Privacy Preservation in Distributed Subgradient Optimization Algorithms

Youcheng Lou, Lean Yu, *Member, IEEE*, Shouyang Wang, and Peng Yi

***Abstract*—In this paper, some privacy-preserving features for distributed subgradient optimization algorithms are considered. Most of the existing distributed algorithms focus mainly on the algorithm design and convergence analysis, but not the protection of agents' privacy. Privacy is becoming an increasingly important issue in applications involving sensitive information. In this paper, we first show that the distributed subgradient synchronous homogeneous-stepsize algorithm is not privacy preserving in the sense that the malicious agent can asymptotically discover other agents' subgradients by transmitting untrue estimates to its neighbors. Then a distributed subgradient asynchronous heterogeneous-stepsize projection algorithm is proposed and accordingly its convergence and optimality is established. In contrast to the synchronous homogeneous-stepsize algorithm, in the new algorithm agents make their optimization updates asynchronously with heterogeneous stepsizes. The introduced two mechanisms of projection operation and asynchronous heterogeneous-stepsize optimization can guarantee that agents' privacy can be effectively protected.**

***Index Terms*—Asynchronous optimization, distributed optimization, heterogeneous-stepsize, privacy preservation.**

I. Introduction

DISTRIBUTED optimization and learning have attracted much research attention in recent years due to their wide applications in engineering, machine learning, and operations research. An efficient way for solving distributed optimization problems is to use a distributed setting instead of conventional centralized settings, in which each agent takes partial knowledge about the task and all agents exchange data with their neighbors via an underlying network communication graph.

A widely studied problem is the sum objective optimization problem $\min \sum_{i=1}^{n} f_i$, where f_i is agent i's objective function and is only known by agent i (see [1]–[3], [6]–[10], [12], [18]–[20]). Agents can solve the optimization problem in a cooperative way by their individual optimization updates and local data sharing among neighbors. Two distributed subgradient algorithms with a constant and time-varying stepsize was respectively proposed in [1] and [2] to solve the sum optimization problem and convergence analysis was provided under mild conditions. Following this, several distributed algorithms under various scenarios were successively proposed, for instance, dual averaging algorithm [7], alternating direction methods [8], [9], primal-dual and regularized primal-dual methods [25]–[27], convex intersection algorithms [11], [13]–[15], continuous-time dynamics [12], [14], [16], [17], [21], [22], nonlinear agent dynamics with external disturbances [31], unbalanced network graphs [3], [10], random network graphs [5], [19], quantization of subgradients [29], communication delays [23], [24], etc.

In distributed algorithms, in order to accomplish the optimization task agents unavoidably need to share their individual information with their neighbors. However, this direct information exchange mechanism may result in disclosure of agents' privacy. Recently, privacy is becoming an increasingly important issue in applications involving sensitive data, especially in distributed settings [32], [33]. It is clearly desirable that agents can cooperatively solve the optimization problem, and at the same time their privacy can also be effectively preserved.

While most existing work do not address the privacy preservation (see [1], [2], [4], [7]–[10], [20]–[22], [24]–[28]), some privacy-preserving algorithms have been proposed to solve distributed optimization problems recently [34]–[39]. Almost all of the existing privacy-preserving methods for distributed optimization are differential privacy-based [35]–[39]. Differential privacy-based methods typically employ a randomized perturbation technique including message perturbation [35]–[38] and objective perturbation [39] to protect agents' privacy. In differential privacy-based message perturbation methods, agents usually transit their perturbed estimates (added by a random noise) to their neighbors to guarantee a certain level of privacy preservation. One main disadvantage of this approach is that there is usually a tradeoff between the quality of the converged solution and the guaranteed level of privacy

Manuscript received February 5, 2017; revised May 20, 2017; accepted July 8, 2017. Date of publication July 31, 2017; date of current version June 14, 2018. This work was supported in part by the Key Program of National Natural Science Foundation of China under Grant 71433001 and Grant 71631005, in part by the National Natural Science Foundation of China under Grant 71401163, in part by the Hong Kong Scholars Program under Grant XJ2015049, and in part by the National Program for Support of Top-Notch Young Professionals. This paper was recommended by Associate Editor Q. Liu. *(Corresponding author: Lean Yu.)*

Y. Lou is with the Department of Systems Engineering and Engineering Management, Chinese University of Hong Kong, Hong Kong, and also with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yclou@se.cuhk.edu.hk).

L. Yu is with the School of Economics and Management, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: yulean@amss.ac.cn).

S. Wang is with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: sywang@amss.ac.cn).

P. Yi is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: peng.yi@utoronto.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2728644

preservation. In other words, this approach cannot generate an exact optimal solution under the requirement of privacy preservation. Moreover, as shown in [39], the accurate optimality still cannot be guaranteed even though there is no noise, or equivalently, without the privacy preservation requirement in the designed algorithms. Instead of message perturbation, Nozari *et al.* [39] proposed a functional perturbation method, in which before executing any distributed algorithm, agents first perturb their objective functions by employing the differential privacy method, and then solve the sum of the perturbed objective functions cooperatively. Although the objective perturbation method can guarantee the accurate optimality in the absence of noise, i.e., the optimality can be recovered when there is no privacy concern, it still suffers from a tradeoff between the accuracy of the converged solution and the ensured level of privacy preservation. Usually the level of privacy preservation should be reduced for improving the quality of the converged solution.

The suboptimality incurred by differential privacy-based approaches motivates us to rethink the privacy preservation problem of the existing distributed subgradient synchronous homogeneous-stepsize algorithm (DSSHSA) in which agents exchange estimates with their neighbors directly without employing any additional privacy-preserving technique and make their optimization updates simultaneously with the homogeneous/same stepsize. In other words, in this paper we will investigate whether the synchronous homogeneous-stepsize algorithm has intrinsic privacy-preserving properties and if not, whether we can design a new distributed algorithm that can achieve both objectives of accurate optimality of converged solutions and privacy preservation. Agents' privacy may refer to different objects in different settings, for example, convex constraint sets [36], agents' states [38], objective functions [39], or subgradients of objective functions [34]. Similar to [34], in this paper, we refer to subgradients of agents' individual objective functions as agents' privacy that needs to be protected. When we investigate the privacy-preserving properties of the DSSHSA, we assume that there is a malicious agent that does not follow the algorithm truthfully and can transmit any (untrue) data/estimates to its neighbors [30]. This malicious agent will keep a record of all data shared with its neighbors in order to discover other agents' subgradients.

The main contribution of this paper can be summarized as follows.

1) Note that most of the existing work on distributed optimization algorithms focus mainly on algorithm design and convergence analysis, but not the protection of agents' private information. In this paper, we first show that the existing DSSHSA, in which all agents optimize their objectives simultaneously with the homogeneous (same) stepsize, is not privacy preserving for almost all adjacency matrices in the sense that the malicious agent can asymptotically discover other agents' subgradients by transmitting untrue estimates to its neighbors.
2) We propose a new distributed subgradient *projection asynchronous heterogeneous-stepsize* algorithm, in which agents make their optimization updates asynchronously and the stepsizes are heterogeneous (different) among the agents. It shows that the introduced two mechanisms of projection operation and asynchronous heterogeneous-stepsize optimization can effectively protect agents' privacy. Moreover, we also establish the convergence and optimality of the newly proposed algorithm with an appropriately selected heterogeneous stepsize.
3) Compared with differential privacy-based approaches, our newly proposed algorithm has the following two advantages: a) our algorithm allows agents to exchange their estimates directly with their neighbors without requiring agents to disguise their estimates or perturb their objective functions. That is, our algorithm is easily executable and b) our algorithm can (asymptotically) achieve the accurate optimality.

This paper is closely related to the recent work [34], in which Yan *et al.* considered the privacy preservation problem of their proposed distributed subgradient online learning synchronous optimization algorithm and showed that their algorithm has intrinsic privacy-preserving properties. The authors also presented necessary and sufficient conditions to ensure the privacy-preserving properties. Different from this paper, we consider the static distributed optimization instead of dynamical (online learning) optimization in order to highlight the main motivation. In fact, the current results can be generalized to the dynamical case. In this paper, we relax the assumption that the malicious agent knows the adjacency matrix of the network graph used in [34] considering that in practice, agents are usually hard to obtain this adjacency matrix, especially in large-scale networks and distributed settings.

The rest of this paper is organized as follows. In Section II, we present some preliminaries on the DSSHSA and the interested privacy preservation problem. In Section III, we investigate the nonprivacy preserving property of the DSSHSA. In Section IV, we first present our distributed subgradient asynchronous heterogeneous-stepsize projection algorithm. Then we discuss its privacy-preserving properties, and establish its convergence and optimality. Finally, some concluding remarks are given in Section V.

Notations: $|\cdot|$ denotes the Euclidean norm of a vector, z' denotes the transpose of vector z. I_ℓ denotes the identity matrix in $\mathbb{R}^{\ell\times\ell}$, $\mathbf{1} = (1,\ldots,1)'$ is the vector of all ones. $\text{Span}\{p_1,\ldots,p_\ell\}$ and $\text{rank}\{p_1,\ldots,p_\ell\}$ denotes the subspace generated by vectors $p_1,\ldots,p_\ell$, and the rank of vectors $p_1,\ldots,p_\ell$, respectively. For a closed convex set $K\subseteq\mathbb{R}^\ell$, P_K denotes the projection operator onto K, i.e., for any $z\in\mathbb{R}^\ell$, $P_K(z)$ is the unique element that belongs to K and satisfies $|z-P_K(z)| = \inf_{y\in K}|z-y|$.

II. Preliminaries and Problem Formulation

In this section, we first introduce the DSSHSA and then state the interested privacy preservation problem of this algorithm.

A. Distributed Subgradient Synchronous Homogeneous-Stepsize Algorithm

Consider a network consisting of n agents with node set $\mathcal{V} = \{1, \ldots, n\}$. The communication among agents is described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where arc $(j, i) \in \mathcal{E}$ means that agent i can receive the estimate sent by agent j. Node j is said to be node i's neighbor if $(j, i) \in \mathcal{E}$. It is assumed that $(i, i) \in \mathcal{E}$ for all i. Let $\mathcal{N}_i = \{j | (j, i) \in \mathcal{E}\}$ denote the set of node i's neighbors. Associated with graph $\mathcal{G}$, there is usually a nonnegative adjacency matrix $\bar{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ to characterize the weights among agents, where the entries a_{ij} are nonnegative and a_{ij} is positive if and only if $(j, i) \in \mathcal{E}$. Graph $\mathcal{G}$ is said to be strongly connected if there exists a path from i to j for each pair of nodes $i, j \in \mathcal{V}$. The objective of this network is to cooperatively solve the following sum objective optimization problem:

$$\min_{x\in\mathbb{R}^m} \sum_{i=1}^{n} f_i(x) \tag{1}$$

where $f_i : \mathbb{R}^m \to \mathbb{R}$ is the convex objective function of agent i to be minimized. In a distributed setting, each agent only knows its own objective function.

An algorithm for solving (1) is the following DSSHSA proposed in [1]:

$$\begin{aligned} x_i(k+1) &= \sum_{j\in\mathcal{N}_i} a_{ij}x_j(k) - \alpha_k d_i(k), k \geq 0 \\ d_i(k) &\in \partial f_i(x_i(k)), \quad i = 1, \ldots, n \end{aligned} \tag{2}$$

where $x_i(k)$ is agent i's estimate for the optimal solution of (1) at time k; $0 < \alpha_k \leq \alpha^*$ is the stepsize, $\alpha^* > 0$; $\partial f_i(x_i(k))$ is the subdifferential that contains all subgradients of f_i at $x_i(k)$.[1] In algorithm (2), before agents generate their estimates at the next step, they first take a weighted average of the estimates received from their neighbors, and then make an optimization update following a negative gradient direction. Here the phrase "synchronous and homogeneous-stepsize" in algorithm (2) means that all agents make their optimization updates simultaneously with the homogeneous or the same stepsize $\{\alpha_k\}_{k\geq 0}$.

Remark 1: Nedić and Ozdaglar [1] proposed algorithm (2) with a constant stepsize $\alpha_k \equiv \alpha$ (and a generalized time-varying network graph) to solve optimization problem (1), where the convergence error between agents' estimates and the optimal function value is presented in terms of the constant stepsize and some other algorithm parameters. Nedić *et al.* [2] further considered a more general constrained optimization problem $\min_{x\in K} \sum_{i=1}^{n} f_i(x)$ and proposed a distributed subgradient projection algorithm with a time-varying stepsize $\{\alpha_k\}_{k\geq 0}$.

We next introduce three basic assumptions on connectivity, adjacency matrix of the network graph, and the boundedness of subgradients [1], [2], [6], [10], [34].

Assumption 1: The graph $\mathcal{G}$ is strongly connected.

Assumption 2: The adjacency matrix $\bar{A}$ is doubly stochastic, i.e., $\sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n} a_{ji} = 1$ for all i.

[1]For a convex function $g : \mathbb{R}^m \to \mathbb{R}$, $v(y)$ is said to be a subgradient of g at point $y \in \mathbb{R}^m$ if $g(z) \geq g(y) + (z - y)'v(y)$, $\forall z \in \mathbb{R}^m$.

Assumption 3: The subgradients of f_i are bounded, i.e., there is $L > 0$ such that $\sup_{q\in\bigcup_i \partial f_i(x)} |q| \leq L$, $\forall x \in \mathbb{R}^m$.

Although each agent only utilizes its own objective function, this simple weighted average information exchange mechanism can ensure that the network achieves an optimal consensus when all agents follow the algorithm truthfully, as indicated in the following theorem. This optimal consensus result can be found in [2, Proposition 2].

Theorem 1: Consider DSSHSA (2) with Assumptions 1–3, $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$. Then the network achieves an optimal consensus, i.e., there exists $\hat{x} \in \arg\min \sum_{i=1}^{n} f_i$ such that $\lim_{k\to\infty} x_i(k) = \hat{x}$, $i = 1, \ldots, n$.

B. Problem Formulation

In DSSHSA (2), agents need to share their estimates for the optimal solution with their neighbors. However, the direct information exchange mechanism may result in privacy leakage. It is desirable that agents can cooperatively accomplish the optimization task, while at the same time, agents' private information can be effectively protected. However, most of the existing distributed optimization algorithms including DSSHSA (2) focus mainly on the algorithm design and convergence analysis, not the privacy preservation (referring to algorithms in [1]–[3], [6], [7], [21], [22], and [24]–[28]) except the differentially private-based methods. Differentially private-based methods typically employ a random perturbation technique to prevent privacy disclosure [35]–[39]. A main disadvantage of differentially private-based methods is that there is a tradeoff between the optimality of the converged solution and the desired level of privacy preservation, especially that the message perturbation method still cannot guarantee the accurate optimality even in the absence of noise, or equivalently, no privacy concern [37].

The disadvantages of differential privacy-based approaches motive us to rethink the privacy-preserving properties of DSSHSA (2) in which agents exchange estimates with their neighbors directly and make their optimization updates simultaneously with homogeneous stepsizes. In this paper, we define agents' subgradients as their privacy that needs to be protected, similar to the setting in [34]. When we investigate the privacy-preserving properties of DSSHSA (2), we assume that there is a malicious agent that does not follow the algorithm correctly and can transmit any data to its neighbors. We call those agents that follow the algorithm correctly as *regular* agents. The malicious agent will keep a record of all the exchanged data with its neighbors trying to discover its neighbors' subgradients.

In this paper, we are interested in the following two privacy preservation problems.

1) Is DSSHSA (2) privacy preserving in the sense that the malicious agent can discover other agents' subgradients based on the received estimates from its neighbors and the "untrue" estimates transmitted by this malicious agent to other agents.
2) If DSSHSA (2) is not privacy preserving, can we design a privacy-preserving distributed subgradient algorithm in which agents can exchange estimates with their neighbors directly without employing any additional

privacy-preserving technique (for instance and differentially private-based method).

We will address the first problem in Section III and the second one in Section IV.

III. Nonprivacy Preserving Property of Synchronous Homogeneous-Stepsize Algorithm

In this section, we will investigate the privacy preserving properties of DSSHSA (2) in which all agents make their optimization updates simultaneously with the homogeneous/same stepsize.

Clearly, if the malicious agent can obtain the adjacency matrix of the network graph and observe all other regular agents' estimates, this malicious agent can discover other agents' subgradients by simple subtraction calculations noting that the stepsizes of all agents are the same. So it is important to consider the adjacency matrix discovery problem of DSSHSA (2). Specifically, we will first consider a special case of DSSHSA (2) with constant objective functions, i.e., the distributed consensus algorithms, and then DSSHSA (2).

In the work by Yan *et al*. [34], it is assumed that the malicious agent knows the adjacency matrix. Different from this, in this paper, we do not impose this assumption because it is generally hard to obtain this adjacency matrix in practice, especially in large-scale directed networks, taking the following two reasons into account: first, the adjacency matrix captures the global network information and then generally cannot be easily obtained by agents in a local setting and second, agents are not willing to leak the weights assigned to their neighbors to other agents from the point of view of privacy preservation.

In this section, we assume without loss of generality that agent n is the malicious agent, agents $1, 2, \ldots, n-1$ (regular agents) are this malicious agent's neighbors and the induced subgraph generated by all regular agents is strongly connected. We also assume $m = 1$ for notational simplicity in this section.

A. Adjacency Matrix Discovery of Distributed Consensus Algorithms

In this subsection, we consider the adjacency matrix discovery of the distributed consensus algorithm

$$x_i(k+1) = \sum_{j\in\mathcal{N}_i} a_{ij}x_j(k), \quad i = 1, \ldots, n, \quad k \geq 0. \tag{3}$$

Specifically, we will investigate whether the malicious agent n can discover the adjacency matrix based on the exchanged estimates with other agents. Note that the malicious agent does not follow the algorithm truthfully and can transmit any data to other regular agents.

Let $\{u(k)\}_{k\geq 0}$ be a data sequence that the malicious agent n transmits to other agents [i.e., $x_n(k) = u(k)$, $k \geq 0$]. Partition adjacency matrix $\bar{A}$ into

$$\bar{A} = \begin{pmatrix} A & b \\ * & * \end{pmatrix}$$
$$A \in \mathbb{R}^{(n-1)\times(n-1)}, \quad b = \left(a_{1n}, \ldots, a_{(n-1)n}\right)' \in \mathbb{R}^{n-1}.$$

Denote $x(k) = (x_1(k), \ldots, x_{n-1}(k))'$. Then we rewrite (3) in a compact form

$$x(k+1) = Ax(k) + bu(k), k \geq 0. \tag{4}$$

In the following, we also denote:

$$b = (b_1, \ldots, b_{n-1})'$$

for notational simplicity.

Note that $b_i > 0$ for all i since we assume that all regular agents are the malicious agent's neighbors. When there is no confusion, we roughly call the weight pair (A, b) describing the weights within regular agents and that between regular agents and the malicious agent as the adjacency matrix of (4). We now formally introduce the definition of adjacency matrix discovery. A vector is called a stochastic vector if it is nonnegative and the sum of its components is one, and a matrix is called a stochastic matrix if all its rows are stochastic vectors.

Definition 1: We say that the adjacency matrix (A, b) of (4) can not be discovered by the malicious agent if there exists another stochastic matrix $(A^*, b^*) \neq (A, b)$ such that

1) each component of b^* is positive;
2) for any sequence $\{u(k)\}_{k\geq 0}$, $x^*(k) = x(k)$ for $k \geq 0$, where $\{x^*(k)\}_{k\geq 0}$ is the estimate sequence generated by the algorithm

$$x^*(k+1) = A^*x^*(k) + b^*u(k), k \geq 0$$

with $x(0) = x^*(0)$ and can be discovered by the malicious agent otherwise.

Theorem 2: The adjacency matrix (A, b) of algorithm (4) can not be discovered by the malicious agent if and only if the following matrix equations with variable z have at least two solutions:

$$\begin{cases} (A-z)A^k b = 0, & k = 0, 1, \ldots, n-2 \\ (A-z)A^k x(0) = 0, & k = 0, 1, \ldots, n-2 \end{cases}$$

subject to $z \in \mathbb{R}^{(n-1)\times(n-1)}$, (z, b) is a stochastic matrix.

Proof (Necessity): According to the definition of adjacency matrix discovery, there exists another stochastic matrix $(A^*, b^*) \neq (A, b)$ such that for any sequence $\{u(k)\}_{k\geq 0}$, the two estimate sequences generated by algorithm (4) with respective (A, b) and (A^*, b^*) are identical. Then

$$\left(A - A^*\right)x(k) + \left(b - b^*\right)u(k) = 0, k \geq 0.$$

As a result, $b = b^*$, and consequently, $(A - A^*)x(k) = 0$ for $k \geq 0$. Therefore, $(A - A^*)x(0) = 0$. From $(A - A^*)x(1) = 0$ and $x(1) = Ax(0) + bu(0)$, we can see that $(A - A^*)b = 0$ and $(A - A^*)Ax(0) = 0$. Analogously, from $(A - A^*)x(2) = (A - A^*)(A^2x(0) + Abu(0) + bu(1))$, we can obtain that $(A - A^*)Ab = 0$, $(A - A^*)A^2x(0) = 0$. Other equations can be obtained in a similar way.

(Sufficiency): The sufficiency can be shown directly from the sufficiency hypothesis and the fact that each $A^k, k \geq n-1$ can be expressed as a linear combination of $I_{n-1}, A, \ldots, A^{n-2}$.

We complete the proof. ∎

The following two corollaries can be obtained directly from Theorem 2.

$$
\begin{aligned}
&(\mathbf{1}, x(1), \ldots, x(n), x(n+1), \ldots, x(2n-1)) \\
&\quad = (A, b)\begin{pmatrix} \mathbf{1} & x(0) & \cdots & x(n-1) & x(n) & \cdots & x(2n-2) \\ 1 & u(0) & \cdots & u(n-1) & u(n) & \cdots & u(2n-2) \end{pmatrix} \\
&\quad = (A, b) \\
&\qquad \times \begin{pmatrix} \mathbf{1} & x(0) & \cdots & A^{n-1}x(0) + \sum_{r=0}^{n-2} A^{n-2-r} b u(r) & A^n x(0) + \sum_{r=0}^{n-1} A^{n-1-r} b u(r) & \cdots & A^{2n-2}x(0) + \sum_{r=0}^{2n-3} A^{2n-3-r} b u(r) \\ 1 & u(0) & \cdots & u(n-1) & u(n) & \cdots & u(2n-2) \end{pmatrix} \qquad (6)
\end{aligned}
$$

$$
\begin{pmatrix} \mathbf{1} & x(0) & Ax(0) & \cdots & A^{n-2}x(0) & A^{n-1}x(0) & A^n x(0) + b & \cdots & A^{2n-2}x(0) + \sum_{r=n-1}^{2n-3} A^{2n-3-r} b \\ 1 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \end{pmatrix} \qquad (7)
$$

Corollary 1: If

$$
\operatorname{span}\left\{\mathbf{1}, b, Ab, \ldots, A^{n-2}b, x(0), Ax(0), \ldots, A^{n-2}x(0)\right\} = \mathbb{R}^{n-1} \tag{5}
$$

then the adjacency matrix (A, b) of algorithm (4) can be discovered by the malicious agent.

Corollary 2: If the single-input control system (4) is completely controllable [equivalently, $\operatorname{rank}(b, Ab, \ldots, A^{n-2}b) = n-1$], then the adjacency matrix (A, b) of algorithm (4) can be discovered by the malicious agent.

From Corollaries 1 and 2 we can see that for almost all adjacency matrices except a zero Lebesgue measure weight set, the adjacency matrix (A, b) of algorithm (4) can be discovered by the malicious agent. The following theorem presents a necessary and sufficient condition that the adjacency matrix can be discovered for a special class of graphs.

Theorem 3: Assume there is a node i, $i \neq n$ in graph $\mathcal{G}$ such that each node j, $j \neq i$, $j \neq n$ is a neighbor of node i. Then the adjacency matrix (A, b) of algorithm (4) can be discovered by the malicious agent if and only if (5) holds.

Proof: The sufficiency comes from Corollary 1. We now show by contradiction the necessity. We assume without loss of generality that nodes $2, \ldots, n-1$ are node 1's neighbors. As a result, all components of the first row of A, which is denoted as $\mathbf{a}$, are positive. Select a nonzero vector

$$
\mathbf{c} \in \operatorname{span}\left\{\mathbf{1}, b, Ab, \ldots, A^{n-2}b, x(0), Ax(0), \ldots, A^{n-2}x(0)\right\}^{\perp}
$$

with sufficiently small components such that all components of $\mathbf{a}-\mathbf{c}$ are positive ($^{\perp}$ denotes the orthogonal complement of a subspace). Note that $(\mathbf{a}-\mathbf{c}, b_1)$ is also a stochastic vector because $(\mathbf{a}-\mathbf{c})'\mathbf{1} = \mathbf{a}'\mathbf{1} = 1 - b_1$. Therefore, the matrix z with the first row being $\mathbf{a}-\mathbf{c}$ and all other rows are the same as that of A is also a solution of the matrix equations in Theorem 2. This contradicts Theorem 2 and consequently, the necessity follows. The proof is completed. ■

We next present a necessary and sufficient condition on adjacency matrix recovery when the network contains only three agents.

Theorem 4: Consider algorithm (4) with a completely connected graph and $n = 3$. Then the adjacency matrix (A, b) of algorithm (4) can not be discovered by the malicious agent if and only if $b_1 = b_2$, $x_1(0) = x_2(0)$ and $a_{11} + a_{12} = a_{21} + a_{22}$.

Proof: The sufficiency is straightforward. In fact, when the sufficient conditions hold, any nonnegative matrix $\begin{pmatrix} z_1 & z_2 \\ z_3 & z_4 \end{pmatrix}$ satisfying $z_1 + z_2 = z_3 + z_4 = a_{11} + a_{12}$ is a solution of the matrix equations in Theorem 2.

We now show the necessity by contradiction. Hence suppose $x_1(0) \neq x_2(0)$. Then from $(A - z)x(0) = 0$ and (z, b) is a stochastic matrix, we get that $a_{11}x_1(0) + a_{12}x_2(0) = z_1x_1(0) + z_2x_2(0)$ and $a_{11} + a_{12} = z_1 + z_2$. That is

$$
\begin{pmatrix} x_1(0) & x_2(0) \\ 1 & 1 \end{pmatrix}\begin{pmatrix} a_{11} - z_1 \\ a_{12} - z_2 \end{pmatrix} = 0.
$$

The above equation implies that $z_1 = a_{11}$, $z_2 = a_{12}$ due to $x_1(0) \neq x_2(0)$. Similarly, we can show that $z_3 = a_{21}$, $z_4 = a_{22}$. This implies that the matrix equations in Theorem 2 has a unique solution, which raises a contradiction. Therefore, $x_1(0) = x_2(0)$. Analogously, from $(A - z)Ax(0) = 0$ we can also prove that the two entries of $Ax(0)$ are the same. Then it follows that $a_{11} + a_{12} = a_{21} + a_{22}$. From the first matrix equation in Theorem 2 we can also show that $b_1 = b_2$ in a similar way. The proof is completed. ■

We now consider how the malicious agent chooses an appropriate sequence $\{u(k)\}_{k\geq 0}$ to discover the adjacency matrix (A, b) when condition (5) holds.

Theorem 5: Assume (5) holds. Then the adjacency matrix (A, b) of distributed algorithm (4) can be discovered by the malicious agent by choosing

$$
\begin{aligned}
u(0) = u(1) = \cdots = u(n-2) = 0 \\
u(n-1) = u(n) = \cdots = u(2n-2) = 1.
\end{aligned}
$$

Proof: From Corollary 1 we know that the adjacency matrix (A, b) of algorithm (4) can be discovered by the malicious agent under the condition (5). By (4) we get (6), as shown at the top of this page.

Rewrite the matrix (6) as $Z = (A, b)Y$. Note that the square matrix YY' is invertible if and only if Y is full row rank. So if matrix Y has full row rank, then (A, b) is uniquely determined by

$$
(A, b) = ZY'\left(YY'\right)^{-1}.
$$

We next show that the matrix Y has full row rank under condition (5) by choosing $u(0), \ldots, u(2n-2)$ given in this theorem. By letting $u(0) = u(1) = \cdots = u(n-2) = 0$ and $u(n-1) = \cdots = u(2n-2) = 1$ in Y yields the matrix in (7), as shown at the top of this page. Noting that any A^k, $k \geq n-1$

can be expressed as a linear combination of $I_{n-1}, A, \ldots, A^{n-2}$, we can find that the matrix in (7) is certainly full row rank. ■

B. Adjacency Matrix Discovery of DSSHSA (2)

In last subsection, we study the adjacency matrix discovery problem of distributed consensus algorithms. In this subsection, we proceed to consider this problem of DSSHSA (2). We rewrite algorithm (2) in the following compact form:

$$x(k+1) = Ax(k) + bu(k) - \varepsilon(k), k \geq 0 \tag{8}$$

where $\varepsilon(k) = (\varepsilon_1(k), \ldots, \varepsilon_{n-1}(k))'$, $\varepsilon_i(k) = \alpha_k d_i(k)$.

We first present a useful lemma for the following analysis.

Lemma 1: Assume Assumptions 1–3 hold. Then the estimates $x_i(k), i, k$ generated by DSSHSA (2) or algorithm (8) are bounded if the $u(k), k \geq 0$ transmitted by the malicious agent to other agents are bounded.

Proof: By Assumption 3, we have $|\varepsilon_i(k)| \leq \alpha_k L \leq \alpha^* L$. It is also easy to see that $||A||_\infty := \max_{1\leq i\leq n-1} \sum_{j=1}^{n-1} a_{ij} = \max_{1\leq i\leq n-1}(1-b_i) < 1$. From (8) by induction we have

$$x(k+1) = A^{k+1}x(0) + \sum_{r=0}^{k} A^{k-r}(bu(r) - \varepsilon(r)), k \geq 0.$$

The proceeding three relations imply that for any k

$$\begin{aligned} ||x(k+1)||_\infty &\leq ||A||_\infty^{k+1}||x(0)||_\infty + \sum_{r=0}^{k} ||A||_\infty^{k-r}\left(u^* + \alpha^* L\right) \\ &\leq ||x(0)||_\infty + \frac{u^* + \alpha^* L}{1 - ||A||_\infty} \\ &< \infty \end{aligned}$$

where $u^* := \sup_{k\geq 0}|u(k)|$ is a finite number by the hypothesis. Then the proof is completed. ■

It is time to present our first important result of this paper.

Theorem 6: Consider DSSHSA (2) with Assumptions 1–3, $\text{rank}(b, Ab, \ldots, A^{n-2}b) = n-1$ and $\lim_{k\to\infty}\alpha_k = 0$. Then the adjacency matrix (A, b) of DSSHSA (2) or algorithm (8) can be discovered asymptotically by the malicious agent by choosing an appropriate sequence $\{u(k)\}_{k\geq 0}$.

Proof: Denote $s_{r,k} = r(2n-1)+k, r \geq 0, k = 0, \ldots, 2n-2$ and let $u(s_{r,0}) = u(s_{r,1}) = \cdots = u(s_{r,n-2}) = 0$, $u(s_{r,n-1}) = u(s_{r,n}) = \cdots = u(s_{r,2n-2}) = 1$ for each $r \geq 0$. Similar to the analysis in the proof of Theorem 5

$$\begin{aligned} (A, b) = Z_r Y_r'\left(Y_r Y_r'\right)^{-1} + \left(0, \varepsilon\left(s_{r,1}-1\right), \ldots, \varepsilon\left(s_{r,2n-1}-1\right)\right) \\ \times\ Y_r'\left(Y_r Y_r'\right)^{-1} \end{aligned} \tag{9}$$

where $Z_r = (\mathbf{1}, x(s_{r,1}), x(s_{r,2}), \ldots, x(s_{r,2n-1}))$, Y_r is defined similarly as the matrix given in (7) by replacing $x(0)$ with $x(s_{r,0})$. Under the hypothesis of $\text{rank}(b, Ab, \ldots, A^{n-2}b) = n-1$, $Y_r Y_r'$ is full row rank and hence the inverse $(Y_r Y_r')^{-1}$ exists.

By Lemma 1, the estimates $x_i(k), i, k$ are bounded. Then $Y_r Y_r', r \geq 0$ are bounded and as a result, we can show by contradiction that $(Y_r Y_r')^{-1}, r \geq 0$ are also bounded based on the following two facts.

1) Let $\{B_r\}, \{C_r\}$ be two square matrix sequences. If $B_r = C_r^{-1}$ for any r and $\lim_{r\to\infty} B_r = B$, where the inverse B^{-1} exists, then $\lim_{r\to\infty} C_r = B^{-1}$.
2) Under the condition $\text{rank}(b, Ab, \ldots, A^{n-2}b) = n-1$, the inverse of $Y_r Y_r'$ exists for any $x(s_{r,0})$, here we regard $Y_r Y_r'$ as a matrix function with variable $x(s_{r,0})$ in a bounded closed set.

The boundedness of $(Y_r Y_r')^{-1}, r \geq 0$ combines with the hypothesis condition $\lim_{k\to\infty}\alpha_k = 0$ imply that the second term in (9) tends to zero as $r \to \infty$. Then we conclude that (A, b) can be discovered asymptotically by the malicious agent in the sense that

$$\lim_{r\to\infty} |Z_r Y_r'\left(Y_r Y_r'\right)^{-1} - (A, b)| = 0.$$

We complete the proof. ■

Remark 2: We can see that the stepsize condition $\lim_{k\to\infty}\alpha_k = 0$ given in Theorem 6 naturally holds under the condition $\sum_{k=0}^{\infty}\alpha_k^2 < \infty$ in Theorem 1.

C. Nonprivacy Preserving Property of DSSHSA (2)

The result in Theorem 6 implies that the synchronous homogeneous-stepsize algorithm (2) *is not privacy preserving* in the sense that the malicious agent can asymptotically discover the adjacency matrix and other agents' subgradients by choosing an appropriate data sequence transmitted to other regular agents. In fact, according to the proof of Theorem 6, $\lim_{r\to\infty} A_r = A$, $\lim_{r\to\infty} b_r = b$, where (A_r, b_r) is the matrix pair such that $Z_r Y_r'(Y_r Y_r')^{-1} =: (A_r, b_r)$. Then we can find that regular agents' subgradients at any time k can be obtained by

$$\frac{A_r x(k) + b_r u(k) - x(k+1)}{\alpha_k}$$

with sufficiently large r. Under the assumption that the malicious agent knows the adjacency matrix, Yan *et al.* [34] showed that the malicious agent can discover other regular agents' subgradients if and only if all other regular agents are the malicious agent's neighbors. This is consistent with our result.

IV. Distributed Subgradient Asynchronous Heterogeneous-Stepsize Projection Algorithm

In last section, we showed that when the malicious agent does not follow the algorithm correctly and can observe all other regular agents' estimates, for almost all adjacency matrices except a zero Lebesgue measure weight set, DSSHSA (2) is not privacy preserving in the sense that regular agents' subgradients can be asymptotically discovered by the malicious agent. In this section, we will propose a new privacy-preserving distributed subgradient algorithm and strictly establish its convergence of optimality.

The main design idea of the newly proposed algorithm is that agents optimize their individual objective functions asynchronously and the stepsizes are heterogeneous. Additionally, we artificially introduce a projection set in the estimate iterations.

Note that in Algorithm 1, c_i is a constant, $\kappa_i(r)$ is the time when agent i makes its rth optimization update. Here c_i and $\{\kappa_i(r)\}_{r\geq 1}$ are referred to as agent i's privacy preservation constant and optimization update time sequence, respectively, which are deterministic and only known by agent i.

Algorithm 1 Distributed Subgradient Asynchronous Heterogeneous-Stepsize Projection Algorithm

Initialization: privacy preservation constants $c_i \geq 0$, optimization update sequence $\{\kappa_i(r)\}_{r\geq 1}$, initial conditions $x_i(0) \in \mathbb{R}^m, i = 1, \ldots, n$, closed convex projection set X, adjacency matrix $\bar{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$.

Algorithm:

for $i = 1, ..., n$, take $d_i(k) \in \partial f_i(x_i(k))$ and let

$$x_i(k+1) = P_X\Big(\sum_{j\in\mathcal{N}_i} a_{ij}x_j(k) - \frac{1}{c_i + r}d_i(k)\Big)$$

if $k = \kappa_i(r)$ for some r, and

$$x_i(k+1) = P_X\Big(\sum_{j\in\mathcal{N}_i} a_{ij}x_j(k)\Big)$$

otherwise.

Output: agent i's estimate sequence $\{x_i(k)\}_{k\geq 0}$ for the optimal solution of optimization problem (1), $i = 1, \ldots, n$.

After taking a weighted average of the estimates received from its neighbors, each agent will take a subgradient optimization step and a projection onto set X to generate the estimate at the next step if the current time is this agent's optimization update time, and will just take the projection of the weighted average onto set X as the estimate at the next step otherwise.

Each agent does not know other agents' optimization update times and then implies that agents make their optimization updates *asynchronously*. When one agent makes its optimization update at some time, the stepsize at this time is taken as the inverse of the sum of some constant and the number of optimization update times up to the current time. Then the stepsizes are *heterogeneous* among the agents since agents have different optimization update time sequences and different privacy preservation constants. Here, we also artificially introduce a bounded closed convex projection set X, which is known by all agents and is assumed to contain all the optimal solutions of $\min \sum_{i=1}^n f_i$. Under this assumption, we can see that both the optimal solutions of $\min \sum_{i=1}^n f_i$ and $\min_X \sum_{i=1}^n f_i$ are identical.

Different from differential privacy-based methods [35]–[39], in our algorithm agents neither need to perturb their objective functions nor disguise their estimates. Instead, agents exchange the estimates with their neighbors directly. This makes this new algorithm easily executable. Besides this advantage, the following discussions also illustrate that the introduced projection operation and asynchronous heterogeneous-stepsize optimization mechanism can ensure that the proposed algorithm is privacy preserving.

Remark 3: In Algorithm 1, after taking a weighted average of the estimates received from their neighbors and before generating the estimates at the next step, agents make their optimization updates or not. That is, agents make their optimization updates just at some times. In fact, this intermittent optimization update mechanism has appeared in the literature, for instance, the random sleep algorithms [13], [15], and random asynchronous algorithms [18]–[20]. In [19] and [20] agents choose to make their optimization updates or not randomly, and the stepsize is random and taken as the inverse (or some power of the inverse) of the number of optimization update times up to the current time. Different from them, the stepsize in our algorithm is deterministic. In fact, these randomized optimization algorithms without constraints and stochastic error are not privacy preserving in some sense since based on the results in last section, the malicious agent can discover other agents' stepsizes and then the subgradients with a positive probability if the malicious agent can take the full knowledge of the adjacency matrix and observe all other agents' estimates.

Remark 4: The stepsize choice is extremely important to the optimality of the converged solution in distributed subgradient algorithm design. In fact, [10, Ths. 4.2 and 4.4] show that for a network graph with doubly stochastic adjacency matrix, the optimality can be guaranteed by a homogeneous stepsize and may be not if the stepsizes are different among agents. However, the results in last section show that the homogeneous-stepsize design and simultaneous optimization update mechanism make algorithm (2) not privacy preserving. Therefore, the stepsize design brings a new challenge when we take the privacy into account. In Algorithm 1, we take the stepsize as the inverse of the sum of the privacy preservation constant and the times that agents make their optimization updates up to the current time, similar to that in [18]–[20]. Our result shows that the optimality can be guaranteed provided that for each agent, the number of its optimization update times is the same over different time intervals with the same length.

Remark 5: In Algorithm 1, for the unconstrained optimization problem $\min \sum_{i=1}^n f_i$, we artificially introduce a projection set from the viewpoint of privacy preservation. We can see that Algorithm 1 also works for the constrained optimization problem $\min_K \sum_{i=1}^n f_i$. For this constrained optimization problem, X can be taken as a subset that contains all the optimal solutions of $\min_K \sum_{i=1}^n f_i$.

A. Privacy-Preserving Properties

Before establishing the convergence and optimality of Algorithm 1, in this subsection we first illustrate that Algorithm 1 is privacy preserving from the two aspects of projection operation and asynchronous heterogeneous-stepsize optimization mechanism. In this section, we denote

$$\tilde{x}_i(k) = \sum_{j\in\mathcal{N}_i} a_{ij}x_j(k)$$

for notational simplicity.

First, when the "estimate" $\tilde{x}_i(k) - (1/c_i + r)d_i(k)$ locates outside the projection set X, from the property of convex projection operator

$$P_X(z) = P_X(P_X(z) + \lambda(z - P_X(z))),^2\ \forall z \notin X,\ \lambda \geq 0$$

[2]This property of convex projection operator follows from the fact that $w = P_X(z)$ if and only if $(z - w)'(y - w) \leq 0$ for any $y \in X$. This fact can be shown directly from the definition of convex projection.

we know that the malicious agent cannot infer other agents' subgradients at time k based on its received estimates even though the malicious agent knows the adjacency matrix. Moreover, when the estimate $\tilde{x}_i(k) - 1/(c_i + r)d_i(k)$ locates inside set X, Algorithm 1 evolves in the following form:

$$x_i(k+1) = \begin{cases} \tilde{x}_i(k) - \frac{1}{c_i+r} d_i(k), & \text{if } k = \kappa_i(r) \text{ for some } r \\ \tilde{x}_i(k), & \text{otherwise.} \end{cases}$$

This reveals that this malicious agent can also not discover other agents' subgradients at time k based on the following reasons. On one hand, even if the malicious agent knows the adjacency matrix (A, b) and can observe all regular agents' estimates, but note that since the malicious agent does not know whether the regular agent i, $i \neq n$ makes its optimization update at time k, so in this asynchronous heterogeneous-stepsize algorithm, knowing $x_i(k+1) - \tilde{x}_i(k)$ can not help the malicious agent discover the subgradients; on the other hand, even if the malicious agent also knows that agent i makes its optimization update at time k, which helps the malicious agent discover $1/(c_i + r)d_i(k)$ from $x_i(k+1) - \tilde{x}_i(k)$, but this malicious agent still can not discover the subgradient since it does not know the stepsize $1/(c_i + r)$ considering that the optimization update time sequences and privacy preservation constants are different among the agents and each agent only knows its own privacy preservation constant and update time sequence.

As a sum, we conclude that when agents are far from the projection set X, both the projection operation and the asynchronous heterogeneous-stepsize optimization mechanism can effectively protect agents' privacy, while when close to the desired optimal solution $x^* \in \arg\min \sum_{i=1}^n f_i$ (the convergence and optimality will be proven in the following theorem), it is the asynchronous heterogeneous-stepsize optimization mechanism that mainly protects agents' privacy.

B. Convergence and Optimality

In this subsection, we will establish the convergence and optimality of the newly proposed asynchronous heterogeneous-stepsize algorithm. We next make an assumption on agents' optimization update time sequences $\{\kappa_i(r)\}_{r\geq 1}$, $i = 1, \ldots, n$.

Assumption 4: For each agent i, there exists an integer $T_i > 0$ such that $1 \leq t_i(s_1) = t_i(s_2) < \infty$, $\forall s_1, s_2$, where

$$t_i(s) = |\{r | sT_i \leq \kappa_i(r) < (s+1)T_i\}|$$

denotes the times of agent i's optimization updates within the interval $[sT_i, (s+1)T_i)$.

Assumption 4 requires that each agent makes its own optimization update with a constant number of times within any time interval with some fixed length. Note that the numbers of optimization updates within the time interval with this fixed length may be different among the agents. We can see that Assumption 4 holds if each agent makes its optimization update in a periodic way, no matter whether the periods for agents are the same.

We now establish the convergence and optimality of Algorithm 1.

Theorem 7: Consider distributed subgradient asynchronous heterogeneous-stepsize projection Algorithm 1 with Assumptions 1, 2, and 4. Then the network will achieve an optimal consensus, i.e., there exists $\hat{x} \in \arg\min \sum_{i=1}^n f_i$ such that $\lim_{k\to\infty} x_i(k) = \hat{x}$, $i = 1, \ldots, n$.

Proof: Without loss of generality, we assume in this proof that the privacy preservation constants $c_i = 0$, $i = 1, \ldots, n$ since we can similarly show the convergence and optimality for the general case. First, it follows from $x_j(k) \in X$, Assumption 2 and the convexity of X that $\tilde{x}_i(k) \in X$. Then for $k \geq 1$, Algorithm 1 can be rewritten as

$$x_i(k+1) = \tilde{x}_i(k) + \chi_{i,k}\omega_i(k) \tag{10}$$

where $\omega_i(k) = P_X(\tilde{x}_i(k) - (1/r)d_i(k)) - \tilde{x}_i(k)$, $\chi_{i,k} = 1$ if $k = \kappa_i(r)$ for some r and $\chi_{i,k} = 0$ otherwise.

In this proof, we still denote by L the upper bound of subgradients of agents' objective functions on X, i.e., $L := \sup_{q\in\bigcup_{i,x\in X}\partial f_i(x)} |q|$, which is a finite number because of the boundedness of X and the convexity of f_i. This implies that Assumption 3 holds. Therefore,

$$|\omega_i(k)| \leq \frac{1}{r}|d_i(k)| \leq \frac{1}{r}L$$

and then it follows from Assumption 4 that $\lim_{k\to\infty} |\omega_i(k)| = 0$.[3] As a result, the network achieves a consensus by [31, Th. 1], i.e., $\lim_{k\to\infty} h(k) = 0$, where $h(k) := \max_{1\leq i,j\leq n} |x_i(k) - x_j(k)|$.

Take freely $x^* \in \arg\min_X \sum_{i=1}^n f_i (= \arg\min \sum_{i=1}^n f_i)$ and denote

$$\eta_i(k) = |x_i(k) - x^*|^2.$$

By applying the similar arguments for distributed subgradient algorithms in [1], [2], and [10], we have that when $k = \kappa_i(r)$ for some r

$$\begin{aligned}
\eta_i(k+1) &= \left| P_X\left(\tilde{x}_i(k) - \frac{1}{r}d_i(k)\right) - x^* \right|^2 \\
&\leq \left| \tilde{x}_i(k) - \frac{1}{r}d_i(k) - x^* \right|^2 \\
&\leq \left|\tilde{x}_i(k) - x^*\right|^2 + \frac{|d_i(k)|^2}{r^2} - \frac{2}{r}\big(x_i(k) - x^*\big)' d_i(k) \\
&\quad + \frac{2L}{r}|x_i(k) - \tilde{x}_i(k)| \\
&\leq \sum_{j\in\mathcal{N}_i} a_{ij}\eta_j(k) + \frac{L^2}{r^2} - \frac{2}{r}\big(f_i(\bar{x}(k)) - f_i\big(x^*\big)\big) \\
&\quad + \frac{4L}{r}h(k)
\end{aligned}$$

where $\bar{x}(k) := (1/n)\sum_{i=1}^n x_i(k)$ denotes the average of agents' estimates at time k. Moreover, when $k \neq \kappa_i(r)$ for any r, we have $\eta_i(k+1) \leq \sum_{j\in\mathcal{N}_i} a_{ij}\eta_j(k)$. By the above two cases and the double stochasticity in Assumption 2, we have

$$\begin{aligned}
\sum_{i=1}^n \eta_i(k+1) \leq \sum_{i=1}^n \eta_i(k) + \sum_{i=1}^n \chi_{i,k}\bigg(&\frac{L^2}{r^2} - \frac{2}{r}\big(f_i(\bar{x}(k)) - f_i\big(x^*\big)\big) \\
&+ \frac{4L}{r}h(k)\bigg).
\end{aligned}$$

[3] We use the property of convex projection operator: $|P_X(y) - z| \leq |y - z|$ for any $y \in \mathbb{R}^m$ and $z \in X$, which comes from [2, Lemma 1(b)].

Let T be the least common multiple of $T_i, i = 1, \ldots, n$ given in Assumption 4. Then we get

$$\sum_{i=1}^{n} \eta_i((s+1)T) \le \sum_{i=1}^{n} \eta_i(sT) + \sum_{i=1}^{3} \mu_i(s) \tag{11}$$

where

$$\mu_1(s) = \sum_{k=sT}^{(s+1)T-1} \sum_{i=1}^{n} \chi_{i,k} \frac{L^2}{r^2}$$
$$\mu_2(s) = -\sum_{k=sT}^{(s+1)T-1} \sum_{i=1}^{n} \chi_{i,k} \frac{2}{r} \big(f_i(\bar{x}(k)) - f_i\big(x^*\big)\big)$$
$$\mu_3(s) = \sum_{k=sT}^{(s+1)T-1} \sum_{i=1}^{n} \chi_{i,k} \frac{4L}{r} h(k).$$

We next estimate the sum of $\mu_i(s), i = 1, 2, 3$ over $s \ge 1$. Define $\nu_i(s) = |\{r | sT \le \kappa_i(r) < (s+1)T\}|$. By Assumption 4, $\nu_i(s_1) = \nu_i(s_2), \forall s_1, s_2$. Thus, $\Delta_i(s) := \nu_i(0) + \cdots + \nu_i(s-1) = \nu_i(0)s$. We have

$$\sum_{s=1}^{\infty} \mu_1(s) = \sum_{s=1}^{\infty} \sum_{i=1}^{n} \sum_{r=1}^{\nu_i(s)} \frac{L^2}{(\Delta_i(s)+r)^2} \le \sum_{s=1}^{\infty} \sum_{i=1}^{n} \frac{\nu_i(s) L^2}{(\Delta_i(s))^2} = \sum_{s=1}^{\infty} \frac{1}{s^2} \sum_{i=1}^{n} \frac{L^2}{\nu_i(0)} < \infty. \tag{12}$$

We also have

$$\begin{aligned}
&\mu_2(s) + \frac{2}{s} \sum_{i=1}^{n} \big(f_i(\bar{x}(sT)) - f_i\big(x^*\big)\big) \\
&= -\sum_{i=1}^{n} \sum_{r=1}^{\nu_i(s)} \frac{2\big[f_i(\bar{x}(\kappa_i(\Delta_i(s)+r))) - f_i(\bar{x}(sT))\big]}{\Delta_i(s)} \\
&\quad - \sum_{i=1}^{n} \sum_{r=1}^{\nu_i(s)} \left(\frac{2}{\Delta_i(s)+r} - \frac{2}{\Delta_i(s)} \right) \\
&\quad \times \big(f_i(\bar{x}(\kappa_i(\Delta_i(s)+r))) - f_i(x^*)\big) \\
&\le \sum_{i=1}^{n} \frac{2L\nu_i(s)}{\Delta_i(s)} \Gamma(s) + \sum_{i=1}^{n} \sum_{r=1}^{\nu_i(s)} \frac{2L\zeta r}{(\Delta_i(s))^2}
\end{aligned} \tag{13}$$

where $\Gamma(s) := \max_{sT \le r < (s+1)T} |\bar{x}(r) - \bar{x}(sT)|$, $\zeta := \sup_{s \ge 0} \max_{sT \le r < (s+1)T} |\bar{x}(r) - x^*| < \infty$ by the boundedness of X and the fact that $\bar{x}(r) \in X$. Taking the average of both sides of (10), by Assumption 2 we have $\bar{x}(k+1) = \bar{x}(k) + (1/n) \sum_{i=1}^{n} \chi_{i,k} \omega_i(k)$ and then

$$\Gamma(s) \le \sum_{k=sT}^{(s+1)T-2} \frac{1}{n} \sum_{i=1}^{n} \chi_{i,k} |\omega_i(k)| \le \frac{1}{n} \sum_{i=1}^{n} \sum_{r=1}^{\nu_i(s)} \frac{L}{\Delta_i(s)+r} \le \frac{L}{s}.$$

This implies

$$\sum_{s=1}^{\infty} \sum_{i=1}^{n} \frac{2L\nu_i(s)}{\Delta_i(s)} \Gamma(s) \le \sum_{s=1}^{\infty} \frac{2L^2 n}{s^2} < \infty. \tag{14}$$

Moreover, we also have

$$\sum_{s=1}^{\infty} \sum_{i=1}^{n} \sum_{r=1}^{\nu_i(s)} \frac{2L\zeta r}{(\Delta_i(s))^2} \le \sum_{s=1}^{\infty} \frac{1}{s^2} \sum_{i=1}^{n} \frac{2L\zeta}{(\nu_i(0))^2} \frac{(1+\nu_i(0))\nu_i(0)}{2} < \infty. \tag{15}$$

Combining with (13)–(15) together, we get

$$\sum_{s=1}^{\infty} \left(\mu_2(s) + \frac{2}{s} \sum_{i=1}^{n} \big(f_i(\bar{x}(sT)) - f_i\big(x^*\big)\big) \right) < \infty. \tag{16}$$

By the similar arguments given in the proof of [2, Lemma 8] or [10, Lemma 4.3], we can also show that

$$\begin{aligned}
\sum_{s=1}^{\infty} \mu_3(s) &\le \sum_{s=1}^{\infty} \sum_{i=1}^{n} \sum_{r=1}^{\nu_i(s)} \frac{4L}{\Delta_i(s)+r} \max_{sT \le k < (s+1)T} h(k) \\
&\le \sum_{s=1}^{\infty} \sum_{i=1}^{n} \frac{4L\nu_i(s)}{\Delta_i(s)} \max_{sT \le k < (s+1)T} h(k) \\
&= \sum_{s=1}^{\infty} \frac{4Ln}{s} \max_{sT \le k < (s+1)T} h(k) < \infty.
\end{aligned} \tag{17}$$

By (11), (12), (16), and (17), we conclude that the limit $\lim_{s\to\infty} \sum_{i=1}^{n} |x_i(sT) - x^*|^2$ exists,

$$\sum_{s=1}^{\infty} \frac{1}{s} \sum_{i=1}^{n} \big(f_i(\bar{x}(sT)) - f_i\big(x^*\big)\big) < \infty$$

and thus $\liminf_{s\to\infty} \sum_{i=1}^{n} (f_i(\bar{x}(sT)) - f_i(x^*)) = 0$. Let $\{\bar{x}(s^r T)\}_{r \ge 0}$ be a subsequence of the bounded sequence $\{\bar{x}(sT)\}_{s \ge 0}$ such that $\lim_{r\to\infty} \sum_{i=1}^{n} (f_i(\bar{x}(s^r T)) - f_i(x^*)) = 0$ and the limit $\lim_{r\to\infty} \bar{x}(s^r T) = \hat{x}$ exists for some $\hat{x}$. Therefore, it follows from the continuity of f_i and the closedness of X that $\hat{x} \in \arg\min_X \sum_{i=1}^{n} f_i$. By replacing x^* with $\hat{x}$, we can similarly show that the limit $\lim_{s\to\infty} \sum_{i=1}^{n} |x_i(sT) - \hat{x}|^2$ exists. This combines with $\lim_{r\to\infty} \bar{x}(s^r T) = \hat{x}$ and what we have shown that the consensus is achieved imply that $\lim_{s\to\infty} x_i(sT) = \hat{x}$, $i = 1, \ldots, n$. By the proceeding relation and considering that $\lim_{k\to\infty} |\omega_i(k)| = 0$, we obtain $\lim_{k\to\infty} x_i(k) = \hat{x}$, $i = 1, \ldots, n$ with $\hat{x} \in \arg\min \sum_{i=1}^{n} f_i$.

We complete the proof. ■

Remark 6: In the designed stepsize, $c_i, i = 1, \ldots, n$ are taken as nonnegative constants. In fact, from the proof of the above theorem we can find that the convergence and optimality can also be ensured if these constants are generalized to time-varying cases provided that these time-varying constants are bounded. It can be seen that this generalization can further improve the level of privacy preservation.

Remark 7: We here present some discussions on the convergence rate of our algorithm. The convergence rate estimate is generally complicated because it depends on the property of objective functions, the choice of initial conditions, the algorithm parameters, and the network structure. But for the case of strongly convex objective functions and completely connected graphs with uniform weights, by the arguments in the

proof of Theorem 7 we can get the following convergence estimate:

$$\left|\bar{x}((k+1)T) - x^*\right|^2 \leq \left(1 - \frac{b}{k}\right)|\bar{x}(kT) - x^*|^2 + \frac{c}{k^2}$$

for some $b > 0$, $c > 0$, where x^* is the unique optimal solution, T is the least common multiple of $T_i, i = 1, \ldots, n$ given in Assumption 4.

C. Numerical Example

We here present a numerical example to illustrate the algorithm performance. We consider a network with 1000 agents. The network graph is connected, undirected and is generated by the Erdős-Rényi random graph model $G(200, 0.075)$, where 0.075 is the probability that each possible arc is included in this graph. The adjacency matrix $\bar{A}$ is taken as

$$\bar{A} = I_{1000} - \frac{1}{d_{\max} + 1}\mathcal{L}$$

where $d_{\max}$, $\mathcal{L}$ is the maximum degree and the Laplacian matrix of this network graph, respectively. The individual objective function of agent $i, i = 1, \ldots, 1000$ is

$$f_i(x) = x'Q_i x + d_i' x, \quad x \in \mathbb{R}^5$$

where $Q_i \in \mathbb{R}^{5\times 5}$ is a randomly generated symmetric positive definite matrix and $d_i \in \mathbb{R}^5$ is a random generated vector. Here, we do not consider the artificially introduced constrained set X, i.e., we take $X = \mathbb{R}^5$ in Algorithm 1. The initial conditions are taken as $x_i(0) = 0, i = 1, \ldots, 1000$.

We compare our algorithm 1 with the standard DSSHSA (2). Let us specify the stepsize setting for each algorithm.

1) *Our Algorithm:* The optimization time sequence is randomly generated before executing the algorithm. For 500 agents, it makes 100 optimization updates within anyone of a set of consecutive intervals with length 200 at some randomly generated time instances [i.e., for these agents $T_i = 200$, $t_i(s) = 100$ in Assumption 4], while for the left 500 agents, it makes 150 optimization updates within anyone of a set of consecutive intervals with length 200 at some randomly generated time instances [i.e., $T_i = 200, t_i(s) = 150$]. The privacy preservation constants are taken as zero and the stepsize of each agent is taken as the inverse of the total number of this agent's optimization update times up to the current time if the current time is this agent's optimization time.
2) *DSSHSA (2):* The stepsize is taken as $\alpha_k = 1/(k+1)$.

We take the following two measures as the algorithm performance index.

1) *The Sum of the Distance of All Agents' Estimates to the Optimal Solution:* $\sum_{i=1}^{1000} |x_i(k) - x^*|$, where x^* is the unique optimal solution of the sum objective $\min \sum_{i=1}^{1000} f_i$.
2) *Consensus Errors of Agents' Estimates:* $|x(k) - \bar{A} \otimes I_5 x(k)|$, where $x(k) = (x_1'(k), \ldots, x_{1000}'(k))' \in \mathbb{R}^{5000}$, $\otimes$ denotes the Kronecker product.

Figs. 1 and 2 show the trajectories of sum of the distance of all agents' estimates to the optimal solution (SDOS) and consensus errors of agents' estimates (COE) in our algorithm and DSSHSA (2), respectively. Table I shows the values of SDOS and COE at some particular time instances. The figures and the table show that our algorithm converges faster at the early stage of algorithm execution but slower after a period of time than DSSHSA (2). This illustration is intuitive since at the early stage agents are far from the optimal solution, more consensus iterations among agents before optimization operation will lead to faster convergence, while when agents are close to the optimal solution after a period of time, the consensus iteration is somehow no longer necessary and the optimization iteration will dominate the consensus iteration on the convergence performance. Although our algorithm converges slower than DSSHSA (2) after a period of time, our algorithm has the additional privacy-preserving property as shown in this paper.

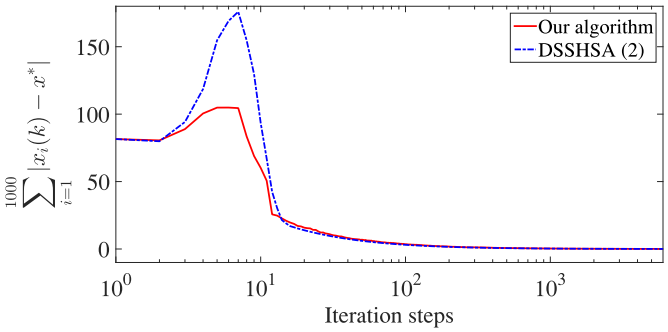


Fig. 1. Two trajectories of the measure SDOS in our algorithm and DSSHSA (2).

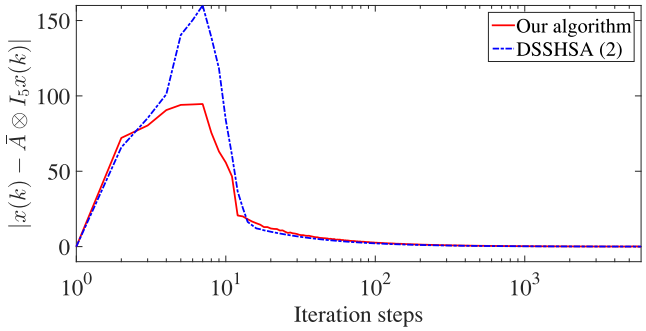


Fig. 2. Two trajectories of the measure COE in our algorithm and DSSHSA (2).

TABLE I
VALUES OF SDOS AND COE IN OUR ALGORITHM AND DSSHSA (2) AT SOME TIME INSTANCES

Iteration times	Our algorithm		DSSHSA (2)	
	SDOS	COE	SDOS	COE
0	81.5171	0	81.5171	0
50	6.9143	5.2207	6.0292	4.2075
100	3.5302	2.6010	3.1054	2.1532
200	1.7896	1.3485	1.5771	1.0896
500	0.7353	0.5525	0.6368	0.4391
1000	0.3678	0.2704	0.3193	0.2201
2000	0.1839	0.1383	0.1597	0.1102
4000	0.0916	0.0691	0.0797	0.0551
6000	0.0616	0.0457	0.0530	0.0368

V. Conclusion

In this paper, we considered the privacy-preserving features of distributed subgradient optimization algorithms. We first showed that the DSSHSA is not privacy preserving in the sense that the malicious agent can asymptotically discover other agents' subgradients for almost all adjacency matrices except a zero Lebesgue measure weight set. We also proposed a new distributed subgradient asynchronous heterogeneous-stepsize projection algorithm, in which agents make their own optimization updates asynchronously and the stepsizes are different among the agents. Compared with the existing privacy-preserving distributed algorithms, our algorithm allows agents to exchange estimates directly with their neighbors and does not employ any additional privacy-preserving technique. The introduced convex projection set and the asynchronous heterogeneous-stepsize optimization mechanism can effectively protect agents' subgradients. Moreover, we also showed the convergence and optimality of the newly proposed algorithm under mild assumption on the designed stepsize. Other interesting problems, including investigating privacy-preserving properties of other distributed optimization algorithms such as subgradient random algorithms [18]–[20], dual averaging algorithm [7], and ADMM [8], [9], and developing other privacy-preserving algorithms using the proposed techniques in this paper, are still under investigation.

Acknowledgment

The authors would like to thank Prof. Y. Hong, the Editor, the Associate Editor, the anonymous referees for their helpful comments and suggestions, which improve the quality of this paper.

References

[1] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[2] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[3] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

[4] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, Dec. 2010.

[5] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Trans. Automat. Control*, vol. 56, no. 6, pp. 1291–1306, Jun. 2011.

[6] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM J. Optim.*, vol. 20, no. 3, pp. 1157–1170, 2010.

[7] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[8] 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, Jul. 2011.

[9] 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.

[10] Y. Lou, Y. Hong, L. Xie, G. Shi, and K. H. Johansson, "Nash equilibrium computation in subnetwork zero-sum games with switching communications," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2920–2935, Oct. 2016.

[11] Y. Lou, G. Shi, K. H. Johansson, and Y. Hong, "Approximate projected consensus for convex intersection computation: Convergence analysis and critical error angle," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1722–1736, Jul. 2014.

[12] Y. Lou, Y. Hong, and S. Wang, "Distributed continuous-time approximate projection protocols for shortest distance optimization problems," *Automatica*, vol. 69, no. 7, pp. 289–297, Jul. 2016.

[13] Y. Lou, G. Shi, K. H. Johansson, and Y. Hong, "Convergence of random sleep algorithms for optimal consensus," *Syst. Control Lett.*, vol. 62, no. 12, pp. 1196–1202, Dec. 2013.

[14] G. Shi, K. H. Johansson, and Y. Hong, "Reaching an optimal consensus: Dynamical systems that compute intersections of convex sets," *IEEE Trans. Autom. Control*, vol. 58, no. 3, pp. 610–622, Mar. 2013.

[15] G. Shi and K. H. Johansson, "Randomized optimal consensus of multi-agent systems," *Automatica*, vol. 48, no. 12, pp. 3018–3030, Dec. 2012.

[16] J. Wang and N. Elia, "Control approach to distributed optimization," in *Proc. Allerton Conf. Commun. Control Comput.*, Monticello, IL, USA, 2010, pp. 557–561.

[17] J. Lu and C. Y. Tang, "Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2348–2354, Sep. 2012.

[18] D. Jakovetić, D. Bajović, N. Krejić, and N. Krklec-Jerinkić, "Distributed gradient methods with variable number of working nodes," *IEEE Trans. Signal Process.*, vol. 64, no. 15, pp. 4080–4095, Aug. 2016.

[19] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[20] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1337–1351, Jun. 2011.

[21] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, May 2015.

[22] Q. Liu and J. Wang, "A second-order multi-agent network for bound-constrained distributed optimization," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3310–3315, Dec. 2015.

[23] H. Wang, X. Liao, T. Huang, and C. Li, "Cooperative distributed optimization in multiagent networks with delays," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 2, pp. 363–369, Feb. 2015.

[24] S. Yang, Q. Liu, and J. Wang, "Distributed optimization based on a multiagent system in the presence of communication delays," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 5, pp. 717–728, May 2017.

[25] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints via primal-dual subgradient methods," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.

[26] D. Yuan, S. Xu, and H. Zhao, "Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1715–1724, Dec. 2011.

[27] D. Yuan, D. W. C. Ho, and S. Xu, "Regularized primal–dual subgradient method for distributed constrained optimization," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2109–2118, Sep. 2016.

[28] X. Wang, Y. Hong, and H. Ji, "Distributed optimization for a class of nonlinear multiagent systems with disturbance rejection," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1655–1666, Jul. 2016.

[29] P. Yi and Y. Hong, "Quantized subgradient algorithm and data-rate analysis for distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 4, pp. 380–392, Dec. 2014.

[30] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.

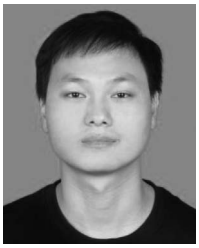
[31] L. Wang and L. Guo, "Robust consensus and soft control of multi-agent systems with noises," *J. Syst. Sci. Complex.*, vol. 21, no. 3, pp. 406–415, Sep. 2008.

[32] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Dallas, TX, USA, 2000, pp. 439–450.

[33] J. Vaidya, C. Clifton, and M. Zhu, *Privacy-Preserving Data Mining*. Heidelberg, Germany: Springer-Verlag, 2005.

[34] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2483–2493, Nov. 2013.

[35] C. Li, P. Zhou, G. Chen, and Y. Jiang. *Differentially Private Distributed Online Learning*. Accessed on May 2015. [Online]. Available: http://arxiv.org/abs/1505.06556
[36] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 50–64, Jan. 2017.
[37] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. Int. Conf. Distrib. Comput. Netw.*, Goa, India, 2015, Art. no. 4.
[38] M. T. Hale and M. Egerstedt, "Differentially private cloud-based multi-agent optimization with constraints," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, 2015, pp. 1235–1240.
[39] E. Nozari, P. Tallapragada, and J. Cortés. *Differentially Private Distributed Convex Optimization Via Functional Perturbation*. Accessed on Dec. 2015. [Online]. Available: https://arxiv.org/abs/1512.00369



Youcheng Lou received the B.Sc. degree in mathematics and applied mathematics from Zhengzhou University, Zhengzhou, China, in 2008 and the Ph.D. degree in complex systems and control from the Academy of Mathematics and Systems Science (AMSS), Chinese Academy of Sciences (CAS), Beijing, China, in 2013.

From 2013, he has been a Post-Doctoral Researcher with the National Center for Mathematics and Interdisciplinary Sciences, AMSS, CAS. From 2016, he is a Post-Doctoral Fellow with the Department of Systems Engineering and Engineering Management, Chinese University of Hong Kong, Hong Kong, funded by Hong Kong Scholars Program. His current research interests include multiagent networks, distributed optimization and the applications of distributed methods in operations research, economics, and finance.

Lean Yu (M'11) received the Ph.D. degree in management science and engineering from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Professor with the School of Economics and Management, Beijing University of Chemical Technology, Beijing. He has published over 70 papers in journals including the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the *Decision Support Systems*, and *Information Sciences*. His current research interests include computational intelligence, computer simulation, decision support systems, data mining, and financial forecasting.

Dr. Yu was a recipient of the China National Science Funds for Distinguished Young Scholars and the National Program for Support of Top-Notch Young Professionals.

Shouyang Wang received the Ph.D. degree in operations research from the Institute of Systems Science, Chinese Academy of Sciences (CAS), Beijing, China, in 1986.

He is currently a Bairen Distinguished Professor with the Academy of Mathematics and Systems Science, CAS and a Changjiang Chair Professor with the School of Economics and Management, University of Chinese Academy of Sciences, Beijing. He is the President of International Society of Knowledge and Systems Sciences, and has published over 30 monographs and 250 papers in leading journals including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the *Journal of Econometrics*, the *Journal of Banking and Finance*, and *Quantitative Finance*. His current research interests include data mining, supply chain management, decision support systems, and financial engineering.

Peng Yi received the B.Sc. degree in automation from the University of Science and Technology of China, Hefei, China, in 2011 and the Ph.D. degree in operations research and cybernetic from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2016.

He is currently a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. His current research interest include multiagent systems, distributed optimization, game theory, and smart grid.