;d_i)-\nabla f_{k,i}(\tilde{x}_k;d_i)] +\nabla f_k(\tilde{x}_k). \tag{A1}$$

Further using $\mathbb{E}\,\|\xi-\mathbb{E}\xi\|^2\le E\,\|\xi\|^2$, we have

$$\begin{aligned}
&\mathbb{E}\left\|v_k^t-\nabla f_k(x_k^t)\right\|^2\\
&=\mathbb{E}\parallel\frac{1}{b_k^t}\sum\nolimits_{d_i\in\tilde{\mathcal{D}}_k^t}[\nabla f_{k,i}(x_k^t;d_i)-\nabla f_{k,i}(\tilde{x}_k;d_i)\\
&\quad-\nabla f_k(x_k^t)+\nabla f_k(\tilde{x}_k)]|x_k^t\parallel\\
&\le\mathbb{E}\left\|\frac{1}{b_k^t}\sum\nolimits_{d_i\in\tilde{\mathcal{D}}_k^t}[\nabla f_{k,i}(x_k^t;d_i)-\nabla f_{k,i}(\tilde{x}_k;d_i)]|x_k^t\right\|^2\\
&\le\frac{1}{b_k^t}\mathbb{E}\left\|[\nabla f_{k,i}(x_k^t;d_i)-\nabla f_{k,i}(\tilde{x}_k;d_i)]|x_k^t\right\|^2\\
&\le\frac{L_k}{b_k^t}\mathbb{E}\left\|x_k^t-\tilde{x}_k\right\|^2=c_k^t\mathbb{E}\left\|x_k^t-\tilde{x}_k\right\|^2.
\end{aligned}\tag{A2}$$

Here, we have set $c_k^t=\frac{L_k}{b_k^t}$ and used the following result that:

$$\mathbb{E}\left[\frac{1}{b_k^t}\sum\nolimits_{d_i\in\tilde{\mathcal{D}}_k^t}\nabla f_{k,i}\left(x_k^t;d_i\right)-\nabla f_{k,i}\left(\tilde{x}_k;d_i\right)\right] =\nabla f_k(x_k^t)-\nabla f_k(\tilde{x}_k). \tag{A3}$$

The proof of *Lemma 1* ends.

B. PROOF OF LEMMA 2

Proof: By using the fact that $L(\{x_k\},\{\lambda_k\},x_0)$ is strongly convex with respect to x_0, we have

$$\begin{aligned}
&\mathbb{E}[L(\{x_{k,t}^{s+1}\},\{\lambda_{k,t}^{s+1}\},x_{0,t+1}^{s+1})-L(\{x_{k,t}^{s+1}\},\{\lambda_{k,t}^{s+1}\},x_{0,t}^{s+1})]\\
&\le\langle\nabla_{x_0}L(\{x_{k,t}^{s+1}\},\{\lambda_{k,t}^{s+1}\},x_{0,t}^{s+1}),x_{0,t+1}^{s+1}-x_{0,t}^{s+1}\rangle\\
&\quad+\frac{\gamma}{2}\left\|x_{0,t+1}^{s+1}-x_{0,t}^{s+1}\right\|^2.
\end{aligned}\tag{B1}$$

Note that as $x_{0,t+1}^{s+1}$ is the minimizer in the update in step 5 for $0\in\mathcal{I}_{t+1}^{s+1}$, hence, by setting $x_0^*=x_{0,t+1}^{s+1}$, we can obtain the desired result in Lemma 2, and the proof of *Lemma 2* ends.

C. PROOF OF LEMMA 3

Proof: Let us first consider the case for $t=1,\ldots,T-1$. For notational simplicity, we omit the notation s. Specifically, we have

$$\begin{aligned}
&\mathbb{E}[L_k(x_k^{t+1},\lambda_k^{t+1},x_0^{t+1})-L_k(x_k^{t+1},\lambda_k^t,x_0^{t+1})]\\
&=\frac{1}{\rho_k}\mathbb{E}\left(\left\|\lambda_k^{t+1}-\lambda_k^t\right\|\right).
\end{aligned}\tag{C1}$$

Note the subproblem in step 10 for the update of x_k, by nulling the derivative of the objective function, we have:

$$v_k^t+B_k^t(x_k^{t+1}-x_k^t)+\lambda_k^t+\rho_k(x_k^{t+1}-x_0^{t+1})=0 \tag{C2}$$

Furthermore, according to the step 13 for the update of λ_k and substituting $\lambda_k^{t+1}=\lambda_k^t+\rho_k(x_k^{t+1}-x_0^{t+1})$ into (19), we have

$$\lambda_k^{t+1}=-\left[v_k^t+B_k^t\left(x_k^{t+1}-x_k^t\right)\right]. \tag{C3}$$

Thus, substituting (C3) into (C1), it yields:

$$\begin{aligned}
&\mathbb{E}[L_k(x_k^{t+1},\lambda_k^{t+1},x_0^{t+1})-L_k(x^{t+1},\lambda_k^t,x_0^{t+1})]\\
&=\frac{1}{\rho_k^2}\mathbb{E}\parallel v_k^t+B_k^t(x_k^{t+1}-x_k^t)-v_k^{t-1}-B_k^{t-1}(x_k^t-x_k^{t-1})\parallel^2\\
&=\frac{1}{\rho_k^2}\mathbb{E}\parallel v_k^t-\nabla f_k(x_k^t)+\nabla f_k(x_k^t)-\nabla f_k(x_k^{t-1})\\
&\quad+\nabla f_k(x_k^{t-1})-v_k^{t-1}+B_k^t(x_k^{t+1}-x_k^t)\\
&\quad-B_k^{t-1}(x_k^t-x_k^{t-1})\parallel^2\\
&\le\frac{1}{\rho_k^2}\mathbb{E}\parallel v_k^t-\nabla f_k(x_k^t)\parallel^2+\frac{1}{\rho_k^2}\mathbb{E}\parallel\nabla f_k(x_k^t)-\nabla f_k(x_k^{t-1})\parallel^2\\
&\quad+\frac{1}{\rho_k^2}\mathbb{E}\parallel B_k^t(x_k^{t+1}-x_k^t)\parallel^2+\frac{1}{\rho_k^2}\mathbb{E}\parallel B_k^{t-1}(x_k^t-x_{k-1}^t)\parallel^2\\
&\le\frac{c_k^t}{\rho_k^2}\cdot\mathbb{E}\left\|x_k^t-\tilde{x}_k\right\|^2+\frac{L_k^2+\psi_{U_k}^2}{\rho_k^2}\cdot\mathbb{E}\left\|x_k^t-x_k^{t-1}\right\|^2\\
&\quad+\frac{c_k^{t-1}}{\rho_k^2}\cdot\mathbb{E}\left\|x_k^{t-1}-\tilde{x}_k\right\|^2+\frac{\psi_{U_k}^2}{\rho_k^2}\cdot\mathbb{E}\left\|x_k^{t+1}-x_k^t\right\|^2.
\end{aligned}\tag{C4}$$

Thus, we obtained the desired result. As it contains the variable $x_{k,t-1}^{s+1}$, it does not apply to the case for $t=0$. Hence, we continue to proof the result for the case with $t=0$. First, it should be noted that the last iterate $x_{k,T}^s$ within second stage is reserved as the initial value next second stage, i.e., $x_{k,0}^{s+1}=\tilde{x}_k^s=x_{k,T}^s$. Thus, we have

$$\begin{aligned}
&\mathbb{E}[L_k(x_{k,1}^{s+1},\lambda_{k,1}^{s+1},x_{0,1}^{s+1})-L_k(x_{k,1}^{s+1},\lambda_{k,0}^{s+1},x_{0,1}^{s+1})]\\
&=\frac{1}{\rho_k^2}\mathbb{E}\parallel\lambda_{k,1}^{s+1}-\lambda_{k,0}^{s+1}\parallel^2\\
&=\frac{1}{\rho_k^2}\mathbb{E}\parallel v_{k,0}^{s+1}+B_{k,0}^{s+1}(x_{k,1}^{s+1}-x_{k,0}^{s+1})-v_{k,T-1}^s\\
&\quad-B_{k,T-1}^s(x_{k,T}^s-x_{k,T-1}^s)\parallel^2\\
&=\frac{1}{\rho_k^2}\mathbb{E}\parallel v_{k,0}^{s+1}-\nabla f_k(x_{k,T}^s)+\nabla f_k(x_{k,T}^s)-\nabla f_k(x_{k,T-1}^s)\\
&\quad+B_{k,0}^{s+1}(x_{k,1}^{s+1}-x_{k,0}^{s+1})-B_{k,T-1}^s(x_{k,T}^s-x_{k,T-1}^s)\parallel^2\\
&\le\frac{1}{\rho_k^2}\mathbb{E}\parallel v_{k,0}^{s+1}-\nabla f_k(x_{k,T}^s)\parallel^2\\
&\quad+\frac{1}{\rho_k^2}\parallel\nabla f_k(x_{k,T}^s)-\nabla f_k(x_{k,T-1}^s)\parallel^2\\
&\quad+\frac{1}{\rho_k^2}\parallel B_{k,0}^{s+1}(x_{k,1}^{s+1}-x_{k,0}^{s+1})\parallel^2\\
&\quad+\frac{1}{\rho_k^2}\parallel B_{k,T-1}^s(x_{k,T}^s-x_{k,T-1}^s)\parallel^2\\
&\le\frac{L_k^2+\psi_{U_k}^2}{\rho_k^2}\mathbb{E}\left\|x_{k,T}^s-x_{k,T-1}^s\right\|^2+\frac{\psi_{U_k}^2}{\rho_k^2}\mathbb{E}\left\|x_{k,1}^{s+1}-\tilde{x}_k^s\right\|^2\\
&\quad+\frac{c_{k,T-1}^s}{\rho_k^2}\cdot\mathbb{E}\left\|x_{k,T-1}^s-\tilde{x}_k^{s-1}\right\|^2.
\end{aligned}\tag{C5}$$

Thus, we obtained the desired result in Lemma 3.

D. PROOF OF LEMMA 4

Proof: For the notational convenience, we omit the iteration number s of the first stage. For $k \in \mathcal{I}_{t+1}$, we have

$$\begin{aligned}
&\mathbb{E}[L_k(x_k^{t+1}, \lambda_k^t, x_0^{t+1}) - L_k(x_k^t, \lambda_k^t, x_0^{t+1})] \\
&= f_k(x_k^{t+1}) + \frac{\rho_k}{2} \parallel x_k^{t+1} - x_0^{t+1} \parallel^2 \\
&\quad - f_k(x_k^t) - \frac{\rho_k}{2} \parallel x_k^t - x_0^{t+1} \parallel^2 \\
&\leq \langle \nabla f_k(x_k^t), x_k^{t+1} - x_k^t \rangle + \frac{L_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\quad + \frac{\rho_k}{2} \parallel x_k^{t+1} - x_0^{t+1} \parallel^2 - \frac{\rho_k}{2} \parallel x_k^t - x_0^{t+1} \parallel^2 \\
&\quad + \langle \lambda_k^t, x_k^{t+1} - x_k^t \rangle. \qquad \text{(D1)}
\end{aligned}$$

Here, we have used the assumption that the gradient of f_k is Lipschitz continuous with respect to L_k. Recall that the update for x_k in step 10 results in (C1), this can be a guidance for further derivation by adding and subtracting terms. To start with, we first add the term $\frac{\rho_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2$ to the left hand side. Moreover, for a, b and c, the equality $\frac{1}{2} \parallel a-c \parallel^2 - \frac{1}{2} \parallel a-b \parallel^2 - \frac{1}{2} \parallel b-c \parallel^2 = \langle a-b, b-c \rangle$ holds. It subsequently yields:

$$\begin{aligned}
&\mathbb{E}[L_k(x_k^{t+1}, \lambda_k^t, x_0^{t+1}) - L_k(x_k^t, \lambda_k^t, x_0^{t+1})] \\
&\leq \langle \nabla f_k(x_k^t), x_k^{t+1} - x_k^t \rangle - \rho_k (\frac{1}{2} \parallel x_k^t - x_0^{t+1} \parallel^2 \\
&\quad - \frac{1}{2} \parallel x_k^{t+1} - x_0^{t+1} \parallel^2 - \frac{1}{2} \parallel x_k^{t+1} - x_k^t \parallel^2) \\
&\quad + \frac{L_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 - \frac{\rho_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&= \langle \nabla f_k(x_k^t), x_k^{t+1} - x_k^t \rangle - \rho_k \langle x_k^{t+1} - x_0^{t+1}, x_k^{t+1} - x_k^t \rangle \\
&\quad - \frac{\rho_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 + \frac{L_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&= \langle \nabla f_k(x_k^t) - v_k^t, x_k^{t+1} - x_k^t \rangle - \rho_k \langle x_k^{t+1} - x_0^{t+1}, x_k^{t+1} - x_k^t \rangle \\
&\quad + \frac{\rho_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 + \langle v_k^t, x_k^{t+1} - x_k^t \rangle \\
&\quad + \frac{L_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\quad + \langle B_k^t (x_k^{t+1} - x_k^t), x_k^{t+1} - x_k^t \rangle + \langle \lambda_k^t, x_k^{t+1} - x_k^t \rangle \\
&\quad - \langle B_k^t (x_k^{t+1} - x_k^t), x_k^{t+1} - x_k^t \rangle \\
&= \langle \nabla f_k(x_k^t) - v_k^t, x_k^{t+1} - x_k^t \rangle - \frac{\rho_k}{2} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\quad - \langle B_k^t (x_k^{t+1} - x_k^t), x_k^{t+1} - x_k^t \rangle \\
&= \langle \nabla f_k(x_k^t) - v_k^t, x_k^{t+1} - x_k^t \rangle \\
&\quad - \left(\frac{\rho_k}{2} + \psi_{L_k} - \frac{L_k}{2} \right) \parallel x_k^{t+1} - x_k^t \parallel^2 . \qquad \text{(D2)}
\end{aligned}$$

where we have used (C1) for the third equality. Next, we derive the term $\mathbb{E}\langle \nabla f_k(x_k^t) - v_k^t, x_k^{t+1} - x_k^t \rangle$. Specifically, since x_k^{t+1} is obtained through (C1), we further have

$$x_k^{t+1} = (B_k^t + \rho_k I)^{-1} (B_k^t x_k^t - v_k^t - \lambda_k^t + \rho_k x_0^{t+1}). \qquad \text{(D3)}$$

Hence, conditioned on x_k^t, the expectation $\mathbb{E}[(x_k^{t+1} - x_k^t) | x_k^t]$ is computed as:

$$\begin{aligned}
\mathbb{E}[(x_k^{t+1} - x_k^t) | x_k^t] &= (B_k^t + \rho_k I)^{-1} \\
&\quad \times \left[\rho_k \left(x_0^{t+1} - x_k^t \right) - v_k^t - \lambda_k^t \right]. \qquad \text{(D4)}
\end{aligned}$$

Substituting (D3), it further yields:

$$\begin{aligned}
&\mathbb{E}[L_k(x_k^{t+1}, \lambda_k^t, x_0^{t+1}) - L_k(x_k^t, \lambda_k^t, x_0^{t+1})] \\
&\leq \mathbb{E}\langle \nabla f_k(x_k^t) - v_k^t, \mathbb{E}[(x_k^{t+1} - x_k^t) | x_k^t] \rangle \\
&\quad - (\frac{\rho_k}{2} + \psi_{L_k}) \mathbb{E} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\leq \mathbb{E}\langle \nabla f_k(x_k^t) - v_k^t, (B_k^t + \rho_k I)^{-1} [\rho_k (x_0^{t+1} - x_k^t) - v_k^t - \lambda_k^t] \rangle \\
&\quad - (\frac{\rho_k}{2} + \psi_{L_k} - \frac{L_k}{2}) \mathbb{E} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\leq \frac{1}{\psi_{L_k} + \rho_k} \mathbb{E} \parallel v_k^t \parallel^2 - (\frac{\rho_k}{2} + \psi_{L_k} - \frac{L_k}{2}) \mathbb{E} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\leq -(\frac{\rho_k}{2} + \psi_{L_k} - \frac{L_k}{2} - \frac{c_k^t}{\psi_{L_k} + \rho_k}) \mathbb{E} \parallel x_k^{t+1} - x_k^t \parallel^2 . \qquad \text{(D5)}
\end{aligned}$$

Thus, we have obtained the desired result in Lemma 4.

E. PROOF OF LEMMA 5

Proof: Let us first evaluate the change of the individual augmented Lagrangian function for each update of every variable at agent k. Specifically, we have

$$\begin{aligned}
&\mathbb{E}[L_k(x_k^{t+1}, \lambda_k^{t+1}, x_0^{t+1}) - L_k(x_k^t, \lambda_k^t, x_0^t)] \\
&= \mathbb{E}[L(\{x_k^t\}, \{\lambda_k^t\}, x_0^{t+1}) - L(\{x_k^t\}, \{\lambda_k^t\}, x_0^t)] \\
&\quad + \mathbb{E}[L(\{x_k^{t+1}\}, \{\lambda_k^t\}, x_0^{t+1}) - L(\{x_k^t\}, \{\lambda_k^t\}, x_0^{t+1})] \\
&\quad + \mathbb{E}[L(\{x_k^{t+1}\}, \{\lambda_k^t\}, x_0^{t+1}) - L(\{x_k^{t+1}\}, \{\lambda_k^t\}, x_0^{t+1})] \\
&\leq -(\frac{\rho_k}{2} + \psi_{L_k} - \frac{L_k}{2} - \frac{c_k^t}{\psi_{L_k} + \rho_k} - \frac{\psi_{U_k}^2}{\rho_k^2}) \mathbb{E} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\quad + \frac{c_k^t}{\rho_k^2} \cdot \mathbb{E} \left\| x_k^t - \tilde{x}_k \right\|^2 + \frac{L_k^2 + \psi_{U_k}^2}{\rho_k^2} \cdot \mathbb{E} \left\| x_k^t - \tilde{x}_k \right\|^2 \\
&\quad + \frac{c_k^{t-1}}{\rho_k^2} \cdot \mathbb{E} \left\| x_k^{t-1} - \tilde{x}_k \right\|^2 - \frac{\gamma}{2} \mathbb{I}(k \in \mathcal{I}_{t+1}) \mathbb{E} \left\| x_0^{t+1} - x_0^t \right\|^2 . \qquad \text{(E1)}
\end{aligned}$$

Here, we have omitted the notation for $t = 1, \ldots, T-1$. According to (E1), we can add and subtract corresponding terms to construct the sequence $\zeta_{k,t}^s$, subsequently, the difference $\zeta_k^{t+1} - \zeta_k^t$ can be bounded above based on (E1):

$$\begin{aligned}
\zeta_k^{t+1} &= \mathbb{E}[L_k(x_k^{t+1}, \lambda_k^{t+1}, x_0^{t+1})] + \mu_k^{t+1} \\
&\quad \cdot \frac{c^{t+1}}{\rho_k^2} \mathbb{E} \parallel x_k^{t+1} - \tilde{x}_k \parallel^2 \\
&\quad + \frac{c^t}{\rho_k^2} \mathbb{E} \parallel x_k^t - \tilde{x}_k^s \parallel^2 + \frac{L_k^2 + \psi_{U_k}^2}{\rho_k^2} \mathbb{E} \parallel x_k^{t+1} - x_k^t \parallel^2 \\
&\leq \mathbb{E}[L_k(x_k^t, \lambda_k^t, x_0^t)] + \left[\frac{3c^t}{\rho_k^2} + \left(1 + \frac{1}{\alpha} \right) \frac{c^t}{\rho_k^2} \mu_k^{t+1} \right] \\
&\quad \cdot \mathbb{E} \parallel x_k^t - \tilde{x}_k \parallel^2 + \frac{c^{t-1}}{\rho_k^2} \mathbb{E} \parallel x_k^{t-1} - \tilde{x}_k \parallel^2 + \frac{L_k^2 + \psi_{U_k}^2}{\rho_k^2} \\
&\quad \cdot \mathbb{E} \parallel x_k^t - x_k^{t-1} \parallel^2 - \frac{\gamma}{2} \mathbb{I}(0 \in \mathcal{I}^{t+1}) \mathbb{E} \left\| x_0^{t+1} - x_0^t \right\|^2
\end{aligned}$$

$$
\begin{aligned}
&-(\frac{\rho_k}{2}+\psi_{L_k}-\frac{L_k}{2}-\frac{c_k^t}{\psi_{L_k}+\rho_k}-\frac{\psi_{U_k}^2}{\rho_k^2}-\frac{2\psi_{L_k}^2+L_k^2}{\rho_k^2}\\
&-(1+\alpha)\mu_k^{t+1}\cdot\frac{c^{t+1}}{\rho_k})\mathbb{E}\parallel x_k^{t+1}-x_k^t\parallel^2\\
&-\frac{c_k^t}{\rho_k}\mathbb{E}\parallel x_k^t-\tilde{x}_k\parallel^2\\
&=\zeta_k^t-\frac{\gamma}{2}\mathbb{I}(0\in\mathcal{I}^{t+1})\mathbb{E}\left\|x_0^{t+1}-x_0^t\right\|^2\\
&-(\frac{\rho_k}{2}+\psi_{L_k}-\frac{L_k}{2}-\frac{c_k^t}{\psi_{L_k}+\rho_k}-\frac{\psi_{U_k}^2}{\rho_k^2}-\frac{2\psi_{L_k}^2+L_k^2}{\rho_k^2}\\
&-(1+\alpha)\mu_k^{t+1}\cdot\frac{c^{t+1}}{\rho_k})\mathbb{E}\parallel x_k^{t+1}-x_k^t\parallel^2\\
&-\frac{c_k^t}{\rho_k}\mathbb{E}\parallel x_k^t-\tilde{x}_k\parallel^2. \qquad \text{(E2)}
\end{aligned}
$$

Moreover, for $t=0$, we have

$$
\begin{aligned}
&\mathbb{E}[L_k(x_{k,1}^{s+1},\lambda_{k,1}^{s+1},x_{0,1}^{s+1})-L_k(x_{k,0}^{s+1},\lambda_{k,0}^{s+1},x_{0,0}^{s+1})]\\
&=\mathbb{E}[L_k(x_{k,1}^{s+1},\lambda_{k,1}^{s+1},x_{0,1}^{s+1})-L_k(x_{k,1}^{s+1},\lambda_{k,0}^{s+1},x_{0,0}^{s+1})]\\
&\quad+\mathbb{E}[L_k(x_{k,1}^{s+1},\lambda_{k,0}^{s+1},x_{0,1}^{s+1})-L_k(x_{k,0}^{s+1},\lambda_{k,0}^{s+1},x_{0,1}^{s+1})]\\
&\quad+\mathbb{E}[L_k(x_{k,1}^{s+1},\lambda_{k,1}^{s+1},x_{0,1}^{s+1})-L_k(x_{k,1}^{s+1},\lambda_{k,0}^{s+1},x_{0,1}^{s+1})]\\
&\le-(\frac{\rho_k}{2}+\psi_{L_k}-\frac{L_k}{2}-\frac{c_k^t}{\psi_{L_k}+\rho_k}-\frac{\psi_{U_k}^2}{\rho_k^2})\\
&\quad\times\mathbb{E}\parallel x_{k,1}^{s+1}-x_{k,0}^{s+1}\parallel^2\\
&\quad+\frac{c_{k,T-1}^s}{\rho_k^2}\cdot\mathbb{E}\left\|x_{k,T-1}^s-\tilde{x}_k^{s-1}\right\|^2+\frac{L_k^2+\psi_{U_k}^2}{\rho_k^2}\\
&\quad\cdot\mathbb{E}\left\|x_{k,T}^s-x_{k,T-1}^s\right\|^2-\frac{\gamma}{2}\mathbb{I}(k\in\mathcal{I}_{t+1})\\
&\quad\times\mathbb{E}\left\|x_{0,1}^{s+1}-x_{0,0}^{s+1}\right\|^2. \qquad \text{(E3)}
\end{aligned}
$$

Based on this, we further have

$$
\begin{aligned}
&\zeta_{k,1}^{s+1}=\mathbb{E}[L_k(x_{k,1}^{s+1},\lambda_{k,1}^{s+1},x_{0,1}^{s+1})]+\mu_{k,1}^{s+1}\cdot\frac{c_k^1}{\rho_k^2}\mathbb{E}\parallel x_{k,1}^{s+1}-\tilde{x}_k\parallel^2\\
&\quad+\frac{c_k^0}{\rho_k^2}\mathbb{E}\parallel x_{k,0}^{s+1}-\tilde{x}_k^s\parallel^2+\frac{L_k^2+\psi_{U_k}^2}{\rho_k^2}\mathbb{E}\parallel x_{k,1}^{s+1}-x_{k,0}^{s+1}\parallel^2\\
&\le\mathbb{E}[L_k(x_{k,T}^s,\lambda_{k,T}^s,x_{0,T}^s)]+\frac{c_k^T}{\rho_k^2}\mathbb{E}\parallel x_{k,T}^s-\tilde{x}_k^{s-1}\parallel^2\\
&\quad+\frac{c_k^{T-1}}{\rho_k^2}\mathbb{E}\parallel x_{k,T-1}^s-\tilde{x}_k^{s-1}\parallel^2+\frac{L_k^2+\psi_{U_k}^2}{\rho_k^2}\\
&\quad\cdot\mathbb{E}\parallel x_{k,T}^s-x_{k,T-1}^s\parallel^2-\frac{\gamma}{2}\mathbb{I}(0\in\mathcal{I}_1^{s+1})\mathbb{E}\left\|x_{0,1}^{s+1}-x_{0,0}^{s+1}\right\|^2\\
&\quad-(\frac{\rho_k}{2}+\psi_{L_k}-\frac{L_k}{2}-\frac{c_k^t}{\psi_{L_k}+\rho_k}-\frac{2\psi_{U_k}^2+c_k^1\mu_{k,1}^{s+1}+L_k^2}{\rho_k^2})\\
&\quad\cdot\mathbb{E}\parallel x_{k,1}^{s+1}-x_{k,0}^{s+1}\parallel^2-\frac{c_k^t}{\rho_k}\mathbb{E}\parallel x_k^t-\tilde{x}_k\parallel^2\\
&\quad-\frac{c_k^T}{\rho_k^2}\mathbb{E}\parallel x_{k,T}^s-\tilde{x}_k^{s-1}\parallel^2-\frac{c_{k,T}^s}{\rho_k^2}\mathbb{E}\parallel x_{s,T}^s-\tilde{x}_k^{s-1}\parallel^2\\
&\quad\mathbb{E}\parallel x_{k,T}^s-x_{k,T-1}^s\parallel^2-\frac{\gamma}{2}\mathbb{I}(0\in\mathcal{I}_1^{s+1})\mathbb{E}\left\|x_{0,1}^{s+1}-x_{0,0}^{s+1}\right\|^2\\
&=\zeta_{k,0}^{s+1}-(\frac{\rho_k}{2}+\psi_{L_k}-\frac{L_k}{2}-\frac{c_k^t}{\psi_{L_k}+\rho_k}\\
&\quad-\frac{2\psi_{U_k}^2+c_k^1\mu_{k,1}^{s+1}+L_k^2}{\rho_k^2})\\
&\quad\cdot\mathbb{E}\parallel x_{k,1}^{s+1}-x_{k,0}^{s+1}\parallel^2-\frac{c_k^t}{\rho_k}\mathbb{E}\parallel x_k^t-\tilde{x}_k\parallel^2\\
&\quad-\frac{c_k^T}{\rho_k^2}\mathbb{E}\parallel x_{k,T}^s-\tilde{x}_k^{s-1}\parallel^2-\frac{c_{k,T}^s}{\rho_k^2}\mathbb{E}\parallel x_{s,T}^s-\tilde{x}_k^{s-1}\parallel^2. \qquad \text{(E4)}
\end{aligned}
$$

Subsequently, from the definition of the sequences $\{\mu_{t,k}^s\}$ and $\{\mathcal{M}_{t,k}^s\}$, combining the above we further have

$$
\begin{aligned}
&\zeta_{k,t+1}^{s+1}-\zeta_{k,t}^{s+1}\\
&=\begin{cases}
-\mathcal{M}_{k,t}^s\mathbb{E}\left\|x_{k,t+1}^s-x_{k,t}^s\right\|^2\\
-\frac{\gamma}{2}\mathbb{I}(0\in\mathcal{I}_{t+1}^s)\mathbb{E}\left\|z_{t+1}^s-z_t^s\right\|^2,\\
-\frac{c_{k,t}^s}{\rho_k^2}\mathbb{E}\left\|x_{k,t}^s-\tilde{x}_k^s\right\|^2,\ \text{if } 1\le t\le T-1,\\
-\mathcal{M}_{k,0}^s\mathbb{E}\left\|x_{k,1}^s-x_{k,0}^s\right\|^2\\
-\frac{\gamma}{2}K\cdot\mathbb{I}(0\in\mathcal{I}_1^s)\mathbb{E}\left\|z_1^s-z_0^s\right\|^2,\\
-\frac{c_{k,T}^s}{\rho_k^2}\mathbb{E}\left\|x_{k,T}^s-\tilde{x}_k^{s-1}\right\|^2,\ \text{if } t=0.
\end{cases} \qquad \text{(E5)}
\end{aligned}
$$

By taking summation, we can obtain the desired result.

F. PROOF OF THEOREM 1

Proof: we first show that the sequence $\{\zeta_t^s\}$ is bounded below. First

$$
\begin{aligned}
\zeta_{t+1}^s&\ge\sum_{k=1}^K\mathbb{E}[L_k(x_{k,t+1}^s,\lambda_{k,t+1}^s,x_{0,t+1}^s)]\\
&=\mathbb{E}[g(x_{0,t+1}^s)+\sum_{k=1}^K f_k(x_{k,t+1}^s)\\
&\quad+\langle\lambda_{k,t+1}^s,x_{k,t+1}^s-x_{0,t+1}^s\rangle+\frac{\rho_k}{2}\parallel x_{k,t+1}^s-x_{0,t+1}^s\parallel^2]\\
&=\mathbb{E}[g(x_{0,t+1}^s)+\sum_{k=1}^K f_k(x_{k,t+1}^s)]+\frac{1}{2\rho_k}\parallel\lambda_{k,t+1}^{s+1}\parallel^2\\
&\quad-\frac{1}{2\rho_k}\parallel\lambda_{k,t(k)-1}^{s+1}\parallel^2. \qquad \text{(F1)}
\end{aligned}
$$

Here, $t(k)$ denotes the last iteration that the kth agent updates its variables. Taking summation over the above equation, we have

$$
\begin{aligned}
\frac{1}{S\cdot T}\sum_{s=1}^S\sum_{t=0}^{T-1}\zeta_{t+1}^s&\le g(x_0^*)+\sum_{k=1}^K f_k(x^*)+g(x_0^*)\\
&\quad+\sum_{k=1}^K\frac{1}{2\rho_k}\parallel\lambda_{k,T}^s\parallel^2-\frac{1}{2\rho_k}\parallel\lambda_{k,0}^1\parallel^2. \qquad \text{(F2)}
\end{aligned}
$$

Thus, the sequence $\{\zeta_t^s\}$ must be bounded below. Moreover, as it is monotonically decreasing, $\{\zeta_t^s\}$ is convergent, i.e., $\lim_{s\to\infty}\zeta_t^s=\zeta^*$. Next, we will use this useful result to show

that our proposed stochastic ADMM method is convergent. Specifically, for $s = 1, \ldots, S$ and $t = 0, \ldots, T-1$, we take summation, it subsequently yields:

$$\begin{aligned}
\zeta_T^{S+1} - \zeta_0^2 &= \sum_{s=1}^{S} \sum_{t=0}^{T-1} (\zeta_{t+1}^{s+1} - \zeta_t^{s+1}) \\
&= \sum_{s=1}^{S} \sum_{t=1}^{T-1} (\zeta_{t+1}^{s+1} - \zeta_t^{s+1}) + \sum_{s=1}^{S} \zeta_1^{s+1} - \zeta_0^{s+1} \\
&\leq -\sum_{s=1}^{S} \sum_{t=0}^{T-1} \sum_{k \neq 0, k \in \mathcal{I}_{t+1}^{s+1}} \mathcal{M}_{t+1}^{s+1} \mathbb{E} \parallel x_{k,t+1}^{s+1} - x_{k,t}^{s+1} \parallel^2 \\
&\quad - \sum_{s=1}^{S} \sum_{t=0}^{T-1} \sum_{k=1}^{K} \frac{c_k^t}{\rho_k^2} \mathbb{E} \parallel x_{k,t}^{s+1} - \tilde{x}_t^s \parallel^2 \\
&\quad - \sum_{s=1}^{S} \sum_{t=0}^{T-1} \frac{\gamma K}{2} \mathbb{I}(0 \in \mathcal{I}_{t+1}^{s+1}) \mathbb{E} \parallel x_{0,t+1}^{s+1} - x_{0,t}^{s+1} \parallel^2 \\
&\quad - \sum_{s=1}^{S} \sum_{k=1}^{K} \frac{c_k^T}{\rho_k} \mathbb{E} \parallel x_{k,T}^{s} - \tilde{x}_k^{s-1} \parallel^2 .
\end{aligned} \tag{F3}$$

It should be noted that the sequences $\{\mathcal{M}_t^s\}$ and $\{\frac{c_k^t}{\rho_k}\}$ are bounded below, hence we can choose a constant $\kappa > 0$ such that

$$\kappa := \min\left(\{\mathcal{M}_t^s\}, \left\{ \frac{c_k^t}{\rho_k^2} \right\}, \frac{\gamma K}{2} \right). \tag{F4}$$

Moreover, for notational simplicity, we denote $e_{k,t}^s$ as follows:

$$\begin{aligned}
e_{k,t+1}^{s+1} = \mathbb{E} \parallel x_{k,t+1}^{s+1} - x_{k,t}^{s+1} \parallel^2 + \mathbb{E} \parallel x_{k,t}^{s+1} - \tilde{x}_t^s \parallel^2 \\
+ \mathbb{E} \parallel x_{0,t+1}^{s+1} - x_{0,t}^{s+1} \parallel^2 + \mathbb{E} \parallel x_{k,T}^{s} - \tilde{x}_k^{s-1} \parallel^2
\end{aligned} \tag{F5}$$

Therefore, by setting S to infinity and using the fact that each agent updates its variables at once every T iterations, we obtain:

$$\zeta_0^2 - \zeta^* \geq \kappa \sum_{s=1}^{S} \sum_{k=1}^{K} e_{k,t(s)+1}^{s+1}, \tag{F6}$$

subsequently, we have $\lim_{s \to \infty} e_t^s = 0$, which implies that $x_k \to x^*$ and $x_0 \to x^*$ with probability 1. Hence, for a given sufficiently small value $\varepsilon' > 0$, we can always find a S_1 such that $e_{k,t(s)}^{s'} < \varepsilon'$ for all $s' \geq S_1$. Therefore, we further obtain that:

$$\zeta_0^2 - \zeta^* \geq \kappa S \sum_{k=1}^{K} e_{k,t(s')}^{s'}. \tag{F7}$$

Here, $t(s)$ means at the s iteration of first stage, x_0 is updated at $t(s)$ iteration of the second stage. With the straightforward derivation, we can obtain

$$\mathbb{E} \left\| \nabla f_k(x_{k,t}^s) + \lambda_{k,t}^s \right\| \leq \eta_1 \sum_{k=1}^{K} e_{k,t(s')}^{s'}, \tag{F8}$$

$$\mathbb{E} \left\| x_{k,t}^s - x_{0,t}^s \right\| \leq \eta_2 \sum_{k=1}^{K} e_{k,t(s')}^{s'}, \tag{F9}$$

$$\mathbb{E} \left[d\left(\sum_{k=1}^{K} \lambda_{k,t}^s, \partial g\left(x_{0,t}^s\right) \right) \right] \leq \eta_3 \sum_{k=1}^{K} e_{k,t(s')}^{s'}. \tag{F10}$$

Hence, we can always choose a sufficiently small $\varepsilon > 0$ such that for all $s > S$ and $s' > S$, the desired results hold with $S \geq \frac{\eta(\zeta_0^2 - \zeta^*)}{\tau \varepsilon}$, where $\eta := \max(\eta_1, \eta_2, \eta_3)$. The proof of *Theorem 1* ends.

REFERENCES

[1] L. Zhang, H. C. Wu, C. H. Ho, and S.-C. Chan, "A multi-Laplacian prior and augmented Lagrangian approach to the exploratory analysis of time-varying gene and transcriptional regulatory networks for gene microarray data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 6, pp. 1816–1829, Nov./Dec. 2019.

[2] S.-C. Chan, L. Zhang, H.-C. Wu, and K.-M. Tsui, "A maximum a posteriori probability and time-varying approach for inferring gene regulatory networks from time course gene microarray data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 1, pp. 123–135, Jan./Feb. 2015.

[3] M. Neely, "Distributed stochastic optimization via correlated scheduling," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 759–772, Apr. 2016.

[4] P. Si, J. Yang, S. Chen, and H. Xi, "Smoothness constraint based stochastic optimization for wireless scalable video streaming," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 759–762, May 2015.

[5] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6369–6386, Dec. 2010.

[6] P. Hansen, B. Jaumard, and S. H. Lu, "On the use of estimates of the Lipschitz constant in global optimisation," *J. Optim. Theory Appl.*, vol. 75, pp. 195–200, Oct. 1992.

[7] P. Hansen, B. Jaumard, and S.-H. Lu, "Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison," *Math. Program.*, vol. 55, pp. 251–272, Apr. 1992.

[8] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, pp. 1303–1347, May 2013.

[9] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, May 2015.

[10] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[11] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[12] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Springer, 2010, pp. 177–186.

[13] R. M. Gower, M. Schmidt, F. Bach, and P. Richtarik, "Variance-reduced methods for machine learning," 2020, *arXiv:2010.00892*. [Online]. Available: http://arxiv.org/abs/2010.00892

[14] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 315–323.

[15] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Math. Program.*, vol. 162, nos. 1–2, pp. 83–112, 2017.

[16] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2014, pp. 1646–1654.

[17] H. Benson and Y. Shanno, "Cubic regularization in symmetric rank-1 quasi-Newton methods," *Math. Program. Comput., Math. Comput.*, vol. 10, no. 4, pp. 457–486, 2018.

[18] H. F. Khalfan, R. H. Byrd, and R. B. Schnabel, "A theoretical and experimental study of the symmetric rank-one update," *SIAM J. Optim.*, vol. 3, no. 1, pp. 1–24, Feb. 1993.

[19] A. R. Conn, N. I. M. Gould, and P. L. Toint, "Convergence of quasi-Newton matrices generated by the symmetric rank one update," *Math. Program.*, vol. 50, nos. 1–3, pp. 177–195, Mar. 1991.

[20] J. E. Dennis and J. J. Moré, "A characterization of superlinear convergence and its application to quasi-Newton methods," *Math. Comput.*, vol. 28, no. 128, pp. 549–560, Apr. 1974.

[21] W. C. Davidon, "Variable metric method for minimization," *SIAM J. Optim.*, vol. 1, no. 1, pp. 1–17, 1991.

[22] C. G. Broyden, J. E. Dennis, Jr., and J. J. Moré, "On the local and superlinear convergence of quasi-Newton methods," *IMA J. Appl. Math.*, vol. 13, no. 3, pp. 223–245, Dec. 1973.

[23] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms: 2. The new algorithm," *IMA J. Appl. Math.*, vol. 6, no. 3, pp. 222–231, 1970.

[24] N. N. Schraudolph, J. Yu, and S. Günter, "A stochastic quasi-Newton method for online convex optimization," in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, 2007, pp. 433–440.
[25] A. Mokhtari and A. Ribeiro, "Global convergence of online limited memory BFGS," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3151–3181, Jan. 2015.
[26] B. Antoine, B. Leon, and G. Patrick, "SGD-QN: Careful quasi-Newton stochastic gradient descent," *J. Mach. Learn. Res.*, vol. 10, pp. 1737–1754, Dec. 2009.
[27] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, Mar. 2020.
[28] H. Liu, B. Song, H. Qin, and Z. Qiu, "An adaptive-ADMM algorithm with support and signal value detection for compressed sensing," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 315–318, Apr. 2013.
[29] Z. Du, X. Chen, H. Zhang, and B. Yang, "Compressed-sensing-based periodic impulsive feature detection for wind turbine systems," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2933–2945, Dec. 2017.
[30] E. Vlachos, G. C. Alexandropoulos, and J. Thompson, "Massive MIMO channel estimation for millimeter wave systems via matrix completion," *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1675–1679, Nov. 2018.
[31] Y. Tsai, L. Zheng, and X. Wang, "Millimeter-wave beamformed full-dimensional MIMO channel estimation based on atomic norm minimization," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6150–6163, Dec. 2018.
[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[33] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.
[34] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.
[35] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 4051–4064, Aug. 2015.
[36] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Math. Program.*, vol. 162, pp. 165–199, Mar. 2017.
[37] H. Ouyang, N. He, L. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 80–88.
[38] Y. Yu and L. Huang, "Fast stochastic variance reduced ADMM for stochastic composition optimization," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3364–3370.
[39] F. Huang, S. Chen, and Z. Lu, "Stochastic alternating direction method of multipliers with variance reduction for nonconvex optimization," 2016, *arXiv:1610.02758*. [Online]. Available: http://arxiv.org/abs/1610.02758
[40] F. Huang, S. Gao, S. Chen, and H. Huang, "Zeroth-order stochastic alternating direction method of multipliers for nonconvex nonsmooth optimization," 2019, *arXiv:1905.12729*. [Online]. Available: http://arxiv.org/abs/1905.12729
[41] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classiffication," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.
[42] T. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.
[43] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM J. Imag. Sci.*, vol. 7, no. 3, pp. 1588–1623, 2014.

LIN WU is currently pursuing the Ph.D. degree with the School of Computer and Cyberspace Security, Communication University of China. He is also a Senior Engineer/Master's Tutor with the School of Computer and Cyberspace Security, Communication University of China. In charge of multiple projects from the Science and Technology Commission, Chaoyang, Beijing. In charge of multiple projects of university-level projects, developing multiple media application platform. He is participating and completing 863 project sub-topics of the Natural Science Foundation of China National CNGI. He have been published eight articles (including six EI articles) and eight software copyright certificates. His main research interests include theory research and development of application software including intelligent big data analysis, machine learning, and machine translation.

YONGBIN WANG is currently a Professor with the School of Computer and Cyberspace Security, Communication University of China. He is also the Director of the Technical Support Platform of the Internet Information Institute, the Deputy Director of the Key Laboratory of Convergent Media and Intelligent Technology, the Director of Academic Committee of the Key Laboratory of Acoustic Visual Technology and Intelligent Control System, Ministry of Culture and Tourism, the Director of the Academic Committee of the Beijing Key Laboratory of Modern Entertainment Technology, and the Deputy Director of the Intelligent Media Technology Center of the Beijing Collaborative Innovation Institute. His research interest includes new media technology. He is a member of the ninth and tenth council of the Chinese Computer Society and the Director of the ninth council of the Chinese Acoustics Society. He is a member of the 12th Committee of the Beijing Municipal CPPCC, a member of the NLD Central Education Committee, and the Deputy Director of the Beijing Municipal Science and Technology Commission. He received the Third Prize of the National Technology Invention Award, the Second Prize of the Beijing Science and Technology Award, the Second Prize of the Guangdong Science and Technology Progress Award, the President Award of CUC, and the Second Prize of the Beijing Education and Teaching Achievement Award. He is also the Editor-in-Chief of the *Journal of Communication University of China*.

TUO SHI was born in Beijing, China. She received the Ph.D. degree in signal and information processing from the Communication University of China, Beijing. She is currently an Associate Professor with Beijing Police College, Beijing. She is also a Postdoctoral Researcher of China CITIC Institute, Beijing. Her research interests include machine learning, data mining, deep learning, and their applications in the field of crime.