Dynamics based Privacy Preservation in Decentralized Optimization ⋆

Huan Gao [a,1], Yongqiang Wang [b,2], Angelia Nedić [c]

[a] *School of Automation, Northwestern Polytechnical University, Xi'an 710129, China*

[b] *Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA*

[c] *School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85281, USA*

Abstract

With decentralized optimization having increased applications in various domains ranging from machine learning, control, to robotics, its privacy is also receiving increased attention. Existing privacy solutions for decentralized optimization achieve privacy by patching information-technology privacy mechanisms such as differential privacy or homomorphic encryption, which either sacrifices optimization accuracy or incurs heavy computation/communication overhead. We propose an inherently privacy-preserving decentralized optimization algorithm by exploiting the robustness of decentralized optimization dynamics. More specifically, we present a general decentralized optimization framework, based on which we show that the privacy of participating nodes' gradients can be protected by adding randomness in optimization parameters. We further show that the added randomness has no influence on the accuracy of optimization, and prove that our inherently privacy-preserving algorithm has R-linear convergence when the global objective ction is smooth and strongly convex. We also prove that the proposed algorithm can avoid the gradient of a node from being inferable by other nodes. Simulation results confirm the theoretical predictions.

Key words: Privacy preservation; decentralized optimization.

1 Introduction

Decentralized optimization has received increased attention due to its vast applications in online learning (Yan et al. 2012), distributed sensing (Bazerque & Giannakis 2010), formation control (Raffard et al. 2004), source localization (Zhang & Wang 2018*a*), and power system control (Gan et al. 2013). In many of these applications, a network of nodes collectively solve the following problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) \triangleq \sum_{i=1}^n f_i(\mathbf{x}), \quad f_i : \mathbb{R}^d \to \mathbb{R} \qquad (1)$$

where f_i is node i's local and private objective function.

Over the past decade, a number of gradient-based algorithms have been developed to solve the problem. Early results include the decentralized subgradient (DGD) algorithm (Nedić & Ozdaglar 2009) which combines average consensus with (sub)gradient descent under diminishing stepsizes. Its convergence rate is $\mathcal{O}((\ln k)/\sqrt{k})$ for general convex functions and $\mathcal{O}((\ln k)/k)$ for strongly convex functions. (Yuan et al. 2016) shows that the convergence rate of DGD can be improved under a fixed stepsize but at the expense of optimization accuracy. To guarantee both fast convergence and exact solution under fixed stepsizes, many algorithms propose to replace the local gradients in DGD with an auxiliary variable which tracks the global gradient, with typical examples including Aug-DGM (Xu et al. 2015), DIGing (Qu & Li 2017, Nedić, Olshevsky & Shi 2017), ATC-DIGing (Nedić, Olshevsky, Shi & Uribe 2017), AsynDGM (Xu et al. 2017), $\mathcal{AB}$ (Xin & Khan 2018), and Push-Pull (Pu et al. 2018, Du et al. 2018, Zhang et al. 2019). These algorithms can achieve R-linear convergence [3]. Convergence of such gradient-tracking based algorithms on time-varying graphs have also been discussed in Nedić, Olshevsky & Shi (2017),

⋆ The work was supported in part by the National Science Foundation under Grants ECCS-1912702, CCF-2106293, CCF-2106336, CCF-2215088, and CNS-2219487.

Email addresses: huangao@nwpu.edu.cn (Huan Gao), yongqiw@clemson.edu (Yongqiang Wang), angelia.nedich@asu.edu (Angelia Nedić).

[1] Huan Gao is now with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China. He was with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA. The work was done when Huan Gao was with Clemson University.

[2] Corresponding author.

[3] For a sequence $\{\mathbf{x}^k\}$ converging to $\mathbf{x}^*$ under some norm $||\cdot||$, the convergence is R-linear if there exist constants c and $\rho \in (0,1)$ such that $||\mathbf{x}^k - \mathbf{x}^*|| \leq c\rho^k$ holds for any k (Nedić, Olshevsky & Shi 2017).

Xu et al. (2017), and Saadatniaki et al. (2020). Other relevant algorithms include Shi et al. (2015), Xi et al. (2018), Nedić & Ozdaglar (2015), Hale & Egerstedt (2017), Fazlyab et al. (2018).

However, none of the aforementioned algorithms consider the privacy of individual nodes, which is unacceptable in many applications. For example, in the rendezvous problem where a group of nodes use decentralized optimization to agree on the optimal assembly position, individual nodes may want to keep their initial positions private in hostile environments. As indicated in Huang et al. (2015), without protection by an appropriate privacy mechanism, a node's initial position can be easily inferred by an adversary in rendezvous algorithms. Another example is collaborative machine learning where gradients/model updates exchanged among participating machines may contain sensitive information such as personal medical record and salary (Yan et al. 2012).

Recently, results have emerged on privacy-preserving decentralized optimization. For example, differential-privacy based approaches are proposed in Nozari et al. (2016), Huang et al. (2015), and Wang & Nedić (2022). However, except Wang & Nedić (2022) which retains almost sure convergence to the optimal solution, all such approaches will unavoidably compromise the accuracy of optimization results. To enable privacy protection with guaranteed optimization accuracy, partially homomorphic encryption based approaches have been proposed in our own prior results (Zhang & Wang 2018*b*, Zhang et al. 2018) as well as others' (Lu & Zhu 2018). However, such approaches will incur heavy computation and communication overhead. Yan et al. (2012) and Lou et al. (2018) showed that privacy can be obtained by incorporating a projection step or injecting constant uncertainties in stepsizes. However, both approaches have limitations in privacy protection: projection based defense requires individual agents to have a priori knowledge of the optimal solution, whereas constant uncertainties in stepsizes are unable to cover arbitrarily large variations on the gradients. Other approaches include Gade & Vaidya (2018), Li et al. (2020), and Wang & Poor (2022). However, they are only applicable to undirected graphs.

Through using random coefficients and/or initial conditions, others as well as our group have proposed several private consensus algorithms (Manitara & Hadjicostis 2013, Charalambous et al. 2019, Pilet et al. 2019, Hadjicostis & Dominguez-Garcia 2020, Kia et al. 2015, Mo & Murray 2017, Gupta et al. 2017, He et al. 2018, Gao et al. 2018, Ruan et al. 2019, Ridgley et al. 2019). Inspired by this line of research, in this paper, we propose to protect the gradients of participating nodes in decentralized optimization by leveraging the robustness of decentralized optimization dynamics. More specifically, by judiciously injecting uncertainties in optimization dynamics, we obfuscate exchanged information without affecting convergence to the exact optimal solution. We prove that the proposed algorithm can avoid the gradient of a node from being inferable by other nodes. Since protecting the gradient means protecting the values of the gradient function over the entire domain (or protecting both function types and function parameters), it is much more challenging than protecting a single initial value in the private consensus problem.

The main contributions of the paper are as follows: 1) We propose a dynamics based privacy approach for decentralized optimization that neither sacrifices accuracy nor incurs heavy computation/communication overhead. This is in distinct difference from existing approaches based on differential privacy (which compromise optimization accuracy) and approaches based on homomorphic encryption (which incur heavy computation and communication overhead); 2) Our approach is also different from the multi-party secure computation approach in He et al. (2020) which requires each node to communicate with two non-colluding external servers. By introducing randomness into interaction parameters, our approach is implementable in a fully decentralized manner without the assistance of any external servers; 3) We propose a new privacy definition based on the indistinguishability of gradient's arbitrary variations to adversaries from the viewpoint of accessible information, which is stricter than the unobservability/unsolvability based privacy definitions; 4) To facilitate the dynamics based privacy design, we propose a general framework for gradient-tracking based decentralized optimization which includes as special cases many existing algorithms, such as Aug-DGM (Xu et al. 2015), DIGing (Qu & Li 2017, Nedić, Olshevsky & Shi 2017), ATC-DIGing (Nedić, Olshevsky, Shi & Uribe 2017), AsynDGM (Xu et al. 2017), $\mathcal{AB}$ (Xin & Khan 2018), and Push-Pull (Pu et al. 2018, Du et al. 2018, Zhang et al. 2019); 5) We prove that despite the time-varying randomness injected into interaction parameters and the general framework, our approach can still maintain R-linear convergence when the global objective function is strongly convex.

Notations: $\mathbb{R}$ and $\mathbb{Z}_{\geq 0}$ denote real numbers and nonnegative integers, respectively. $\mathbb{R}^n$ denotes the Euclidean space of dimension n. $\mathbf{0}_n \in \mathbb{R}^n$ and $\mathbf{0}_{m\times n} \in \mathbb{R}^{m\times n}$ denote zero vector and $m\times n$ zero matrix, respectively. $\mathbf{1}_n \in \mathbb{R}^n$ denotes the $n\times 1$ all-ones vector, $\mathbf{1}_{m\times n} \in \mathbb{R}^{m\times n}$ denotes the $m\times n$ all-ones matrix, and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix. $\|\cdot\|$ denotes the Euclidean norm of vectors and the spectral norm of matrices. $\otimes$ represents the Kronecker product.

2 Problem Formulation

We characterize the interaction as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the node set. $\mathcal{E} \subset \mathcal{V}\times\mathcal{V}$ is the set of edges, whose elements are such that an ordered pair (i, j) belongs to $\mathcal{E}$ if and only if there exists a directed link from node j to node i. We assume no self edges, i.e., $(i, i) \notin \mathcal{E}$ for all $i \in \mathcal{V}$. The out-neighbor set of node i is denoted as $\mathcal{N}_i^{out} = \{j \in \mathcal{V} \,|\, \forall\, (j, i) \in \mathcal{E}\}$. Similarly, the in-neighbor set of node i is $\mathcal{N}_i^{in} = \{j \in \mathcal{V} \,|\, \forall\, (i, j) \in \mathcal{E}\}$.

Assumption 1 *$\mathcal{G}$ is strongly connected, i.e., for any $i, j \in \mathcal{V}$*

Table I. Particular selections of parameters in the proposed framework to obtain some existing algorithms.

	$\mathbf{R}^k$	$\mathbf{A}^k$	$\mathbf{C}^k$	$\mathbf{B}^k$	$\mathbf{\Lambda}^k$
Aug-DGM (Xu et al. 2015)	$\mathbf{W}$	$\mathbf{W}$	$\mathbf{W}$	$\mathbf{W}$	$\mathbf{\Lambda}$
DIGing (Qu & Li 2017)	$\mathbf{W}$	$\mathbf{I}$	$\mathbf{W}$	$\mathbf{I}$	$\lambda\mathbf{I}$
DIGing (Nedić, Olshevsky & Shi 2017)	$\mathbf{W}^k$	$\mathbf{I}$	$\mathbf{W}^k$	$\mathbf{I}$	$\lambda\mathbf{I}$
ATC-DIGing (Nedić, Olshevsky, Shi & Uribe 2017)	$\mathbf{W}$	$\mathbf{W}$	$\mathbf{W}$	$\mathbf{W}$	$\mathbf{\Lambda}$
AsynDGM (Xu et al. 2017)	$\mathbf{W}^k$	$\mathbf{W}^k$	$\mathbf{W}^k$	$\mathbf{I}$	$\mathbf{\Lambda}$
$\mathcal{AB}$ (Xin & Khan 2018)	$\mathbf{R}$	$\mathbf{I}$	$\mathbf{C}$	$\mathbf{C}$	$\lambda\mathbf{I}$
Push-Pull (Pu et al. 2018)	$\mathbf{R}$	$\mathbf{R}$	$\mathbf{C}$	$\mathbf{C}$	$\lambda\mathbf{I}$
Push-Pull (Du et al. 2018)	$\mathbf{R}$	$\mathbf{I}$	$\mathbf{C}$	$\mathbf{I}$	$\lambda\mathbf{I}$
Push-Pull (Zhang et al. 2019)	$\mathbf{R}$	$\mathbf{R}$	$\mathbf{C}$	$\mathbf{I}$	$\lambda\mathbf{I}$

$\mathbf{W}$ and $\mathbf{W}^k$ are doubly stochastic, $\mathbf{R}$ is row-stochastic, $\mathbf{C}$ is column-stochastic, and $\lambda\mathbf{I}$ and $\mathbf{\Lambda}$ represent homogeneous and heterogeneous stepsize matrices, respectively.

with $i \neq j$, there exists one directed path from i to j in $\mathcal{G}$.

Definition 1 *f is α-strongly convex with $\alpha > 0$ if $(\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}'))^T(\mathbf{x} - \mathbf{x}') \geq \alpha\|\mathbf{x} - \mathbf{x}'\|^2$ holds for $\forall$ $\mathbf{x}$, $\mathbf{x}'$ in $\mathbb{R}^d$.*

Definition 2 *f is β-smooth with $\beta > 0$ if $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\| \leq \beta\|\mathbf{x} - \mathbf{x}'\|$ holds for $\forall$ $\mathbf{x}$, $\mathbf{x}'$ in $\mathbb{R}^d$.*

Assumption 2 *Each f_i is convex and β_i-smooth. The global objective function $F = \sum_{i=1}^n f_i(\mathbf{x})$ is α_F-strongly convex.*

Under Assumption 2, F is β_F-smooth with $\beta_F = \sum_{i=1}^n \beta_i$. And problem (1) has a unique solution, denoted as $\mathbf{x}^*$.

3 A General Decentralized Optimization Framework

3.1 A New Framework for Decentralized Optimization

In this paper, we propose to enable privacy preservation in decentralized optimization by exploiting the robustness of decentralized optimization dynamics. To this end, we first propose a new framework for gradient-tracking based decentralized optimization in Algorithm 1.

By setting $\mathbf{R}_{ij}^k = \mathbf{A}_{ij}^k = \mathbf{0}_{d\times d}$ for $j \notin \mathcal{N}_i^{in} \cup \{i\}$ and $\mathbf{C}_{ji}^k = \mathbf{B}_{ji}^k = \mathbf{0}_{d\times d}$ for $j \notin \mathcal{N}_i^{out} \cup \{i\}$, (2) and (3) become

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \textstyle\sum_{j=1}^n \big(\mathbf{R}_{ij}^k\mathbf{x}_j^k - \mathbf{A}_{ij}^k\mathbf{\Lambda}_j^k\mathbf{y}_j^k\big) \\ \mathbf{y}_i^{k+1} &= \textstyle\sum_{j=1}^n \big(\mathbf{C}_{ij}^k\mathbf{y}_j^k + \mathbf{B}_{ij}^k\big(\nabla f_j^{k+1} - \nabla f_j^k\big)\big) \end{aligned} \tag{4}$$

where ∇f_j^k represents $\nabla f_j(\mathbf{x}_j^k)$. Further rewrite (4) as

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{R}^k\mathbf{x}^k - \mathbf{A}^k\mathbf{\Lambda}^k\mathbf{y}^k \\ \mathbf{y}^{k+1} &= \mathbf{C}^k\mathbf{y}^k + \mathbf{B}^k\big(\nabla f^{k+1} - \nabla f^k\big) \end{aligned} \tag{5}$$

where $\mathbf{x}^k = [(\mathbf{x}_1^k)^T \cdots (\mathbf{x}_n^k)^T]^T$, $\mathbf{y}^k = [(\mathbf{y}_1^k)^T \cdots (\mathbf{y}_n^k)^T]^T$, $\nabla f^k = [(\nabla f_1^k)^T \cdots (\nabla f_n^k)^T]^T$, and $\mathbf{R}^k$, $\mathbf{A}^k$, $\mathbf{C}^k$, and $\mathbf{B}^k$ are block matrices with the (ij)-th block entry being $\mathbf{R}_{ij}^k$, $\mathbf{A}_{ij}^k$, $\mathbf{C}_{ij}^k$, and $\mathbf{B}_{ij}^k$, respectively. $\mathbf{\Lambda}^k$ is a block diagonal matrix with the i-th diagonal block being $\mathbf{\Lambda}_i^k$.

Algorithm 1 A new decentralized optimization framework

Each node i randomly initializes $\mathbf{x}_i^0$ in $\mathbb{R}^d$ and sets $\mathbf{y}_i^0 = \nabla f_i(\mathbf{x}_i^0)$. At iteration k:

1: Node i computes and sends $\mathbf{x}_i^k$ as well as $\mathbf{\Lambda}_i^k\mathbf{y}_i^k$ to its out-neighbors $l \in \mathcal{N}_i^{out}$, where $\mathbf{\Lambda}_i^k$ is a diagonal matrix denoting the stepsize.

2: After receiving $\mathbf{x}_j^k$ and $\mathbf{\Lambda}_j^k\mathbf{y}_j^k$ from its in-neighbors $j \in \mathcal{N}_i^{in}$, node i updates $\mathbf{x}_i$ as:

$$\mathbf{x}_i^{k+1} = \textstyle\sum_{j\in\mathcal{N}_i^{in}\cup\{i\}} \big(\mathbf{R}_{ij}^k\mathbf{x}_j^k - \mathbf{A}_{ij}^k\mathbf{\Lambda}_j^k\mathbf{y}_j^k\big) \tag{2}$$

where $\mathbf{R}_{ij}^k$ and $\mathbf{A}_{ij}^k$ are coupling weight matrices.

3: After updating $\mathbf{x}_i$, node i computes and sends $\mathbf{C}_{li}^k\mathbf{y}_i^k + \mathbf{B}_{li}^k\big(\nabla f_i(\mathbf{x}_i^{k+1}) - \nabla f_i(\mathbf{x}_i^k)\big)$ to its out-neighbors $l \in \mathcal{N}_i^{out}$, where $\mathbf{C}_{li}^k$ and $\mathbf{B}_{li}^k$ are coupling weight matrices.

4: After receiving $\mathbf{C}_{ij}^k\mathbf{y}_j^k + \mathbf{B}_{ij}^k\big(\nabla f_j(\mathbf{x}_j^{k+1}) - \nabla f_j(\mathbf{x}_j^k)\big)$ from its in-neighbors $j \in \mathcal{N}_i^{in}$, node i updates $\mathbf{y}_i$ as:

$$\begin{aligned} \mathbf{y}_i^{k+1} = &\textstyle\sum_{j\in\mathcal{N}_i^{in}\cup\{i\}} \big(\mathbf{C}_{ij}^k\mathbf{y}_j^k \\ &+ \mathbf{B}_{ij}^k\big(\nabla f_j(\mathbf{x}_j^{k+1}) - \nabla f_j(\mathbf{x}_j^k)\big)\big) \end{aligned} \tag{3}$$

3.2 Relationship with Existing Algorithms

The proposed decentralized optimization framework includes as special cases many popular decentralized optimization algorithms. Table I summarizes the particular selections of parameters $\mathbf{R}^k$, $\mathbf{A}^k$, $\mathbf{B}^k$, $\mathbf{C}^k$, and $\mathbf{\Lambda}^k$ in (5) to obtain some commonly used algorithms. In fact, by setting $\mathbf{R}^k$, $\mathbf{A}^k$, $\mathbf{C}^k$, $\mathbf{B}^k$, and $\mathbf{\Lambda}^k$ to $\mathbf{R}$, $\mathbf{I}$, $\mathbf{C}$, $\mathbf{I}$, and $\mathbf{\Lambda}$, respectively, our proposed algorithm in (5) can be rewritten as $\mathbf{x}^{k+1} = (\mathbf{R} + \mathbf{C})\mathbf{x}^k - \mathbf{C}\mathbf{R}\mathbf{x}^{k-1} - \mathbf{\Lambda}\big(\nabla f^k - \nabla f^{k-1}\big)$ which becomes EXTRA (Shi et al. 2015).

Besides reducing to existing algorithms when the parameters are selected appropriately, our proposed algorithm can also give rise to new algorithms with special properties. Next, we show that it results in new algorithms having inherent privacy-preserving capabilities.

Table II. Parameter design for each node i in our proposed framework.

	Iterations $k < K$ †	Iterations $k \geq K$
$\mathbf{\Lambda}_i^k$	$\mathbf{\Lambda}_i^k = \text{diag}\{\lambda_i^k(1), \cdots, \lambda_i^k(d)\}$, where $\lambda_i^k(1), \ldots, \lambda_i^k(d)$ are chosen following selected distributions with support $\mathbb{R}$	$\mathbf{\Lambda}_i^k = \lambda \mathbf{I}_d$, $\lambda > 0$
$\mathbf{R}_{ij}^k$	$\mathbf{R}_{ij}^k = \text{diag}\{r_{ij}^k(1), \cdots, r_{ij}^k(d)\}$, where $r_{ij}^k(l)$ are chosen from $\mathbb{R}$ for $j \in \mathcal{N}_i^{in} \cup \{i\}$ and $l = 1, \ldots, d$ following selected distributions with support $\mathbb{R}$	$\mathbf{R}_{ij}^k = r_{ij}^k \mathbf{I}_d$, where r_{ij}^k are selected from $[\eta, 1]$ for $j \in \mathcal{N}_i^{in} \cup \{i\}$ subject to $\sum_{j=1}^n r_{ij}^k = 1$
$\mathbf{A}_{ij}^k$	$\mathbf{A}_{ij}^k = \text{diag}\{a_{ij}^k(1), \cdots, a_{ij}^k(d)\}$, where $a_{ij}^k(l)$ are chosen from $\mathbb{R}$ for $j \in \mathcal{N}_i^{in} \cup \{i\}$ and $l = 1, \ldots, d$ following selected distributions with support $\mathbb{R}$	$\mathbf{A}_{ij}^k = a_{ij}^k \mathbf{I}_d$, where a_{ij}^k are selected from $[\eta, 1]$ for $j \in \mathcal{N}_i^{in} \cup \{i\}$ subject to $\sum_{j=1}^n a_{ij}^k = 1$
$\mathbf{C}_{ji}^k$	$\mathbf{C}_{ji}^k = \text{diag}\{c_{ji}^k(1), \cdots, c_{ji}^k(d)\}$, where $c_{ji}^k(l)$ are chosen from $\mathbb{R}$ for $j \in \mathcal{N}_i^{out}$ and $l = 1, \ldots, d$ following selected distributions with support $\mathbb{R}$, and $\mathbf{C}_{ii}^k$ is set as $\mathbf{C}_{ii}^k = \mathbf{I}_d - \sum_{j \in \mathcal{N}_i^{out}} \mathbf{C}_{ji}^k$	$\mathbf{C}_{ji}^k = c_{ji}^k \mathbf{I}_d$, where c_{ji}^k are selected from $[\eta, 1]$ for $j \in \mathcal{N}_i^{out} \cup \{i\}$ subject to $\sum_{j=1}^n c_{ji}^k = 1$
$\mathbf{B}_{ji}^k$	$\mathbf{B}_{ji}^k = \text{diag}\{b_{ji}^k(1), \cdots, b_{ji}^k(d)\}$, where $b_{ji}^k(l)$ are chosen from $\mathbb{R}$ for $j \in \mathcal{N}_i^{out}$ and $l = 1, \ldots, d$ following selected distributions with support $\mathbb{R}$, and $\mathbf{B}_{ii}^k$ is set as $\mathbf{B}_{ii}^k = \mathbf{I}_d - \sum_{j \in \mathcal{N}_i^{out}} \mathbf{B}_{ji}^k$	$\mathbf{B}_{ji}^k = b_{ji}^k \mathbf{I}_d$ select b_{ji}^k from $[\eta, 1]$ for $j \in \mathcal{N}_i^{out} \cup \{i\}$ subject to $\sum_{j=1}^n b_{ji}^k = 1$

† K can be any positive integer. Its influence will be discussed in detail in Remark 2 and Remark 6. Its influence will be discussed in detail Its influence will be discussed in detail

4 Privacy-preserving Decentralized Optimization

4.1 Privacy-preserving Design

To enable privacy, we propose to add randomness in stepsize $\mathbf{\Lambda}^k$ and coupling weights $\mathbf{R}^k$, $\mathbf{A}^k$, $\mathbf{C}^k$, and $\mathbf{B}^k$ for iterations $k < K$, where K is a positive integer. The detailed parameter design for each node i is given in Table II. More specifically, for iterations $k < K$, each node i selects $\mathbf{C}_{ji}^k = \text{diag}\left\{c_{ji}^k(1), \ldots, c_{ji}^k(d)\right\}$ and $\mathbf{B}_{ji}^k = \text{diag}\left\{b_{ji}^k(1), \ldots, b_{ji}^k(d)\right\}$ for $j \in \mathcal{N}_i^{out}$ following any chosen random distributions with support $\mathbb{R}$ such as Gaussian or Laplace distribution (so the coupling weights can be negative, positive, or zero). Since we do not require $\mathbf{R}^k$ or $\mathbf{A}^k$ to be row-stochastic for iterations $k < K$, each node i can select $\mathbf{R}_{ij}^k$ and $\mathbf{A}_{ij}^k$ for $j \in \mathcal{N}_i^{in} \cup \{i\}$ following any random distributions.

Furthermore, as indicated in Table II, the column stochastic property of $\mathbf{C}^k$ and $\mathbf{B}^k$ and the row stochastic property of $\mathbf{R}^k$ and $\mathbf{A}^k$ are required for iterations $k \geq K$. Let us take $\mathbf{C}^k$ as an example to show how to ensure these properties in a fully decentralized manner. To meet the requirements of Table II, each node i selects a set of real values $\left\{c_{ji}^k \in [\eta, 1] \,\middle|\, j \in \mathcal{N}_i^{out} \cup \{i\}\right\}$ with their sum equal to one, then sets $\mathbf{C}_{ji}^k$ as $\mathbf{C}_{ji}^k = c_{ji}^k \mathbf{I}_d$ for $j \in \mathcal{N}_i^{out} \cup \{i\}$. Note that there are many ways to obtain such a set of real values with sum equal to one. For example, node i first randomly selects a set of real values $\left\{p_{ji}^k \,\middle|\, j \in \mathcal{N}_i^{out} \cup \{i\}\right\}$ from $[0, 1]$, then it sets c_{ji}^k by normalizing these values $\{p_{ji}^k\}$ via $c_{ji}^k = \frac{[1-(|\mathcal{N}_i^{out}|+1)\eta][(1-\eta)p_{ji}^k+\eta]}{(1-\eta)\sum_{l \in \mathcal{N}_i^{out} \cup \{i\}} p_{li}^k + (|\mathcal{N}_i^{out}|+1)\eta} + \eta$ for $j \in \mathcal{N}_i^{out} \cup \{i\}$. One can verify $\sum_{j \in \mathcal{N}_i^{out} \cup \{i\}} c_{ji}^k = 1$ and $c_{ji}^k \in [\eta, 1]$ for $j \in \mathcal{N}_i^{out} \cup \{i\}$, and the column stochastic property of $\mathbf{C}^k$ is guaranteed in a fully decentralized manner.

Remark 1 *We allow each agent i to randomly choose its associated coupling weights and stepsizes for iterations $k < K$ following any distributions. These distributions can be any continuous probability distribution with support $\mathbb{R}$, e.g., Gaussian distribution, Laplace distribution, etc.*

4.2 Convergence Analysis

Theorem 1 *Under Assumptions 1 and 2, and the parameter design in Table II, our algorithm has R-linear convergence when the stepsize parameter λ is sufficiently small.*

Proof: The proof is inspired by Saadatniaki et al. (2020). We first analyze the influence of randomly time-varying parameters in iterations $0 \leq k \leq K-1$. From Table II, we have $\sum_{i=1}^n \mathbf{C}_{ij}^k = \mathbf{I}_d$ and $\sum_{i=1}^n \mathbf{B}_{ij}^k = \mathbf{I}_d$, which, in combination with (4) leads to

$$\left(\sum_{i=1}^n \mathbf{y}_i^{k+1}\right)^T = \left(\sum_{i=1}^n (\mathbf{y}_i^k + \nabla f_i^{k+1} - \nabla f_i^k)\right)^T \quad (6)$$

for $k \leq K-1$. We can further rewrite (6) as $\sum_{i=1}^n \mathbf{y}_i^{k+1} - \sum_{i=1}^n \nabla f_i^{k+1} = \sum_{i=1}^n \mathbf{y}_i^k - \sum_{i=1}^n \nabla f_i^k$. Given $\mathbf{y}_i^0 = \nabla f_i^0$, one can obtain

$$\textstyle\sum_{i=1}^n \left(\mathbf{y}_i^K - \nabla f_i^K\right) = \cdots = \sum_{i=1}^n \left(\mathbf{y}_i^0 - \nabla f_i^0\right) = \mathbf{0}_d \quad (7)$$

The constraint (7) reflects the influence of random time-varying parameters in iterations $0 \leq k \leq K-1$. Next, under this constraint, we study the dynamics after iteration $K-1$.

For our parameter design in Table II, we have $\mathbf{\Lambda}_i^k = \lambda \mathbf{I}_d$, $\mathbf{R}_{ij}^k = r_{ij}^k \mathbf{I}_d$, $\mathbf{A}_{ij}^k = a_{ij}^k \mathbf{I}_d$, $\mathbf{C}_{ij}^k = c_{ij}^k \mathbf{I}_d$, and $\mathbf{B}_{ij}^k = b_{ij}^k \mathbf{I}_d$ for $k \geq K$. Constructing $n \times n$ matrices $\bar{\mathbf{R}}^k$, $\bar{\mathbf{A}}^k$, $\bar{\mathbf{C}}^k$, and $\bar{\mathbf{B}}^k$ with the (ij)-th elements being r_{ij}^k, a_{ij}^k, c_{ij}^k, and b_{ij}^k, respectively, we have $\mathbf{R}^k = \bar{\mathbf{R}}^k \otimes \mathbf{I}_d$, $\mathbf{A}^k = \bar{\mathbf{A}}^k \otimes \mathbf{I}_d$, $\mathbf{C}^k = \bar{\mathbf{C}}^k \otimes \mathbf{I}_d$, and $\mathbf{B}^k = \bar{\mathbf{B}}^k \otimes \mathbf{I}_d$ for $k \geq K$. Then for $k \geq K$, we can rewrite the system dynamics (5) as

$$\begin{aligned} \mathbf{x}^{k+1} &= (\bar{\mathbf{R}}^k \otimes \mathbf{I}_d)\mathbf{x}^k - \lambda(\bar{\mathbf{A}}^k \otimes \mathbf{I}_d)\mathbf{y}^k \\ \mathbf{y}^{k+1} &= (\bar{\mathbf{C}}^k \otimes \mathbf{I}_d)\mathbf{y}^k + (\bar{\mathbf{B}}^k \otimes \mathbf{I}_d)\left(\nabla f^{k+1} - \nabla f^k\right) \end{aligned} \quad (8)$$

Next, we introduce a state transformation, $\mathbf{s}^k = ((\bar{\mathbf{V}}^k)^{-1} \otimes \mathbf{I}_d)\mathbf{y}^k$, where $\bar{\mathbf{V}}^k = \text{diag}(\mathbf{v}^k)$ with the evolution of $\mathbf{v}^k$ governed by $\mathbf{v}^{k+1} = \bar{\mathbf{C}}^k\mathbf{v}^k$ for $k \geq K$. $\mathbf{v}^K$ is set as $\mathbf{v}^K = \frac{1}{n}\mathbf{1}_n$. Then for $k \geq K$, (8) can be rewritten as

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathbf{R}^k\mathbf{x}^k - \lambda(\bar{\mathbf{A}}^k\bar{\mathbf{V}}^k \otimes \mathbf{I}_d)\mathbf{s}^k \\ \mathbf{s}^{k+1} &= \mathbf{P}^k\mathbf{s}^k + ((\bar{\mathbf{V}}^{k+1})^{-1}\bar{\mathbf{B}}^k \otimes \mathbf{I}_d)\big(\nabla f^{k+1} - \nabla f^k\big)\end{aligned} \tag{9}$$

where $\mathbf{P}^k = \bar{\mathbf{P}}^k \otimes \mathbf{I}_d$ and $\bar{\mathbf{P}}^k = (\bar{\mathbf{V}}^{k+1})^{-1}\bar{\mathbf{C}}^k\bar{\mathbf{V}}^k$. Note that $\bar{\mathbf{P}}^k$ is row-stochastic and the sequence $\{\bar{\mathbf{P}}^k\}$ is ergodic for $k \geq K$. Furthermore, the sequence $\{\mathbf{v}^k\}$ for $k \geq K$ is an absolute probability sequence for $\{\bar{\mathbf{P}}^k\}$. We define $\bar{\mathbf{x}}_{\text{w}}^k = ((\boldsymbol{\phi}^k)^T \otimes \mathbf{I}_d)\mathbf{x}^k$, $\mathbf{r}^k = \mathbf{1}_n \otimes \bar{\mathbf{x}}_{\text{w}}^k - \mathbf{1}_n \otimes \mathbf{x}^*$, $\tilde{\mathbf{x}}_{\text{w}}^k = \mathbf{x}^k - \mathbf{1}_n \otimes \bar{\mathbf{x}}_{\text{w}}^k$, and $\tilde{\mathbf{s}}_{\text{w}}^k = \mathbf{s}^k - (\mathbf{1}_n(\mathbf{v}^k)^T \otimes \mathbf{I}_d)\mathbf{s}^k$, respectively, where for $k \geq K$, $\{\boldsymbol{\phi}^k\}$ is an absolute probability sequence for the ergodic sequence of row-stochastic matrices $\{\bar{\mathbf{R}}^k\}$. It can be verified that if the stepsize λ satisfies $\lambda \leq 1/\beta_F$, then

$$\|\mathbf{r}^{k+1}\| \leq \lambda n\bar{\beta}\|\tilde{\mathbf{x}}_{\text{w}}^k\| + \big(1 - \lambda n^{-1}\eta^{n-1}\alpha_F\big)\|\mathbf{r}^k\| + \lambda n\|\tilde{\mathbf{s}}_{\text{w}}^k\| \tag{10}$$

holds for $k \geq K$. Further combining with the constraint in (7), we have the following inequalities

$$\begin{aligned}\|\tilde{\mathbf{x}}_{\text{w}}^{k+1}\| &\leq \big(r_R + \lambda Q_R n\sqrt{n}\bar{\beta}\big)\|\tilde{\mathbf{x}}_{\text{w}}^{k-\bar{N}+1}\| \\ &+ \lambda Q_R n\sqrt{n}\bar{\beta}\Big(\sum_{l=0}^{\bar{N}-2}\|\tilde{\mathbf{x}}_{\text{w}}^{k-l}\| + \|\mathbf{r}^{k-\bar{N}+1}\| + \sum_{l=0}^{\bar{N}-2}\|\mathbf{r}^{k-l}\|\Big) \\ &+ \lambda Q_R\sqrt{n}\big(\|\tilde{\mathbf{s}}_{\text{w}}^{k-\bar{N}+1}\| + \sum_{l=0}^{\bar{N}-2}\|\tilde{\mathbf{s}}_{\text{w}}^{k-l}\|\big)\end{aligned} \tag{11}$$

and

$$\begin{aligned}&\|\tilde{\mathbf{s}}_{\text{w}}^{k+1}\| \\ &\leq \big(\frac{2n^2\bar{\beta}Q_P}{\eta^{n-1}} + \lambda\frac{n^3\bar{\beta}^2Q_P}{\eta^{n-1}}\big)\big(\|\tilde{\mathbf{x}}_{\text{w}}^{k-\bar{N}+1}\| + \sum_{l=0}^{\bar{N}-2}\|\tilde{\mathbf{x}}_{\text{w}}^{k-l}\|\big) \\ &\quad + \lambda\frac{n^3\bar{\beta}^2Q_P}{\eta^{n-1}}\big(\|\mathbf{r}^{k-\bar{N}+1}\| + \sum_{l=0}^{\bar{N}-2}\|\mathbf{r}^{k-l}\|\big) \\ &\quad + \big(r_P + \lambda\frac{n^2\bar{\beta}Q_P}{\eta^{n-1}}\big)\|\tilde{\mathbf{s}}_{\text{w}}^{k-\bar{N}+1}\| + \lambda\frac{n^2\bar{\beta}Q_P}{\eta^{n-1}}\sum_{l=0}^{\bar{N}-2}\|\tilde{\mathbf{s}}_{\text{w}}^{k-l}\|\end{aligned} \tag{12}$$

for $k \geq K + \bar{N} - 1$ where $\bar{\beta} = \max\{\beta_1, \cdots, \beta_n\}$, $Q_R = 2n(1 + \eta^{-(n-1)})/(1 - \eta^{n-1})$, $r_R = Q_R(1 - \eta^{n-1})^{\frac{N_R-1}{n-1}}$, $Q_P = 2n\big(1 + (n\eta^{-n})^{n-1}\big)/\big(1 - (n^{-1}\eta^n)^{n-1}\big)$, $r_P = Q_P\big(1 - (n^{-1}\eta^n)^{n-1}\big)^{\frac{N_P-1}{n-1}}$, and $\bar{N} = \max\{N_R, N_P\}$. Due to space limitation, here we omit the derivations of (10), (11), and (12). Please refer to Gao et al. (2022) for the detailed derivations.

Denoting $\boldsymbol{\xi}^k$ as $\boldsymbol{\xi}^k = \big[\|\tilde{\mathbf{x}}_{\text{w}}^k\|, \|\mathbf{r}^k\|, \|\tilde{\mathbf{s}}_{\text{w}}^k\|\big]^T$ and invoking (10), (11), and (12), we have the following inequality

$$\begin{bmatrix}\boldsymbol{\xi}^{k+1} \\ \vdots \\ \boldsymbol{\xi}^{k-\bar{N}+2}\end{bmatrix} \leq \underbrace{(\mathbf{M}^1 + \lambda\mathbf{M}^2)}_{\mathbf{M}(\lambda)}\begin{bmatrix}\boldsymbol{\xi}^k \\ \vdots \\ \boldsymbol{\xi}^{k-\bar{N}+1}\end{bmatrix} \tag{13}$$

for $k \geq K + \bar{N} - 1$ where $\mathbf{M}^1$ and $\mathbf{M}^2$ are given by

$$\mathbf{M}^1 = \begin{bmatrix}\mathbf{M}_a^1 & \mathbf{M}_b^1 & \cdots & \mathbf{M}_b^1 & \mathbf{M}_c^1 \\ \mathbf{I}_3 & & & & \\ & & \ddots & & \\ & & & & \mathbf{I}_3\end{bmatrix}, \quad \mathbf{M}^2 = \begin{bmatrix}\mathbf{M}_a^2 & \mathbf{M}_b^2 & \cdots & \mathbf{M}_b^2 & \mathbf{M}_c^2 \\ \mathbf{0}_{3\times3} & & & & \\ & & \ddots & & \\ & & & & \mathbf{0}_{3\times3}\end{bmatrix} \tag{14}$$

respectively, with

$$\begin{aligned}\mathbf{M}_a^1 &= \begin{bmatrix}0 & 0 & 0 \\ 0 & 1 & 0 \\ 2t & 0 & 0\end{bmatrix}, \quad \mathbf{M}_a^2 = \begin{bmatrix}n\bar{\beta}q & n\bar{\beta}q & q \\ n\bar{\beta} & -\eta^{n-1}m & n \\ n\bar{\beta}t & n\bar{\beta}t & t\end{bmatrix} \\ \mathbf{M}_b^1 &= \begin{bmatrix}0 & 0 & 0 \\ 0 & 0 & 0 \\ 2t & 0 & 0\end{bmatrix}, \quad \mathbf{M}_b^2 = \begin{bmatrix}n\bar{\beta}q & n\bar{\beta}q & q \\ 0 & 0 & 0 \\ n\bar{\beta}t & n\bar{\beta}t & t\end{bmatrix} \\ \mathbf{M}_c^1 &= \begin{bmatrix}r_R & 0 & 0 \\ 0 & 0 & 0 \\ 2t & 0 & r_P\end{bmatrix}, \quad \mathbf{M}_c^2 = \begin{bmatrix}n\bar{\beta}q & n\bar{\beta}q & q \\ 0 & 0 & 0 \\ n\bar{\beta}t & n\bar{\beta}t & t\end{bmatrix} \\ t &= n^2\bar{\beta}Q_P/\eta^{n-1}, \quad q = Q_R\sqrt{n}, \quad m = \alpha_F/n\end{aligned} \tag{15}$$

Following Theorem 3.2 in Powell (2011), the determinant of $z\mathbf{I} - \mathbf{M}^1$ is given by $\det(z\mathbf{I} - \mathbf{M}^1) = (z^{\bar{N}} - r_R)(z^{\bar{N}} - r_P)(z-1)z^{\bar{N}-1}$. Since $r_R, r_P \in (0, 1)$ holds, we have $\rho(\mathbf{M}^1) = 1$. Moreover, the eigenvalue 1 is simple, and its corresponding right and left eigenvectors are $\mathbf{u} = \mathbf{1}_{\bar{N}} \otimes [0\ 1\ 0]^T$ and $\mathbf{w} = [0\ 1\ 0\ \cdots\ 0]^T$, respectively. Denote the simple eigenvalue of $\mathbf{M}(\lambda)$ as a function of λ, i.e., $p(\lambda)$. Given $\mathbf{M}(\lambda) = \mathbf{M}^1 + \lambda\mathbf{M}^2$, we have $p(0) = 1$. Using Theorem 6.3.12 in Horn & Johnson (2012), one can obtain $\frac{dp(\lambda)}{d\lambda}\Big|_{\lambda=0} = \frac{\mathbf{w}^T\mathbf{M}^2\mathbf{u}}{\mathbf{w}^T\mathbf{u}} = -n^{-1}\eta^{n-1}\alpha_F < 0$. Since eigenvalues are continuous functions of the elements of a matrix, $p(\lambda)$ is strictly less than 1 when λ is sufficiently small, implying that the spectral radius of $\mathbf{M}(\lambda)$ is less than 1 when λ is sufficiently small. Noting that $\mathbf{M}(\lambda)$ has nonnegative entries and $\mathbf{M}(\lambda)^{\bar{N}+1}$ has all positive entries, from Theorems 8.5.1 and 8.5.2 in Horn & Johnson (2012), we have that each entry of $\mathbf{M}(\lambda)^k$ will converge to 0 at the rate of $\mathcal{O}(\rho(\mathbf{M}(\lambda))^k)$. Therefore, when λ is sufficiently small, $\|\mathbf{x}^k - \mathbf{1}_n \otimes \mathbf{x}^*\|$ converges to 0 at the rate of $\mathcal{O}(\rho(\mathbf{M}(\lambda))^k)$, implying that our algorithm has R-linear convergence. ■

Remark 2 *Although adding randomness in optimization parameters in the first K iterations may delay convergence (as shown in simulations in Fig. 4), it has no influence on the R-linear convergence rate. The added randomness in the first K iterations is key to enable privacy protection, as elaborated in the following subsection.*

4.3 Privacy Analysis

To protect $f_i(\cdot)$, it suffices to protect the gradient function $\nabla f_i(\cdot)$. In decentralized optimization, every node receives messages from its neighbors, and hence could exploit them to infer other nodes' private gradients. We denote all information accessible to node j as $\mathcal{I}_j$, which contains the coupling weights, stepsizes, and states associated with node j, sent information from node j to its out-neighbors, and received information from node j's in-neighbors to itself. A mathematical representation of $\mathcal{I}_j$ is given as follows:

$$\begin{aligned}\mathcal{I}_j =&\big\{\mathcal{I}_j^{\text{state}}(k)\cup\mathcal{I}_j^{\text{send}}(k)\cup\mathcal{I}_j^{\text{receive}}(k)\,\big|\,k=0,1,\dots\big\}\\ &\cup\big\{\mathbf{\Lambda}_j^k,\mathbf{R}_{jj}^k,\mathbf{A}_{jj}^k,\mathbf{C}_{jj}^k,\mathbf{B}_{jj}^k\big|\,k<K\big\}\\ &\cup\big\{\mathbf{R}_{jl}^k,\mathbf{A}_{jl}^k\big|\,k<K,\forall\, l\in\mathcal{N}_j^{in}\big\}\\ &\cup\big\{\mathbf{C}_{mj}^k,\mathbf{B}_{mj}^k\big|\,k<K,\forall\, m\in\mathcal{N}_j^{out}\big\}\\ &\cup\big\{\mathbf{\Lambda}_l^k,\mathbf{R}_{lm}^k,\mathbf{A}_{lm}^k,\mathbf{C}_{lm}^k,\mathbf{B}_{lm}^k\big|\,k\geq K,\forall\, l,m\in\mathcal{V}\big\}\end{aligned}\tag{16}$$

where

$$\begin{aligned}\mathcal{I}_j^{\text{state}}(k)&=\big\{\mathbf{x}_j^k,\mathbf{y}_j^k,\mathbf{\Lambda}_j^k\mathbf{y}_j^k,\mathbf{C}_{jj}^k\mathbf{y}_j^k+\mathbf{B}_{jj}^k(\nabla f_j^{k+1}-\nabla f_j^k)\big\}\\ \mathcal{I}_j^{\text{send}}(k)&=\big\{\mathbf{x}_j^k,\mathbf{\Lambda}_j^k\mathbf{y}_j^k,\\ &\qquad\mathbf{C}_{mj}^k\mathbf{y}_j^k+\mathbf{B}_{mj}^k(\nabla f_j^{k+1}-\nabla f_j^k)\big|\,\forall\, m\in\mathcal{N}_j^{out}\big\}\\ \mathcal{I}_j^{\text{receive}}(k)&=\big\{\mathbf{x}_l^k,\mathbf{\Lambda}_l^k\mathbf{y}_l^k,\\ &\qquad\mathbf{C}_{jl}^k\mathbf{y}_l^k+\mathbf{B}_{jl}^k(\nabla f_l^{k+1}-\nabla f_l^k)\big|\,\forall\, l\in\mathcal{N}_j^{in}\big\}\end{aligned}\tag{17}$$

represent the respective state information, sent information, and received information of node j at iteration k.

Using the defined information set, we are in position to introduce the attacker model and our definition of privacy.

Definition 3 *A node j is called honest-but-curious if it follows all protocol steps correctly but is curious and tries to infer the gradient functions of other participating nodes using the information $\mathcal{I}_j$ accessible to itself.*

Definition 4 *For a network of n nodes, the privacy of node i is preserved if honest-but-curious adversaries cannot distinguish the actual gradient $\nabla f_i(\cdot)$ of node i from an arbitrarily large variation of the gradient $\nabla\tilde{f}_i(\cdot)=\nabla f_i(\cdot)+\boldsymbol{\delta}$ where $\boldsymbol{\delta}$ can be any vector in $\mathbb{R}^d$. That is to say, there exist feasible coupling weights and stepsizes satisfying the requirements in Table II that make the information accessible to honest-but-curious adversaries exactly unchanged under arbitrary variations of node i's gradient.*

Definition 4 means that an adversary cannot even identify a range for a private function's values and thus is more stringent than the unobservability based privacy in Pequito et al. (2014) and Alaeddini et al. (2017) which defines privacy as the inability of an adversary to uniquely determine a protected value. It is also more stringent than the opacity based privacy (Saboori & Hadjicostis 2013, Lefebvre & Hadjicostis 2020, Ramasubramanian et al. 2019), which considers adversaries having access to snapshots of the output and the set of controls. Furthermore, in contrast to unobservability and opacity that protect some state values, we protect gradient function values over the entire domain (both function types and function parameters), which is more challenging.

We first show that using deterministic parameters may cause privacy breaches. Taking the $\mathcal{AB}$ algorithm in Xin & Khan (2018) as an example, node i updates its states $\mathbf{x}_i^k$ and $\mathbf{y}_i^k$ as follows:

$$\begin{aligned}\mathbf{x}_i^{k+1}&=\textstyle\sum_{j\in\mathcal{N}_i^{in}\cup\{i\}}r_{ij}\mathbf{x}_j^k-\lambda\mathbf{y}_i^k\\ \mathbf{y}_i^{k+1}&=\textstyle\sum_{j\in\mathcal{N}_i^{in}\cup\{i\}}c_{ij}\big(\mathbf{y}_j^k+\nabla f_j^{k+1}-\nabla f_j^k\big)\end{aligned}\tag{18}$$

At $k=0$, node i sends $c_{ji}\big(\mathbf{y}_i^0+\nabla f_i^1-\nabla f_i^0\big)=c_{ji}\nabla f_i^1$ to its out-neighbor j where $\mathbf{y}_i^0=\nabla f_i^0$. At $k=1$, node i sends $\mathbf{x}_i^1$ to its out-neighbor j. Note that in Xin & Khan (2018), node i sets c_{ji} as $c_{ji}=1/(|\mathcal{N}_i^{out}|+1)$ where $|\mathcal{N}_i^{out}|$ represents the number of node i's out-neighbors. As a result, using $c_{ji}\nabla f_i^1$ obtained at $k=0$ and state $\mathbf{x}_i^1$ received at $k=1$, node j can uniquely determine the gradient of node i at $\mathbf{x}_i^1$ as long as it knows the number of node i's out-neighbors. Therefore, deterministic parameters make the gradients of participating nodes easily inferable by their respective neighboring nodes.

Next, we show that by adding randomness in interaction parameters, our algorithm can protect the privacy of gradients.

Theorem 2 *In Algorithm 1, the privacy of node i can be preserved if honest-but-curious nodes do not share information with each other and $|\mathcal{N}_i^{out}\cup\mathcal{N}_i^{in}|\geq 2$ holds.*

Proof: According to Definition 4, we have to prove that when $\nabla f_i(\cdot)$ is altered to $\nabla\tilde{f}_i(\cdot)$ (could have an *arbitrarily* large difference from $\nabla f_i(\cdot)$), the information accessible to any honest-but-curious node j, i.e., $\tilde{\mathcal{I}}_j$, could be exactly the same as $\mathcal{I}_j$ in (16). Therefore, we only need to prove that there exists such $\nabla\tilde{f}_i(\cdot)$ that makes $\tilde{\mathcal{I}}_j=\mathcal{I}_j$ hold. Given $|\mathcal{N}_i^{out}\cup\mathcal{N}_i^{in}|\geq 2$, there must exist a node $m\in\mathcal{N}_i^{out}\cup\mathcal{N}_i^{in}$ such that $m\neq j$ holds. So we only need to show that there exist feasible parameters (coupling weights and stepsizes satisfying the requirements in Table II) making $\tilde{\mathcal{I}}_j=\mathcal{I}_j$ hold under $\nabla\tilde{f}_i(\cdot)=\nabla f_i(\cdot)+\boldsymbol{\delta}$ and $\nabla\tilde{f}_m(\cdot)=\nabla f_m(\cdot)-\boldsymbol{\delta}$ for *any* $\boldsymbol{\delta}=[\delta_1,\cdots,\delta_d]^T\in\mathbb{R}^d$.

We consider $m\in\mathcal{N}_i^{out}$ and $m\in\mathcal{N}_i^{in}$, separately (note that if $m\in\mathcal{N}_i^{out}\cap\mathcal{N}_i^{in}$ holds, either of the considered cases can be used in the argument to draw a same conclusion):

Case I: $m\in\mathcal{N}_i^{out}$. One can prove $\tilde{\mathcal{I}}_j=\mathcal{I}_j$ for any $\boldsymbol{\delta}\in\mathbb{R}^d$ under $\nabla\tilde{f}_i(\cdot)=\nabla f_i(\cdot)+\boldsymbol{\delta}$, $\nabla\tilde{f}_m(\cdot)=\nabla f_m(\cdot)-\boldsymbol{\delta}$, and

$$\begin{cases}\tilde{\mathbf{\Lambda}}_p^0=\mathbf{\Lambda}_p^0 \quad \forall p\in\mathcal{V}\setminus\{i,\,m\}\\ \tilde{\lambda}_i^0(l)=\lambda_i^0(l)\mathbf{y}_i^0[l]/(\mathbf{y}_i^0[l]+\delta_l)\\ \tilde{\lambda}_m^0(l)=\lambda_m^0(l)\mathbf{y}_m^0[l]/(\mathbf{y}_m^0[l]-\delta_l)\\ \tilde{\mathbf{R}}_{pq}^0=\mathbf{R}_{pq}^0,\tilde{\mathbf{A}}_{pq}^0=\mathbf{A}_{pq}^0 \quad \forall p,q\in\mathcal{V}\\ \tilde{\mathbf{C}}_{pq}^0=\mathbf{C}_{pq}^0 \quad \forall p,q\in\mathcal{V}\setminus\{i,\,m\}\\ \tilde{c}_{pi}^0(l)=c_{pi}^0(l)\mathbf{y}_i^0[l]/(\mathbf{y}_i^0[l]+\delta_l) \quad \forall p\in\mathcal{V}\setminus\{m\}\\ \tilde{c}_{mi}^0(l)=(c_{mi}^0(l)\mathbf{y}_i^0[l]+\delta_l)/(\mathbf{y}_i^0[l]+\delta_l)\\ \tilde{c}_{pm}^0(l)=c_{pm}^0(l)\mathbf{y}_m^0[l]/(\mathbf{y}_m^0[l]-\delta_l) \quad \forall p\in\mathcal{V}\setminus\{m\}\\ \tilde{c}_{mm}^0(l)=(c_{mm}^0(l)\mathbf{y}_m^0[l]-\delta_l)/(\mathbf{y}_m^0[l]-\delta_l)\\ \tilde{\mathbf{C}}_{pq}^0=\mathbf{C}_{pq}^0,\tilde{\mathbf{B}}_{pq}^0=\mathbf{B}_{pq}^0 \quad \forall p,q\in\mathcal{V}\\ \tilde{\mathbf{\Lambda}}_p^k=\mathbf{\Lambda}_p^k \quad \forall p\in\mathcal{V},\,k=1,2,\ldots\\ \tilde{\mathbf{R}}_{pq}^k=\mathbf{R}_{pq}^k,\tilde{\mathbf{A}}_{pq}^k=\mathbf{A}_{pq}^k \quad \forall p,q\in\mathcal{V},\,k=1,2,\ldots\\ \tilde{\mathbf{C}}_{pq}^k=\mathbf{C}_{pq}^k,\tilde{\mathbf{B}}_{pq}^k=\mathbf{B}_{pq}^k \quad \forall p,q\in\mathcal{V},\,k=1,2,\ldots\end{cases} \tag{19}$$

where $l=1,\cdots,d$ and "\" represents set subtraction.

Case II: $m\in\mathcal{N}_i^{in}$. One can obtain $\tilde{\mathcal{I}}_j=\mathcal{I}_j$ for any $\boldsymbol{\delta}\in\mathbb{R}^d$ under $\nabla\tilde{f}_i(\cdot)=\nabla f_i(\cdot)+\boldsymbol{\delta}$, $\nabla\tilde{f}_m(\cdot)=\nabla f_m(\cdot)-\boldsymbol{\delta}$, and

$$\begin{cases}\tilde{\mathbf{\Lambda}}_p^0=\mathbf{\Lambda}_p^0 \quad \forall p\in\mathcal{V}\setminus\{i,\,m\}\\ \tilde{\lambda}_i^0(l)=\lambda_i^0(l)\mathbf{y}_i^0[l]/(\mathbf{y}_i^0[l]+\delta_l)\\ \tilde{\lambda}_m^0(l)=\lambda_m^0(l)\mathbf{y}_m^0[l]/(\mathbf{y}_m^0[l]-\delta_l)\\ \tilde{\mathbf{R}}_{pq}^0=\mathbf{R}_{pq}^0,\tilde{\mathbf{A}}_{pq}^0=\mathbf{A}_{pq}^0 \quad \forall p,q\in\mathcal{V}\\ \tilde{\mathbf{C}}_{pq}^0=\mathbf{C}_{pq}^0 \quad \forall p,q\in\mathcal{V}\setminus\{i,\,m\}\\ \tilde{c}_{pi}^0(l)=c_{pi}^0(l)\mathbf{y}_i^0[l]/(\mathbf{y}_i^0[l]+\delta_l) \quad \forall p\in\mathcal{V}\setminus\{m\}\\ \tilde{c}_{ii}^0(l)=(c_{ii}^0(l)\mathbf{y}_i^0[l]+\delta_l)/(\mathbf{y}_i^0[l]+\delta_l)\\ \tilde{c}_{pm}^0(l)=c_{pm}^0(l)\mathbf{y}_m^0[l]/(\mathbf{y}_m^0[l]-\delta_l) \quad \forall p\in\mathcal{V}\setminus\{m\}\\ \tilde{c}_{im}^0(l)=(c_{im}^0(l)\mathbf{y}_m^0[l]-\delta_l)/(\mathbf{y}_m^0[l]-\delta_l)\\ \tilde{\mathbf{C}}_{pq}^0=\mathbf{C}_{pq}^0,\tilde{\mathbf{B}}_{pq}^0=\mathbf{B}_{pq}^0 \quad \forall p,q\in\mathcal{V}\\ \tilde{\mathbf{\Lambda}}_p^k=\mathbf{\Lambda}_p^k \quad \forall p\in\mathcal{V},\,k=1,2,\ldots\\ \tilde{\mathbf{R}}_{pq}^k=\mathbf{R}_{pq}^k,\tilde{\mathbf{A}}_{pq}^k=\mathbf{A}_{pq}^k \quad \forall p,q\in\mathcal{V},\,k=1,2,\ldots\\ \tilde{\mathbf{C}}_{pq}^k=\mathbf{C}_{pq}^k,\tilde{\mathbf{B}}_{pq}^k=\mathbf{B}_{pq}^k \quad \forall p,q\in\mathcal{V},\,k=1,2,\ldots\end{cases} \tag{20}$$

where $l=1,\cdots,d$.

To see why the parameter setting in (19) can ensure $\tilde{\mathcal{I}}_j=\mathcal{I}_j$ under $m\in\mathcal{N}_i^{out}$, we consider $k=0$ and $k\geq 1$, separately. Under the feasible parameters for $k=0$, the information accessible to node j at $k=0$ keeps unchanged and the states of each node $p\in\mathcal{V}$ satisfy $\tilde{\mathbf{x}}_p^1=\mathbf{x}_p^1$ and $\tilde{\mathbf{y}}_p^1=\mathbf{y}_p^1$. Then by setting feasible parameters for $k\geq 1$ the same as the original ones without gradient variations, it is apparent that the information accessible to node j keeps unchanged for $k\geq 1$. Following a similar argument, one can verify that the parameter setting in (20) ensures $\tilde{\mathcal{I}}_j=\mathcal{I}_j$ under $m\in\mathcal{N}_i^{in}$.

In summary, we have $\tilde{\mathcal{I}}_j=\mathcal{I}_j$ for $\nabla\tilde{f}_i(\cdot)=\nabla f_i(\cdot)+\boldsymbol{\delta}$ under *any* $\boldsymbol{\delta}\in\mathbb{R}^d$. Therefore, our algorithm can protect the privacy of node i if honest-but-curious nodes do not share information with each other and $|\mathcal{N}_i^{out}\cup\mathcal{N}_i^{in}|\geq 2$ holds. ■

Remark 3 *If we view the feasible parameters as solutions to guaranteeing $\tilde{\mathcal{I}}_j=\mathcal{I}_j$, then there exist infinitely many solutions. (19) and (20) just provide one such solution.*

Next we show that if the condition in Theorem 2 is not met, then the privacy of node i can be breached.

Theorem 3 *In Algorithm 1, the privacy of node i cannot be preserved against node j if node j is the only in-neighbor and out-neighbor of node i, i.e., $\mathcal{N}_i^{out}=\mathcal{N}_i^{in}=\{j\}$.*

Proof: When $\mathcal{N}_i^{out}=\mathcal{N}_i^{in}=\{j\}$ holds, one can get the dynamics of $\mathbf{y}_i^k$ from (3) as follows:

$$\begin{aligned}\mathbf{y}_i^{k+1}=\mathbf{C}_{ii}^k\mathbf{y}_i^k+\mathbf{B}_{ii}^k\big(\nabla f_i^{k+1}-\nabla f_i^k\big)+\mathbf{C}_{ij}^k\mathbf{y}_j^k\\ +\mathbf{B}_{ij}^k\big(\nabla f_j^{k+1}-\nabla f_j^k\big)\end{aligned} \tag{21}$$

According to the parameter design in Table II, we have $\mathbf{y}_i^k=\mathbf{C}_{ii}^k\mathbf{y}_i^k+\mathbf{C}_{ji}^k\mathbf{y}_i^k$ and $\nabla f_i^{k+1}-\nabla f_i^k=(\mathbf{B}_{ii}^k+\mathbf{B}_{ji}^k)(\nabla f_i^{k+1}-\nabla f_i^k)$ for $k\in\mathbb{Z}_{\geq 0}$ based on the facts $\mathbf{C}_{ii}^k+\mathbf{C}_{ji}^k=\mathbf{I}_d$ and $\mathbf{B}_{ii}^k+\mathbf{B}_{ji}^k=\mathbf{I}_d$. So we can rewrite (21) as

$$\begin{aligned}\mathbf{y}_i^{k+1}=&\mathbf{y}_i^k-\mathbf{C}_{ji}^k\mathbf{y}_i^k+(\nabla f_i^{k+1}-\nabla f_i^k)+\mathbf{C}_{ij}^k\mathbf{y}_j^k\\ &-\mathbf{B}_{ji}^k(\nabla f_i^{k+1}-\nabla f_i^k)+\mathbf{B}_{ij}^k(\nabla f_j^{k+1}-\nabla f_j^k)\end{aligned} \tag{22}$$

Denote $\mathbf{m}_j^k$ as

$$\begin{aligned}\mathbf{m}_j^k=&-\mathbf{C}_{ji}^k\mathbf{y}_i^k+\mathbf{C}_{ij}^k\mathbf{y}_j^k-\mathbf{B}_{ji}^k(\nabla f_i^{k+1}-\nabla f_i^k)\\ &+\mathbf{B}_{ij}^k(\nabla f_j^{k+1}-\nabla f_j^k)\end{aligned} \tag{23}$$

Note that $\mathbf{m}_j^k$ is accessible to the honest-but-curious node j because $\mathbf{C}_{ji}^k\mathbf{y}_i^k+\mathbf{B}_{ji}^k(\nabla f_i^{k+1}-\nabla f_i^k)$ is the information node i sends to node j, and $\mathbf{C}_{ij}^k\mathbf{y}_j^k+\mathbf{B}_{ij}^k(\nabla f_j^{k+1}-\nabla f_j^k)$ is the information computed by node j. Plugging (23) into (22), node j can obtain $\mathbf{y}_i^{k+1}-\mathbf{y}_i^k=\nabla f_i^{k+1}-\nabla f_i^k+\mathbf{m}_j^k$ and further (note $\mathbf{y}_i^0=\nabla f_i^0$)

$$\mathbf{y}_i^{k+1}=\nabla f_i^{k+1}+\textstyle\sum_{l=0}^k\mathbf{m}_j^l \tag{24}$$

Since $\lim_{k\to\infty}\mathbf{y}_i^{k+1}=\mathbf{0}_d$ holds as k goes to infinity, we have $\lim_{k\to\infty}\nabla f_i^{k+1}=\lim_{k\to\infty}\nabla f_i(\mathbf{x}_i^{k+1})=-\lim_{k\to\infty}\sum_{l=0}^k\mathbf{m}_j^l$. Given $\lim_{k\to\infty}\mathbf{x}_i^{k+1}=\lim_{k\to\infty}\mathbf{x}_j^{k+1}=\mathbf{x}^*$, node j also knows $\lim_{k\to\infty}\mathbf{x}_i^{k+1}$, meaning that an honest-but-curious node j can infer the gradient of node i at the global optimal solution $\mathbf{x}^*$ using (24). Therefore, the privacy of node i cannot be preserved against node j when node j is the only in-neighbor and out-neighbor of node i. ■

Remark 4 *In Theorems 2 and 3, we consider non-colluding case where a single adversary acts on its own. Similar results can be obtained when multiple adversaries collude with each other. Please refer to Theorems 4 and 5 in Gao et al. (2022) for details.*

Remark 5 *Even using time-varying parameters, the $\mathcal{AB}$ algorithm cannot guarantee the privacy in Definition 4. This is because $\mathcal{AB}$ does not allow negative coupling weights, which is key to ensure feasible solutions for the parameters required in (19) and (20), and hence make adversaries' accessible information unchanged under arbitrary variations.*

Remark 6 *From Theorem 1, one can see that the parameter K does not affect optimization accuracy. Furthermore, from (19) and (20), we can see that only changing the coupling weights/stepsizes in the initial iteration $k = 0$ is enough to cover gradient variations. In other words, any $K \geq 1$ is sufficient to protect the defined privacy for gradients. It is also worth noting that although the randomness added in the first K iterations does not affect the convergence rate (as proven in Theorem 1), it does delay the convergence since the algorithm only starts to converge after iteration K (see Fig. 4). Hence, to minimize delay in the convergence process, we can set $K = 1$.*

5 Numerical Simulations

5.1 Privacy Protection in the Rendezvous Problem

We consider the distributed rendezvous problem where a group of nodes want to agree on the nearest meeting point without revealing each other's initial position (Huang et al. 2015). Mathematically this can be modeled as the problem $\min_{x\in\mathbb{R}^d} \ F(\mathbf{x}) = \sum_{i=1}^n f_i(\mathbf{x}) = \sum_{i=1}^n \frac{1}{2}\|\mathbf{x}-\mathbf{p}_i\|^2$, where $\mathbf{p}_i$ represents the initial position of node i. For the simplicity of exposition, we consider the $d = 1$ case but similar results can be obtained when $d \neq 1$. We consider three nodes connected in a directed cycle as shown in Fig. 1 (a). Let node 3 be an honest-but-curious node which collects received data to learn the gradient function of node 1. Node 2 does not collude with node 3. K was set to 3. In the simulation, we first ran our algorithm and recorded $\mathcal{I}_3$, the information accessible to node 3 (cf. (16)). Then we show that information accessible to node 3 can be exactly the same under a completely different gradient function $\nabla\tilde{f}_1(\mathbf{x}_1)$.

Fig. 2 shows $\mathbf{x}_1^k$, $\mathbf{\Lambda}_1^k\mathbf{y}_1^k$, and $\mathbf{C}_{31}^k\mathbf{y}_1^k + \mathbf{B}_{31}^k(\nabla f_1^{k+1} - \nabla f_1^k)$ in $\mathcal{I}_3$ and $\tilde{\mathbf{x}}_1^k$, $\tilde{\mathbf{\Lambda}}_1^k\tilde{\mathbf{y}}_1^k$, and $\tilde{\mathbf{C}}_{31}^k\tilde{\mathbf{y}}_1^k + \tilde{\mathbf{B}}_{31}^k(\nabla \tilde{f}_1^{k+1} - \nabla \tilde{f}_1^k)$ in $\tilde{\mathcal{I}}_3$, respectively. The trajectories of the observations of node 3 in both cases are identical, although the gradients $\nabla f_1(\mathbf{x}_1)$ and $\nabla\tilde{f}_1(\mathbf{x}_1)$ are clearly different (cf. Fig. 3). Since node 3 receives the same information under $\nabla\tilde{f}_1(\mathbf{x}_1) \neq \nabla f_1(\mathbf{x}_1)$, it has no way to infer the real gradient function of node 1.

5.2 Distributed Estimation Problem

In this subsection we focus on the convergence performance of our algorithm and compare our algorithm with other decentralized optimization algorithms. We consider the canonical distributed estimation problem $\min_{\mathbf{x}\in\mathbb{R}^d} \ F(\mathbf{x}) = \sum_{i=1}^n \left(\|\mathbf{z}_i - \mathbf{Q}_i\mathbf{x}\|^2 + \sigma_i\|\mathbf{x}\|^2\right)$ in

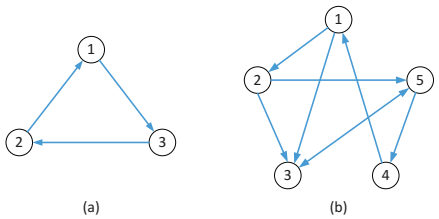

Fig. 1. Graphs used in simulations: (a) a directed cycle graph with 3 nodes; (b) a strongly connected graph with 5 nodes.

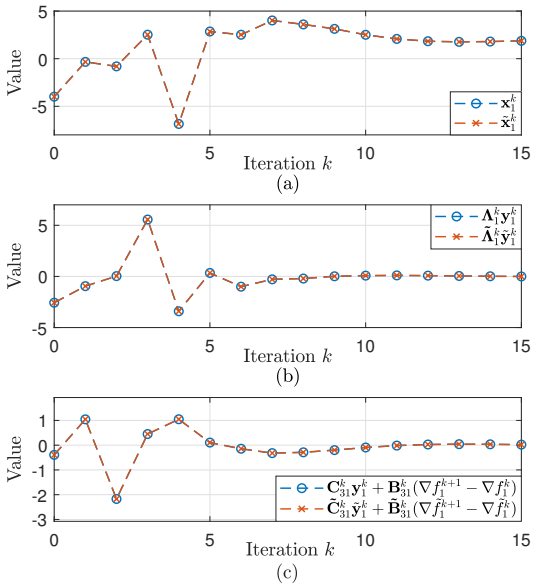

Fig. 2. The information accessible to node 3 are the same under two different gradient functions of node 1 depicted in Fig. 3.

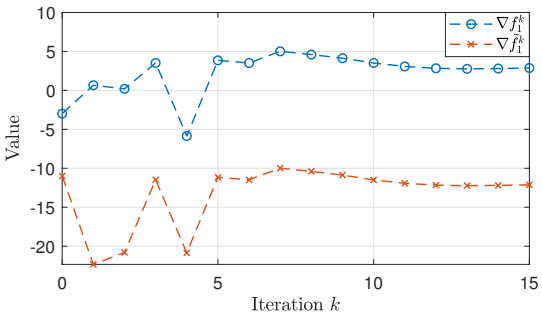

Fig. 3. The two different gradient functions of node 1 that lead to identical observations at node 3.

Xu et al. (2017), where n nodes cooperatively measure a certain unknown parameter $\mathbf{x} \in \mathbb{R}^d$. In this problem, each node i has access to its local cost function $f_i(\mathbf{x}_i) = \|\mathbf{z}_i - \mathbf{Q}_i\mathbf{x}_i\|^2 + \sigma_i\|\mathbf{x}_i\|^2$ with $\mathbf{Q}_i \in \mathbb{R}^{s\times d}$ being its measurement matrix and $\mathbf{z}_i \in \mathbb{R}^s$ being its measurement data. The regularization parameter σ_i can be set to 0 (resp. a positive value) to make f_i general convex (resp. strongly convex) under appropriate $\mathbf{Q}_i$. We considered a network of $n = 5$ nodes interacting on a strongly connected graph in Fig. 1 (b). d and s were set to 2 and 3, respectively. Parameter K was set to 3.

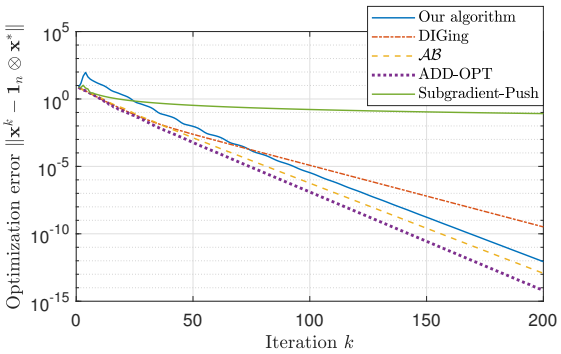

Fig. 4. Comparison of convergence rates of different optimization algorithms.

5.2.1 *Comparison with Other Algorithms*

We compared our algorithm with DIGing (Nedić, Olshevsky, Shi & Uribe 2017), $\mathcal{AB}$ (Xin & Khan 2018), ADD-OPT (Xi et al. 2018), and Subgradient-Push (Nedić & Ozdaglar 2015) to evaluate the influence of privacy design on optimization performance. The stepsize was set to 0.06 for all of the considered algorithms except Subgradient-Push whose stepsize was set to a diminishing sequence $\lambda^k = 1/k$. The simulation results on optimization error $\|\mathbf{x}^k - \mathbf{1}_n \otimes \mathbf{x}^*\|$ are depicted in Fig. 4, which corroborates our statement in Remark 2 that the added randomness has no influence on the R-linear convergence. Of course, in our algorithm, the privacy-induced randomness delayed convergence to iteration step $k \geq 4$.

5.2.2 *Comparison with Huang et al. (2015)*

We also compared our algorithm with the differential privacy based approach in Huang et al. (2015). We ran the algorithm in Huang et al. (2015) under four different privacy levels, i.e., $\epsilon = 0.1, 1, 10, 100$, respectively. The domain of optimization in Huang et al. (2015) was set to $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^2 | \|\mathbf{x}\| \leq 10\}$. Note that the optimal solution $\mathbf{x}^* = [0.6881,\ 0.5103]^T$ resides in $\mathcal{X}$. For each privacy level ϵ, we repeated the simulation for $1,000$ times, and averaged the optimization error trajectories. Under a stepsize 0.06, we also measured the mean optimization error of our algorithm over $1,000$ repetitions. The results in Fig. 5 confirm the trade-off between privacy and accuracy for differential-privacy based approaches and demonstrate the advantage of our algorithm in ensuring optimization accuracy.

5.2.3 *Comparison with Lou et al. (2018)*

Then we compared our algorithm with the privacy approach in Lou et al. (2018). The closed convex projection set $\mathcal{X}$ was set to $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^2 | \|\mathbf{x}\| \leq 10\}$ for the algorithm in Lou et al. (2018). The other parameters for the algorithm in Lou et al. (2018) were set as follows: each node i randomly chose c_i from $(0,\ 5)$, and updated its state with its subgradient once every T_i iterations where T_i was randomly chosen from $\{1, 2, 3, 4, 5\}$. The stepsize of our algorithm was set to 0.06. The results are shown in Fig. 6 (a). Clearly our algorithm

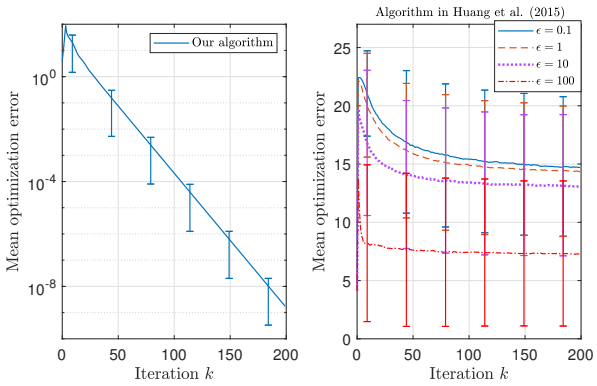

Fig. 5. Comparison of mean optimization error over $1,000$ repetitions between our algorithm and the differential-privacy based approach in Huang et al. (2015).

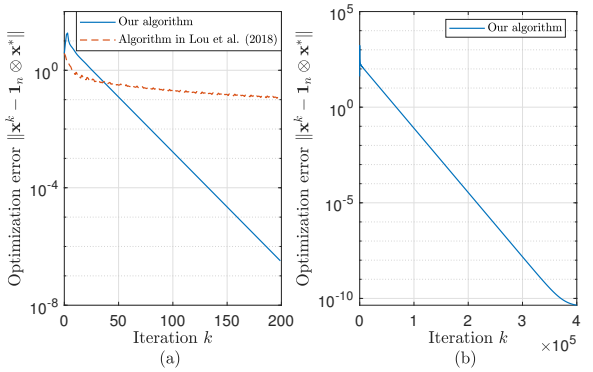

Fig. 6. (a) Comparison of optimization error $\|\mathbf{x}^k - \mathbf{1}_n \otimes \mathbf{x}^*\|$ between our algorithm and the algorithm in Lou et al. (2018); (b) The optimization error trajectory of our algorithm in a network of $n = 100$ nodes.

has a R-linear convergence whereas the convergence rate of Lou et al. (2018) is much slower.

5.2.4 *Numerical Simulations on Large-scale Networks*

Finally, we verified the scalability of our algorithm using a network of $n = 100$ nodes. Each agent i was assumed to have two out-neighbors, i.e., $\mathcal{N}_i^{out} = \{\bar{i}+1,\ \overline{i+1}+1\}$, where the superscript " $\bar{}$ " represents modulo operation on n, i.e., $\bar{i} \triangleq i \mod n$. The evolution of optimization error $\|\mathbf{x}^k - \mathbf{1}_n \otimes \mathbf{x}^*\|$ is shown in Fig. 6 (b). It can be seen that the convergence rate is still linear, meaning that our proposed algorithm can guarantee the convergence of all nodes to the global optimal solution even when the network size is large.

6 Conclusions

In this paper we proposed a dynamics based privacy approach for decentralized optimization. Our approach can enable privacy without compromising optimization accuracy or incurring heavy computation/communication overhead. This is in distinct difference from differential-privacy based approaches which compromise optimization accuracy and

encryption based approaches which incur heavy computation/communication overhead. We rigorously characterized the convergence properties of our algorithm and its privacy-preserving performance. In addition, to facilitate the privacy design, we also proposed a general framework of gradient-tracking based decentralization optimization, which includes many commonly used algorithms as special cases. Finally, we provided numerical simulation results to confirm the effectiveness and efficiency of our proposed algorithm.

References

Alaeddini, A., Morgansen, K. & Mesbah, M. (2017), Adaptive communication networks with privacy guarantees, *in* 'Proc. 2017 American Contr. Conf.', pp. 4460–4465.

Bazerque, J. A. & Giannakis, G. B. (2010), 'Distributed spectrum sensing for cognitive radio networks by exploiting sparsity', *IEEE Trans. Signal Process.* **58**(3), 1847–1862.

Charalambous, T., Manitara, N. E. & Hadjicostis, C. N. (2019), Privacy-preserving average consensus over digraphs in the presence of time delays, *in* 'Proc. 57th Annu. Allerton Conf. Commun. Control Comput.', pp. 238–245.

Du, W., Yao, L., Wu, D., Li, X., Liu, G. & Yang, T. (2018), Accelerated distributed energy management for microgrids, *in* 'Proc. IEEE Power Energy Soc. Gen. Meeting', pp. 1–5.

Fazlyab, M., Ribeiro, A., Morari, M. & Preciado, V. M. (2018), 'Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems', *SIAM J. Optim.* **28**(3), 2654–2689.

Gade, S. & Vaidya, N. H. (2018), Private optimization on networks, *in* 'Proc. 2018 American Contr. Conf.', pp. 1402–1409.

Gan, L., Topcu, U. & Low, S. H. (2013), 'Optimal decentralized protocol for electric vehicle charging', *IEEE Trans. Power Syst.* **28**(2), 940–951.

Gao, H., Wang, Y. & Nedić, A. (2022), 'Dynamics based privacy preservation in decentralized optimization', *arXiv:2207.05350* .

Gao, H., Zhang, C., Ahmad, M. & Wang, Y. (2018), Privacy-preserving average consensus on directed graphs using push-sum, *in* 'Proc. IEEE Conf. Commun. Netw. Security'.

Gupta, N., Katz, J. & Chopra, N. (2017), 'Privacy in distributed average consensus', *IFAC-PapersOnLine* **50**(1), 9515–9520.

Hadjicostis, C. N. & Dominguez-Garcia, A. D. (2020), 'Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus', *IEEE Trans. Autom. Control* **65**(9), 3887–3894.

Hale, M. T. & Egerstedt, M. (2017), 'Cloud-enabled differentially private multiagent optimization with constraints', *IEEE Trans. Control Netw. Syst.* **5**(4), 1693–1706.

He, J., Cai, L., Cheng, P., Pan, J. & Shi, L. (2018), 'Distributed privacy-preserving data aggregation against dishonest nodes in network systems', *IEEE Internet Things J.* **6**(2), 1462–1470.

He, L., Karimireddy, S. P. & Jaggi, M. (2020), 'Secure byzantine-robust machine learning', *arXiv:2006.04747* .

Horn, R. A. & Johnson, C. R. (2012), *Matrix analysis*, Cambridge university press.

Huang, Z., Mitra, S. & Vaidya, N. (2015), Differentially private distributed optimization, *in* 'Proc. Int. Conf. Distrib. Comput. Netw.', pp. 4:1–4:10.

Kia, S. S., Cortés, J. & Martinez, S. (2015), 'Dynamic average consensus under limited control authority and privacy requirements', *Int. J. Robust Nonlinear Control* **25**(13), 1941–1966.

Lefebvre, D. & Hadjicostis, C. N. (2020), 'Privacy and safety analysis of timed stochastic discrete event systems using markovian trajectory-observers', *Discret. Event Dyn. Syst.* pp. 1–28.

Li, Q., Heusdens, R. & Christensen, M. G. (2020), 'Privacy-preserving distributed optimization via subspace perturbation: a general framework', *IEEE Trans. Signal Process.* **68**, 5983–5996.

Lou, Y., Yu, L., Wang, S. & Yi, P. (2018), 'Privacy preservation in distributed subgradient optimization algorithms', *IEEE Trans. Cybern.* **48**(7), 2154–2165.

Lu, Y. & Zhu, M. (2018), 'Privacy preserving distributed optimization using homomorphic encryption', *Automatica* **96**, 314–325.

Manitara, N. E. & Hadjicostis, C. N. (2013), Privacy-preserving asymptotic average consensus, *in* '2013 Eur. Control Conf.', pp. 760–765.

Mo, Y. & Murray, R. M. (2017), 'Privacy preserving average consensus', *IEEE Trans. Autom. Control* **62**(2), 753–765.

Nedić, A., Olshevsky, A. & Shi, W. (2017), 'Achieving geometric convergence for distributed optimization over time-varying graphs', *SIAM J. Optim.* **27**(4), 2597–2633.

Nedić, A., Olshevsky, A., Shi, W. & Uribe, C. A. (2017), Geometrically convergent distributed optimization with uncoordinated step-sizes, *in* 'Proc. 2017 American Contr. Conf.', pp. 3950–3955.

Nedić, A. & Ozdaglar, A. (2009), 'Distributed subgradient methods for multi-agent optimization', *IEEE Trans. Autom. Control* **54**(1), 48–61.

Nedić, A. & Ozdaglar, A. (2015), 'Distributed optimization over time-varying directed graphs', *IEEE Trans. Autom. Control* **60**(3), 601–615.

Nozari, E., Tallapragada, P. & Cortés, J. (2016), 'Differentially private distributed convex optimization via functional perturbation', *IEEE Trans. Control Netw. Syst.* **5**(1), 395–408.

Pequito, S., Kar, S., Sundaram, S. & Aguiar, A. P. (2014), Design of communication networks for distributed computation with privacy guarantees, *in* 'Proc. IEEE 53rd Conf. Decis. Control', pp. 1370–1376.

Pilet, A. B., Frey, D. & Taiani, F. (2019), Robust privacy-preserving gossip averaging, *in* 'Proc. 21st Int. Symp. Stabilization, Saf., Secur. Distrib. Syst.', Springer, pp. 38–52.

Powell, P. D. (2011), 'Calculating determinants of block matrices', *arXiv:1112.4379* .

Pu, S., Shi, W., Xu, J. & Nedić, A. (2018), A push-pull gradient method for distributed optimization in networks,

in 'Proc. IEEE 57th Conf. Decis. Control', pp. 3385–3390.
Qu, G. & Li, N. (2017), 'Harnessing smoothness to accelerate distributed optimization', *IEEE Trans. Control Netw. Syst.* **5**(3), 1245–1260.
Raffard, R. L., Tomlin, C. J. & Boyd, S. P. (2004), Distributed optimization for cooperative agents: Application to formation flight, *in* 'Proc. IEEE 43rd Conf. Decis. Control', Vol. 3, pp. 2453–2459.
Ramasubramanian, B., Cleaveland, W. R. & Marcus, S. (2019), 'Notions of centralized and decentralized opacity in linear systems', *IEEE Trans. Autom. Control* **65**(4), 1442–1455.
Ridgley, I. D., Freeman, R. A. & Lynch, K. M. (2019), Simple, private, and accurate distributed averaging, *in* 'Proc. 57th Annu. Allerton Conf. Commun. Control Comput.', pp. 446–452.
Ruan, M., Gao, H. & Wang, Y. (2019), 'Secure and privacy-preserving consensus', *IEEE Trans. Autom. Control* **64**(10), 4035–4049.
Saadatniaki, F., Xin, R. & Khan, U. A. (2020), 'Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices', *IEEE Trans. Autom. Control* **65**(11), 4769–4780.
Saboori, A. & Hadjicostis, C. N. (2013), 'Verification of initial-state opacity in security applications of discrete event systems', *Inform. Sciences* **246**, 115–132.
Shi, W., Ling, Q., Wu, G. & Yin, W. (2015), 'Extra: An exact first-order algorithm for decentralized consensus optimization', *SIAM J. Optim.* **25**(2), 944–966.
Wang, Y. & Nedić, A. (2022), 'Tailoring gradient methods for differentially-private distributed optimization', *arXiv:2202.01113* .
Wang, Y. & Poor, V. (2022), 'Decentralized stochastic optimization with inherent privacy protection', *IEEE Trans. Autom. Control* .
Xi, C., Xin, R. & Khan, U. A. (2018), 'Add-opt: Accelerated distributed directed optimization', *IEEE Trans. Autom. Control* **63**(5), 1329–1339.
Xin, R. & Khan, U. A. (2018), 'A linear algorithm for optimization over directed graphs with geometric convergence', *IEEE Control Syst. Lett.* **2**(3), 315–320.
Xu, J., Zhu, S., Soh, Y. C. & Xie, L. (2015), Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes, *in* 'Proc. IEEE 54th Conf. Decis. Control,', pp. 2055–2060.
Xu, J., Zhu, S., Soh, Y. C. & Xie, L. (2017), 'Convergence of asynchronous distributed gradient methods over stochastic networks', *IEEE Trans. Autom. Control* **63**(2), 434–448.
Yan, F., Sundaram, S., Vishwanathan, S. V. N. & Qi, Y. (2012), 'Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties', *IEEE Trans. Knowl. Data Eng.* **25**(11), 2483–2493.
Yuan, K., Ling, Q. & Yin, W. (2016), 'On the convergence of decentralized gradient descent', *SIAM J. Optim.* **26**(3), 1835–1854.
Zhang, C., Ahmad, M. & Wang, Y. (2018), 'Admm based privacy-preserving decentralized optimization', *IEEE Trans. Inf. Forensic Secur.* **14**(3), 565–580.
Zhang, C. & Wang, Y. (2018*a*), 'Distributed event localization via alternating direction method of multipliers', *IEEE Trans. Mob. Comput.* **17**(2), 348–361.
Zhang, C. & Wang, Y. (2018*b*), 'Enabling privacy-preservation in decentralized optimization', *IEEE Trans. Control Netw. Syst.* **6**(2), 679–689.
Zhang, S., Yi, X., George, J. & Yang, T. (2019), Computational convergence analysis of distributed optimization algorithms for directed graphs, *in* 'Proc. IEEE 15th Int. Conf. Control Autom.', pp. 1096–1101.

Huan Gao was born in Shandong, China. He received the B.S. degree in automation and the M.Sc. degree in control theory and control engineering from Northwestern Polytechnical University, Xi'an, Shaanxi, China, in 2011 and 2015, respectively, and the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 2020. He is currently an Associate Professor with the School of Automation, Northwestern Polytechnical University. His research interests include decentralized optimization, cooperative control, and privacy preservation in distributed systems.

Yongqiang Wang was born in Shandong, China. He received the B.S. degree in electrical engineering and automation, the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2004, and the M.Sc. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2009. From 2007 to 2008, he was with the University of Duisburg-Essen, Germany, as a Visiting Student. He was a Project Scientist with the University of California at Santa Barbara, Santa Barbara. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Clemson University. His current research interests include distributed control, optimization, and learning, with emphasis on privacy protection. He currently serves as an associate editor for IEEE Transactions on Control of Network systems and IEEE Transactions on Automatic Control.

Angelia Nedić holds a Ph.D. from Moscow State University, Moscow, Russia, in Computational Mathematics and Mathematical Physics (1994), and a Ph.D. from Massachusetts Institute of Technology, Cambridge, USA in Electrical and Computer Science Engineering (2002). She has worked as a senior engineer in BAE Systems North America, Advanced Information Technology Division at Burlington, MA. She is the recipient of an NSF CAREER Award 2007 in Operations Research for her work

in distributed multi-agent optimization. She is a recipient (jointly with her co-authors) of the Best Paper Award at the Winter Simulation Conference 2013 and the Best Paper Award at the International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt) 2015. Also, she is a coauthor of the book Convex Analysis and Optimization. Her current interest is in large-scale optimization, games, control and information processing in networks.