Dependable Distributed Nonconvex Optimization via Polynomial Approximation

Zhiyu He, Jianping He, Cailian Chen and Xinping Guan

Abstract—There has been work on exploiting polynomial approximation to solve distributed nonconvex optimization problems involving univariate objectives. This idea facilitates arbitrarily precise global optimization without requiring local evaluations of gradients at every iteration. Nonetheless, there remains a gap between existing theoretical guarantees and diverse practical requirements for dependability, notably privacy preservation and robustness to network imperfections (e.g., time-varying directed communication and asynchrony). To fill this gap and keep the above strengths, we propose a Dependable Chebyshev-Proxy-based distributed Optimization Algorithm (D-CPOA). Specifically, to ensure both accuracy of solutions and privacy of local objectives, a new privacy-preserving mechanism is designed. This mechanism leverages the randomness in blockwise insertions of perturbed vector states and hence provides an improved privacy guarantee compared to the literature in terms of $(\alpha, \beta)$-data-privacy. Furthermore, to gain robustness to various network imperfections, we use the push-sum consensus protocol as a backbone, discuss its specific enhancements, and evaluate the performance of the proposed algorithm accordingly. Thanks to the linear consensus-based structure of iterations, we avoid the privacy-accuracy trade-off and the bother of selecting appropriate step-sizes in different settings. We provide rigorous analysis of the accuracy, dependability and complexity. It is shown that the advantages brought by the idea of polynomial approximation are maintained when all the above requirements exist. Simulations demonstrate the effectiveness of the developed algorithm.

Index Terms—Distributed optimization, Chebyshev polynomial approximation, dependability, privacy preservation, data-privacy, robustness.

I. INTRODUCTION

Distributed optimization enables multiple agents in a network to agree on the optimal points of the average of local objective functions. This global aim is achieved by exploiting local computations and communication between neighboring agents. Such a distributed architecture is highly preferable in a variety of applications related to large-scale networked systems, e.g., distributed learning [2], energy management [3] and resource allocation [4]. In these applications, the needs of improving efficiency, scalability and robustness as well as protecting privacy have motivated the development of distributed strategies, which serve as plausible alternatives to their centralized counterparts.

A. Motivations

Considerable effort has been devoted to designing efficient gradient-based distributed optimization algorithms, e.g., [2], [5]–[7] and extending them to meet diverse practical requirements, including privacy preservation [4], [8], [9], time-varying directed communication [10]–[12] and asynchronous computations to allow lack of coordination [12], [13], delays and packet drops [14], [15]. Most of these extensions focus on convex problems, and some critical issues including privacy-accuracy trade-off [9], appropriate numbers of iterations [4] and bounds for constant step-sizes [12], [15] are explored.

Despite their wide applicability, gradient-based distributed algorithms only ensure convergence to stationary points for nonconvex problems, and their loads of evaluations of gradients or function values grow with the number of iterations. These issues motivate the study of [16], where polynomial approximations are introduced to substitute for general local objectives, and a gradient-free and consensus-based iteration rule is adopted for the exchange of vectors of coefficients of local approximations. These designs help to achieve arbitrarily precise global optimization and reduce the total costs of evaluations. More importantly, they separate the algorithm in [16] from typical gradient-based methods, and offer a new perspective to solve distributed optimization problems.

Nonetheless, there are two key issues that affect the practical values of the algorithm CPCA in [16]. First, it is not privacy-preserving for the potential leakage of sensitive local objective functions. The leakage results from its consensus-based iterations where vectors of coefficients of local approximations are directly exchanged. Once the adversaries obtain the exact initial vector of the target agent, they can recover a fairly accurate estimate of the corresponding local objective. Hence, how to effectively preserve the privacy of local objectives in this algorithm and to quantify protection results are well worth consideration. Second, it only handles the optimization over static undirected networks with perfect communication. Given that issues including time-varying and directed links, lack of coordination and packet drops are common in applications, it is meaningful to investigate their effects on the performance of this algorithm and find effective measures to gain robustness against network imperfections. The above issues lead to the study of this work. We aim to demonstrate that the novel idea of introducing polynomial approximation into distributed optimization not only allows for further enhancements to meet various practical needs, but also maintains notable advantages in performance when the factors of privacy and robustness are taken into account.

B. Contributions

In this paper, we exploit the idea of introducing polynomial approximation and develop a Dependable Chebyshev-Proxy-
based distributed Optimization Algorithm (D-CPOA), considering typical requirements of privacy preservation and robustness to various network imperfections, including time-varying and directed communication and asynchrony. The core idea is to construct Chebyshev polynomial approximations (i.e., proxies) for univariate objective functions, employ consensus-based iterations with the privacy-preserving mechanism to exchange vectors of coefficients of local proxies, and locally solve an approximate problem by optimizing the recovered global proxy.

We first focus on the need of preserving the privacy of local objective functions. This need is reduced to keeping the initial vectors as secrets due to the design of the proposed algorithm. To achieve this goal, we propose a new privacy-preserving mechanism for consensus-based iterations involving vector states. This mechanism is not confined to obfuscating local states with random noises, which has been extensively studied, e.g., in [17]–[19]. In contrast, we also introduce randomness in blockwise insertions of perturbed initial states to make their dimensions uncertain to the adversaries. Then, we provide a rigorous analysis of the effect of privacy preservation through \((\alpha, \beta)\)-data privacy [20], and characterize the relationship between estimation accuracy and disclosure probability corresponding to the sensitive local information. It is shown that a stronger privacy guarantee (i.e., a reduced disclosure probability) is obtained compared to the design where existing algorithms (e.g., [18], [21]–[23]) are directly extended to handle vector states. Moreover, we avoid the trade-off between privacy and accuracy and consider a more general problem with nonconvex objectives, which are in sharp contrast with existing differentially private distributed convex optimization algorithms [4], [8], [9].

To gain robustness against various imperfections in network communication, we employ the push-sum average consensus protocol [24] as a backbone of iterations to handle time-varying and directed graphs, and then discuss its asynchronous extensions. We analyze in detail the relationship between the accuracy of consensus and that of the obtained solutions, thus verifying that the proposed algorithm keeps effective and accurate when such imperfections are present, and there is no need to carefully select proper step-sizes in different circumstances.

C. Paper Organization

The remainder of this paper is organized as follows. Section II describes the problem of interest and gives some preliminaries. Section III presents the algorithm D-CPOA. Section IV analyzes the accuracy, dependability and complexity of the proposed algorithm. Numerical evaluations are performed in Section V, followed by the review of related work in Section VI. Finally, Section VII concludes this paper.

II. PROBLEM DESCRIPTION AND PRELIMINARIES

Consider a network system consisting of \(N\) agents, each of which owns a univariate local objective function \(f_i(x) : X_i \to \mathbb{R}\) and a local constraint set \(X_i \subset \mathbb{R}\). The network at time \(t(t \in \mathbb{N})\) is described as a directed graph \(G^t = (\mathcal{V}, \mathcal{E}^t)\), where \(\mathcal{V}\) is the set of agents, and \(\mathcal{E}^t \subseteq \mathcal{V} \times \mathcal{V}\) is the set of edges. Note that \((i, j) \in \mathcal{E}^t\) if and only if (iff) agent \(i\) can receive messages from agent \(j\) at time \(t\). In this paper, the superscript \(t\), subscripts \(i, j\) and script in parentheses \(k\) denote the number of iterations, indexes of agents and index of components in a vector, respectively.

A. Problem Description

In this paper, we aim to solve the following constrained optimization problem

\[
\begin{align*}
\min_x & \quad f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x), \\
\text{s.t.} & \quad x \in X = \bigcap_{i=1}^{N} X_i
\end{align*}
\]

in a distributed and dependable manner. Specifically, the global aim of optimization needs to be achieved by means of local communication and computations. Meanwhile, diverse practical requirements will be taken into account, including preservation of the privacy of local objective functions and robustness to time-varying directed communication and asynchrony. Some basic assumptions are given as follows.

Assumption 1. Every \(f_i(x)\) is Lipschitz continuous on \(X_i\).

Assumption 2. All \(X_i\) are closed, bounded and convex sets.

Assumption 1 holds for a wide range of objective functions, including continuously differentiable functions. Both Assumptions 1 and 2 are commonly seen among the literature (e.g., [2], [25], [26]) and are satisfied in problems of practical interests.
Assumption 3. \( \{G^t\} \) is B-strongly-connected, i.e., there exists a positive integer B, such that for any k \( \in \mathbb{N} \), the graph \( (V, \bigcup_{i=k}^{(k+1)B-1} E^t) \) is strongly connected.

Assumption 3 states that the union graph is strongly connected for a time window of length B. It is much weaker than that requiring connectivity at every time, and is sufficient for information flow from one agent in networks to another [10].

In problem (1), the objective functions are (possibly) nonconvex, and the constraint sets are convex. Therefore, it is a constrained distributed nonconvex optimization problem. Under Assumption 2, the local constraint set \( X_i \) is initialized to be \( \{a_i, b_i\} \), where \( a_i, b_i \in \mathbb{R} \). As a result, the intersection set \( X \) becomes \([a, b]\), where \( a = \max_{i \in V} a_i \), \( b = \min_{i \in V} b_i \).

### B. Preliminaries

**Consensus Protocols**

Let \( N_i^{\text{in},t} \) and \( N_i^{\text{out},t} \) be the sets of agent \( i \)’s in-neighbors and out-neighbors, respectively, and \( \delta_i^{\text{out},t} = |N_i^{\text{out},t}| \) be its out-degree, i.e., the cardinality of \( N_i^{\text{out},t} \). Suppose that every agent \( i \) owns a local variable \( x_i^t \in \mathbb{R} \). There are two classical consensus protocols, i.e., maximum consensus and average consensus, that allow agents to reach global agreement through local information exchange only. The maximum consensus protocol [28] is

\[
x_i^{t+1} = \max_{j \in N_i^{\text{in},t}} x_j^t.
\]  

(2)

It can be proven that with (2), all \( x_i^t \) converge to \( \max_{i \in V} x_i^0 \) in \( T((N - 1)B) \) iterations, i.e.,

\[
x_i^t = \max_{i \in V} x_i^0, \quad \forall t \geq T, \quad i \in V.
\]

The push-sum average consensus protocol [24] is

\[
x_i^{t+1} = \sum_{j \in N_i^{\text{in},t}} a_{ij} x_j^t, \quad y_i^{t+1} = \sum_{j \in N_i^{\text{in},t}} a_{ij} y_j^t,
\]  

(3)

where \( y_i^t \in \mathbb{R} \) is initialized to be 1 for all \( i \in V \). The key to the convergence of (3) lies in constructing a column stochastic weight matrix \( A^t \stackrel{\Delta}{=} (a_{ij}^t)_{N \times N} \) [27]. One feasible choice of setting the weight \( a_{ij}^t \) is

\[
a_{ij}^t = \begin{cases} 1/\delta_{j}^{\text{out},t}, & \text{if } j \in N_i^{\text{in},t}, \\ 0, & \text{else.} \end{cases}
\]  

(4)

In the implementation, every agent \( j \) transmits the data \( x_j^t/\delta_{j}^{\text{out},t} \) and \( y_j^t/\delta_{j}^{\text{out},t} \) to its out-neighbors. With (3), the ratio \( z_i^t \triangleq x_i^t/y_i^t \) converges to the average of all the initial values \( \overline{x} = 1/N \sum_{i=1}^N x_i^0 \) [24] geometrically, i.e.,

\[
\lim_{t \to \infty} z_i^t = \overline{x}, \quad \forall i \in V.
\]

**Chebyshev Polynomial Approximation**

Chebyshev polynomial approximation focuses on using truncated Chebyshev series to approximate functions, thus facilitating numerical analysis. These series (i.e., approximations) are efficiently computed by interpolation. The degree \( m \)

\[\text{Chebyshev interpolant } p^{(m)}(x) \text{ corresponding to a Lipschitz continuous function } g(x) \text{ defined on } [a, b] \text{ takes the form as}

\[
p^{(m)}(x) = \sum_{j=0}^{m} c_j T_j \left( \frac{2x - (a + b)}{b - a} \right), \quad x \in [a, b],
\]

(5)

\ where \( c_j \) is the Chebyshev coefficient, and \( T_j(\cdot) \) is the \( j \)-th Chebyshev polynomial defined on \([-1, 1]\) and satisfies \( |T_j(x')| \leq 1, \forall x' \in [-1, 1] \). As \( m \) increases, \( p^{(m)}(x) \) converges uniformly to \( g(x) \) on the entire interval [29], i.e.,

\[
\forall x \in [a, b], \quad |p^{(m)}(x) - g(x)| \to 0, \quad m \to \infty.
\]

In practice, \( p^{(m)}(x) \) with a moderate degree \( m \) generally suffices to be a rather close approximation of \( g(x) \) [29]. The dependence of \( m \) on the smoothness of \( g(x) \) and the specified precision \( \epsilon \) can be found in Sec. IV-D. Consequently, computing \( p^{(m)}(x) \) becomes a practical way to construct an arbitrarily precise polynomial approximation for \( g(x) \), as theoretically ensured by the Weierstrass Approximation Theorem [29, Theorem 6.1].

### C. Models of Adversaries of Privacy

In this paper, we mainly consider honest-but-curious adversaries [23]. These adversaries are agents in the network that faithfully follow the specified protocol but intend to infer \( f_i(x) \) of the target agent \( i \) based on the received data. In terms of these adversaries, we are concerned with the issue of privacy disclosure arising in the consensus iterations of D-CPOA. As for the push-sum consensus protocol, the exchanged information serving as a basis for estimation consists of

\[
I_i^{\text{own},t} = \{a_{ij}, x_j^t\}, \quad I_i^{\text{in},t} = \{a_{ij}, x_j^t | j \in N_i^{\text{in},t}\},
\]

which are information sets of the states and weights of agent \( i \) and those transmitted from \( N_i^{\text{in},t} \) to agent \( i \) at time \( t \), respectively. As has been proven in [18], [19], the knowledge of \( \bigcup_{i \in \mathbb{N}} I_i^{\text{own},t} \), \( \bigcup_{i \in \mathbb{N}} I_i^{\text{in},t} \) and the coupling relationship between the locally added noises is a sufficient condition for the privacy compromise of the noise-adding-based privacy-preserving consensus protocols. We make the following assumption on the abilities of these adversaries.

**Assumption 4.** At every time \( t \), for the target agent \( i \), honest-but-curious adversaries can always access \( I_i^{\text{own},t} \) but can only obtain the full knowledge of \( I_i^{\text{in},t} \) with probability \( p \in (0, 1) \).

**Remark 1.** We assume the constant access of \( I_i^{\text{own},t} \) to include the scenario where some out-neighbors \( j \in N_i^{\text{out},t} \) are adversaries and can therefore always receive the information transmitted by agent \( i \), as considered in [18], [20]. The knowledge of \( I_i^{\text{in},t} \) is assumed to be available with probability \( p \) at time \( t \). The rationality is that the switching nature of time-varying networks can inhibit the persistent and perfect access to \( I_i^{\text{in},t} \). In practice, this setting holds if at time \( t \), there exists a trustworthy agent whose link with agent \( i \) occurs with probability \( 1 - p \), or the adversaries are mobile and contact agent \( i \) with probability \( p \) to gather \( I_i^{\text{in},t} [30] \).

---

1 As [27], we assume that \( i \in N_i^{\text{in},t}, \forall t \in \mathbb{N} \), i.e., agent \( i \) can always access its own information.
D. Privacy Definition

Without loss of generality, we consider the requirement of preserving the privacy of agent $i$’s local objective $f_i(x)$. In CPCPA, local communication happens in its second stage of average consensus iterations, where agents directly exchange and update their local variables $p_i^0 \in \mathbb{R}^{m_i+1}$. These variables are the vectors of coefficients of approximations $p_i(x)$ for local objectives $f_i(x)$. Once the adversaries obtain an estimation $\hat{p}_i$ of $p_i^0$, they can recover an approximation $\hat{f}_i(x) : X \to \mathbb{R}$ for $f_i(x)$. Note that $\hat{f}_i(x)$ is in the form of (5) with its coefficients stored in $\hat{p}_i$. Hence, $p_i^0$ is the sensitive information of $f_i(x)$ and its privacy needs to be preserved.

In this paper, we aim to design a secure average consensus algorithm for D-CPOA to effectively preserve the privacy of $f_i(x)$, or more specifically, $p_i^0$. This algorithm will be tailored to the case where agents own local variables of different dimensions. To characterize the privacy degree, we use $(\alpha, \beta)$-data-privacy, which is a comprehensive measure of the estimation accuracy and disclosure probability [20]. Let $\hat{p}_i$ be the estimation of $p_i^0$ based on the available information set $I$ and the predefined rule. The definition of $(\alpha, \beta)$-data-privacy is given as follows.

Definition 1. A distributed algorithm achieves $(\alpha, \beta)$-data-privacy for $p_i^0$ with a given $I$ if
\[
\Pr \{ \| \hat{p}_i - p_i^0 \|_1 \leq \alpha | I \} \leq \beta. \tag{6}
\]

In the above definition, $\alpha \geq 0$ and $\beta \geq 0$ are parameters that indicate the estimation accuracy and the bound for the disclosure probability of $p_i^0$, respectively. When $\alpha$ is specified, a smaller $\beta$ corresponds to a higher degree of privacy preservation. The original definition of $(\alpha, \beta)$-data-privacy in [20] considers the estimation of scalar states, and it is extended in this paper to handle vector states. We use the $l_1$-norm of the error $\| \hat{p}_i - p_i^0 \|$ to measure the estimation accuracy. This usage contributes to the neat relationship between the estimation accuracy of $p_i^0$ and that of $f_i(x)$, for $f_i(x)$ is closely approximated by $p_i^{(m_i)}(x)$, whose coefficients are stored in $p_i^0$.

Detailed discussions are provided in Remark 5.

III. DESIGN OF DEPENDABLE-CPOA

In this section, we present the design of Dependable-CPOA (D-CPOA). The proposed algorithm consists of three stages, whose details are discussed in the following three subsections.

A. Construction of Local Chebyshev Approximations

In this stage, every agent $i$ computes a polynomial approximation $p_i^{(m_i)}(x)$ of degree $m_i$ for $f_i(x)$ on $X = [a, b]$, s.t.
\[
| f_i(x) - p_i^{(m_i)}(x) | \leq \epsilon_1, \quad \forall x \in [a, b] \tag{7}
\]
holds, where $\epsilon_1 > 0$ is a specified tolerance. This goal is achieved via the adaptive Chebyshev interpolation method [31]. In this method, the degree of the interpolant is systematically increased until a certain stopping criterion is satisfied. The details are as follows. Agent $i$ sets $m_i = 2$ and begins to calculate a Chebyshev interpolant of degree $m_i$. It evaluates $f_i(x)$ at the set $S_{m_i} \triangleq \{ x_0, \ldots, x_{m_i} \}$ of $m_i + 1$ points by
\[
\begin{align*}
x_k &= \frac{b - a}{2} \cos \left( \frac{k\pi}{m_i} \right) + \frac{a + b}{2}, \\
f_k &= f_i(x_k),
\end{align*}
\tag{8}
\]
where $k = 0, 1, \ldots, m_i$. Then, it calculates the Chebyshev coefficients of the interpolant by
\[
c_j = \frac{1}{m_i} (f_0 + f_{m_i} \cos(j\pi)) + \frac{2}{m_i} \sum_{k=1}^{m_i-1} f_k \cos \left( \frac{jk\pi}{m_i} \right), \tag{9}
\]
where $j = 0, 1, \ldots, m_i$ [31]. At every iteration, the degree $m_i$ is doubled until the stopping criterion
\[
\max_{x_k \in (S_{2m_i} \setminus S_{m_i})} | f_i(x_k) - p_i^{(m_i)}(x_k) | \leq \epsilon_1 \tag{10}
\]
is met, where $S_{2m_i} \setminus S_{m_i}$ is the set difference of $S_{2m_i}$ and $S_{m_i}$, and $p_i^{(m_i)}(x)$ takes the form of (5) with $\{ c_j \}$ being its coefficients. Since $S_{m_i} \subset S_{2m_i}$, the evaluations of $f_i(x)$ are constantly reused. The intersection set $X = [a, b]$ of local constraint sets is known by running some numbers of max/min consensus iterations as (2) beforehand. Then, the obtained $p_i^{m_i}(x)$ will satisfy the requirement (7) on accuracy [31].

Remark 2. To handle problems where $X_i$ is an infinite or semi-infinite interval, we can use the technique of change-of-coordinate and choose rational Chebyshev functions as the basis for approximation, with certain technical assumptions on objective functions [32]. If local constraint sets are of mixed types (i.e., bounded, semi-infinite or infinite), the best practice is to make agents run max/min consensus protocols in advance to know the intersection set, and then construct local approximations on a common basis. Additionally, when $a_i$ and $b_i$ cannot be obtained accurately (e.g., due to uncertainties or random noises), we can use the idea of sampling from scenario optimization [33] and apply the proposed algorithm based on a random sample of constraints (i.e., scenarios) that are collected beforehand. The generalization property of the obtained solution can be characterized via the confidence bound for the constraint satisfaction [33].

B. Privacy-Preserving Information Dissemination

After the stage of initialization, each agent owns a local variable $p_i^0 \in \mathbb{R}^{m_i+1}$, which is the vector of coefficients of local polynomial approximation $p_i^{(m_i)}(x)$. In this stage, the goal is to enable agents to agree on the average $\bar{p} = 1/N \sum_{i=1}^{N} p_i^0$ of their initial values$^2$ via a distributed mechanism and, at the same time, the privacy of these initial values is preserved.

We propose a privacy-preserving scheme of information dissemination to achieve the aforementioned goal. The backbone of this scheme is the push-sum average consensus protocol [24]. The key ideas include i) adding random noises to $p_i^0$ to mask the true values, ii) inserting the components of the perturbed initial states block by block to hide them within iterations, thus obtaining a stronger privacy guarantee, and iii)

$^2$ In this expression of the average, those low dimensional vectors are extended with zeros when necessary to ensure the agreement in dimensions.
subtracting the noises separately in several randomly chosen rounds of iterations to guarantee the convergence to the exact average. The details are as follows.

First, every agent $i$ generates a noise vector $	heta_i \in \Theta^{m_i + 1}$ whose components are independent random variables in the domain $\Theta$, and adds $\theta_i$ to its initial state $p^0_i$ to form a perturbed state $\tilde{p}^0_i$, i.e.,

$$\tilde{p}^0_i = p^0_i + \theta_i.$$ Then, agents go on push-sum consensus iterations to exchange and update their local variables $x^t_i$ and $y^t_i$. The initial value of $y^0_i$ is set as 1 for all $i \in \mathcal{V}$. Nonetheless, instead of directly setting the initial value of $x^0_i$ as $\tilde{p}^0_i$, every agent $i$ will gradually extend $x^t_i$ with the components of $\tilde{p}^0_i$ in the first $K_1$ iterations. Let $(d_1^t, \ldots, d_{K_1}^t)$ be drawn from the multinomial distribution with parameters $m_i + 1$ and $\left(\frac{1}{K_1}, \ldots, \frac{1}{K_1}\right)$. Then,

$$\sum_{t=1}^{K_1} d_i^t = m_i + 1, \quad d_i^t \in \{0, \ldots, m_i + 1\}, \forall t.$$ Hence, $(d_1^t, \ldots, d_{K_1}^t)$ can be used to denote the numbers of components of $\tilde{p}^0_i$ that are inserted to $x^t_i$ at every iteration. Let

$$l_i^0 = 0, \quad l_i^t = \sum_{k=1}^{t} d_i^k, \quad t = 1, \ldots, K_1.$$ At the $t$-th iteration, the $(l_i^{t-1} + 1)$-th to $l_i^t$-th components of $x_i^t$ and $\tilde{p}^0_i$ are added together to form the corresponding components of $x_i^{t+1}$, where $t = 1, \ldots, K_1$. The remaining components of $x_i^t$ and $x_i^{t+1}$ are the same. Specifically,

$$x_i^{t+1}(k) = \begin{cases} x_i^t(k) + \tilde{p}^0_i(k), & \text{for } k = l_i^{t-1} + 1, \ldots, l_i^t; \\ x_i^t(k), & \text{else.} \end{cases}$$ Note that if the corresponding $x_i^t(k)$ is null, it is regarded as 0. That is, in the first case of (11), we add scalars and extend the sizes of vectors if necessary, thus avoiding the disagreement in dimensions. Then, agents transmit $x_i^{t+1}$ and $y_i^{t+1}$ to their out-neighbors and update $x_i^{t+1}$ and $y_i^{t+1}$ by

$$x_i^{t+1} = \sum_{j \in \mathcal{N}^\text{out}_i} a_{ij} x_j^{t+1}(k), \forall k; \quad y_i^{t+1} = \sum_{j \in \mathcal{N}^\text{out}_i} a_{ij} y_j.$$ Note that (12) also involves the extension of $x_i^t$. Hence, the dimension of $x_i^{t+1}$ is the same as the largest among $x_j^{t+1}$, where $j \in \mathcal{N}^\text{out}_i$. At the end of the $K_1$-th iteration, all the components of $\tilde{p}^0_i$ have been gradually inserted, and $x_i^t$ has at least $m_i + 1$ components. In the following $K_2 - K_1$ iterations, to guarantee the convergence to the exact average $\bar{p}$, every agent will properly subtract the added noises. Let $L$ be a random integer between 1 and $K_2 - K_1$ such that

$$|\zeta_i(k)| > \alpha, \quad \zeta_i(k) \triangleq \frac{\theta_i(k)}{L}.$$ Note that $L$ can be drawn from various discrete distributions, e.g., the discrete uniform, binomial and hypergeometric distributions. The choices of such distributions are up to the agents and are unknown to the adversaries. For the $k$-th component of $x_i^t(k = 1, \ldots, m_i)$, at $L$ randomly selected numbers of iterations, every agent $i$ subtracts a fraction of the added noise $\zeta_i(k)$ from the updated state, i.e.,

$$x_i^{t+1}(k) = \sum_{j \in \mathcal{N}^\text{out}_i} a_{ij} x_j^{t+1}(k), \quad (14a)$$

$$x_i^{(t+1)+}(k) = x_i^{t+1}(k) - \zeta_i(k), \forall k = 1, \ldots, m_i, \quad (14b)$$

$$y_i^{t+1} = \sum_{j \in \mathcal{N}^\text{out}_i} a_{ij} y_j, \quad (14c)$$ The update in (14a) involves the extension of dimensions if necessary. At the rest of the iterations, agents update their local variables by (12), where $x_i^{t+1}(k)$ is set as $x_i^t(k), \forall k, \forall t \geq K_2 + 1$.

To realize distributed stopping when the precision of iterations has met the requirement, we utilize the max/min-consensus-based stopping mechanism in [34] after the $K_2$-th iteration. Note that the scheme in [34] deals with static digraphs, but it can be easily extended to the settings with time-varying digraphs, given that in this case the max/min consensus protocols can still converge in finite time. The following assumption is required by this mechanism.

**Assumption 5.** Every agent $i$ in $\mathcal{G}$ knows an upper bound $U$ on $(N - 1)B$.

In practice, the bound $U$ can be obtained via the technique in [35] to estimate $N$ and the prior knowledge of $B$. Specifically, there are two auxiliary variables, i.e., $r_i^t$ and $s_i^t$, initialized as $p_i^{K_2} = x_i^{K_2}/y_i^{K_2}$ and updated together with $x_i^t$ and $y_i^t$ by

$$r_i^{t+1}(k) = \max_{j \in \mathcal{N}^\text{out}_i} r_j^t(k), \quad s_i^{t+1}(k) = \min_{j \in \mathcal{N}^\text{out}_i} s_j^t(k), \forall k.$$ Note that the number of iterations that the max/min consensus protocols require to converge is less than $(N - 1)B$ [36], and is therefore less than $U$. Thus, at time no later than $t = K_2 + U$, all the local variables $x_i^t$, $y_i^t$, $r_i^t$ and $s_i^t$ become $(m + 1)$-dimensional vectors, where

$$m \triangleq \max_{i \in \mathcal{V}} m_i$$ is the maximum degree of all the local approximations. The variables $r_i^t$ and $s_i^t$ are reinitialized as $p_i^t$ every $U$ iterations. Hence, the recent information on $p_i^t$ is continually disseminated. When the stopping criterion

$$\|r_i^K - s_i^K\|_{\infty} \leq \delta \triangleq \frac{\epsilon_2}{m + 1}, \quad (16)$$ is satisfied at the $K$-th iteration, agents terminate the iterations and set $p_i^K = x_i^K/y_i^K$.

**C. Polynomial Optimization by Solving SDPs**

In this stage, agents locally optimize the polynomial proxy $p_i^K(x)$ recovered from $p_i^K$ on $X = [a, b]$ to obtain $c$-optimal solutions of problem (1). This optimization problem is transformed to a semidefinite program (SDP) based on sum-of-squares decomposition of non-negative polynomials [37]. It can then be efficiently solved via the interior-point method [38]. We provide such reformulations with the Chebyshev coefficients of $p_i^K(x)$.
Note that $p^K_1(x)$ is a polynomial of degree $m$ and takes the form of (5). The components of $p^K_1 = [c'_0, \ldots, c'_m]^T$ are its coefficients. To simplify the notation, let

$$g^K_1(x') \triangleq p^K_1 \left( \frac{b - a}{2} x' + \frac{a + b}{2} \right) = \sum_{j=0}^{m} c'_j T_j(x'), \ x' \in [-1, 1].$$

The optimal values of $p^K_1(x)$ on $[a, b]$ and $g^K_1(x')$ on $[-1, 1]$ are the same, and the optimal points $x^*_p$ and $x^*_g$ satisfy

$$x^*_p = \frac{b - a}{2} x^*_g + \frac{a + b}{2}. \quad (17)$$

Hence, we turn to solve the following problem

$$\min g^K_1(x), \ s.t. \ x \in [-1, 1], \quad (18)$$

and then use (17) to obtain the optimal value and optimal points of $p^K_1(x)$ on $[a, b]$. To this end, we first transform problem (18) to its equivalent problem

$$\max t, \ s.t. \ g^K_1(x) - t \geq 0, \ \forall x \in [-1, 1]. \quad (19)$$

The equivalence follows from the fact that $(x^*, t^*)$ is optimal for problem (19) if and only if $x^*$ is optimal for problem (18) and $t^* = g^K_1(x^*)$. Then, we introduce new optimization variables $Q, Q'$ that are chosen from the set of symmetric positive semidefinite matrices $S_+$ and reformulate the non-negativity constraint in (19) based on the Markov-Lukács theorem [37, Theorem 3.72] and sum-of-squares decomposition. When $m$ is odd, problem (19) is transformed to

$$\max_{t, Q, Q'} t$$

s.t. $c_0 = t + Q_{00} + Q'_{00} + \frac{1}{2} \left( \sum_{u=1}^{d_1+1} Q_{uu} + \sum_{u=1}^{d_2+1} Q'_{uu} \right)$,

$$c_j = \frac{1}{2} \left( \sum_{u,v \in A} (Q_{uv} + Q'_{uv}) \right),$$

$$+ \frac{1}{4} \sum_{(u,v) \in B} (Q_{uv} - Q'_{uv}),$$

$$Q \in S_{d_1+1}^+, \ Q' \in S_{d_2+1}^+,$$

where the rows and columns of $Q$ and $Q'$ are indexed by $0, 1, \ldots, d_1$, and $d_1 = \lfloor m/2 \rfloor$, $d_2 = \lfloor m - 1/2 \rfloor$,

$$A = \{(u, v) \mid u + v = i \lor |u - v| = i\},$$

$$B = \{(u, v) \mid u + v = i - 1 \lor |u - v| = i - 1 \lor |u + v - 1| = i \lor |u - v - 1| = i\}.$$

When $m$ is even, the transformed problem takes a similar form. We refer readers to our work [16] for more details on the forms and sensibilities of these reformulations. \footnote{Such SDP reformulations are only dependent on the Chebyshev coefficients of the polynomial $p^K_1(x)$ to be optimized, and are independent of the network topology. Hence, the analysis of the reformulations in [16], which considers static undirected networks, also applies to this paper.}

**Algorithm 1** D-CPOA

**Input:** $f_i(x), X_i = [a_i, b_i], U_i$ and $\epsilon$.

**Output:** $f^*_i$ for every agent $i \in V$.

1. Initialize: $a_i^0 = a_i, b_i^0 = b_i, m_i = 2$.
2. for each agent $i \in V$ do
3. for $t = 0, \ldots, U - 1$ do
4. $a_i^{t+1} = \max a_i^t, b_i^{t+1} = \min b_i^t$.
5. end for
6. Set $a_i^t = a_i^*, b_i^t = b_i^*$.
7. Calculate $\{x_i^t\}$ and $\{f_i^t\}$ by (8).
8. Calculate $\{c_{i,k}\}$ by (9).
9. If (10) is satisfied (where $\epsilon_1 = \epsilon/3$), go to step 10. Otherwise, set $m_i \leftarrow 2m_i$ and go to step 7.
10. Set $p_i^t = p_i^0 + \eta_t, x_i^0 = \text{null}, y_i^0 = 1, (d_i^0, \ldots, d_i^{K_1})$ drawn from the multinomial distribution with parameters $m_i + 1$ and $1/K_1(1, \ldots, 1), t = 1$.
11. for $t = 0, 1, \ldots$ do
12. if $t \leq K_1$ then
13. Extend $x_i^t$ to form $x_i^{t+1}$ by (11).
14. Update $x_i^{t+1}, \forall k$ and $y_i^{t+1}$ by (12).
15. else if $K_1 + 1 \leq t \leq K_2$ then
16. for each component $k = 1, \ldots, m_i$ do
17. Update $x_i^{t+1}(k), \forall k$ and $y_i^{t+1}$ by (12), or by (14) if subtractions are performed at certain iterations.
18. end for
19. else
20. if $t = 1U$ then
21. if $\|r_i^t - s_i^0\|_\infty \leq \epsilon_2/(m + 1)$ then
22. $p_i^t = x_i^t/y_i^t$.
23. break
24. $r_i^t = s_i^0 = p_i^t, l \leftarrow l + 1$.
25. end if
26. Update $x_i^{t+1}(k), \forall k$ and $y_i^{t+1}$ by (3).
27. end if
28. end for
29. Solve the reformulated problem, e.g., (20), with $\epsilon_3 = \epsilon/3$ and return $f_i^*$.
30. end for

The aforementioned transformed problems are SDPs, and therefore can be efficiently solved via the primal-dual interior-point method [38]. The iterations of this method are terminated when

$$0 \leq f^*_i - p^* \leq \epsilon_3,$$

where $f^*_i$ is the obtained estimate of the optimal value $p^*$ of $p^K_1(x)$ on $X = [a, b]$, and $\epsilon_3 > 0$ is the specified precision. The optimal points of $g^K_1(x)$ are computed from the complementary slackness condition [37]. The optimal points of $p^K_1(x)$ on $X$ can then be calculated by (17).

The full details of the proposed algorithm are summarized as Algorithm 1. We set all the precision used in three stages, i.e., $\epsilon_1, \epsilon_2$ and $\epsilon_3$, as $\epsilon/3$, such that their sum equals $\epsilon$ and then the reach of $\epsilon$-optimality is ensured.

**Remark 3.** The proposed algorithm can be adjusted to handle nonconvex local constraint sets of the form $X_i = (\bigcup_{k} X_i,k) \cup \left( \bigcup_{k'} \{a_i, k'\} \right)$, where $X_i,k$ is a closed convex set (i.e., a closed interval), and $\{a_i, k'\}$ is a singleton. The main steps include i) constructing piecewise polynomial approximations for local objective functions on $X_i$, ii) going
on privacy-preserving iterations to exchange the vectors of coefficients of approximations and the associated subdomains (i.e., $X_{i,k}$) and singletons, and iii) optimizing the recovered proxy on the obtained intersection set $X$ via the stationary-point-based method (which is discussed in Sec. V-D of [16] and helps to avoid the trouble of solving SDPs at every subdomain and reduce the overall computational costs).

IV. PERFORMANCE ANALYSIS

A. Accuracy

We establish the accuracy of D-CPOA in this subsection. The following lemma guarantees the accuracy of the consensus-based iterations in the proposed algorithm.

Lemma 1. If Assumptions 3 and 5 hold, when (16) is satisfied, we have

$$\max_{i \in \mathcal{V}} \|p^K_i - \bar{p}\|_\infty \leq \delta = \frac{\epsilon_2}{m+1}. \quad (21)$$

Proof. The proof is provided in Appendix A. \Box

In the following theorem, we characterize the distance between the obtained solution $f^*_c$ and the optimal value $f^*$ of problem (1), and the distance between the optimal point $x^*_p$ of $p^K_i(x)$ on $X$ (i.e., the returned solution) and the optimal point $x^*_f$ of problem (1).

Theorem 2. Suppose that Assumptions 1-5 hold. D-CPOA ensures that every agent obtains $\epsilon$-optimal solutions $f^*_c$ for problem (1), i.e.,

$$|f^*_c - f^*| \leq \epsilon,$$

Moreover,

$$|x^*_p - x^*_f| \leq \text{diam}(S), \quad S = \{x \in X | f(x) \leq f(x^*_f) + \frac{\epsilon}{3}\}.$$

Proof. The proof is provided in Appendix B. \Box

Note that in Theorem 2, $\epsilon$ is any arbitrarily small specified precision, and $\text{diam}(S)$ is the diameter of $S$, i.e., the maximum distance between any two points in $S$.

Remark 4. If each agent $i$ has a different error tolerance $\epsilon_i$, then D-CPOA still works, but the accuracy will be the largest value of local tolerances, i.e., $\max_{i \in \mathcal{V}} \epsilon_i$. This phenomenon chiefly results from the change in the accuracy of constructing $p^{(m)}_i(x)$. We note that for any $x \in X$, the difference between $\bar{p}(x)$ (i.e., the average of all $p^{(m)}_i(x)$) and $f(x)$ satisfy

$$|\bar{p}(x) - f(x)| = \left| \frac{1}{N} \sum_{i=1}^{N} (p^{(m)}_i(x) - f_i(x)) \right| \leq \frac{1}{N} \sum_{i=1}^{N} |p^{(m)}_i(x) - f_i(x)| \leq \frac{1}{N} \sum_{i=1}^{N} \epsilon_i, \quad \epsilon_i \leq \epsilon_{i,1},$$

where $\epsilon_{i,1}$ is the specified tolerance corresponding to $p^{(m)}_i(x)$ and can be set as $\epsilon_i/3$. Based on the proof similar to that of Theorem 2, the difference between $p^K_i(x)$ and $f(x)$ on $X$ will not exceed $\max_{i \in \mathcal{V}} 2\epsilon_i/3$. Since the bound for the accuracy of solving SDPs locally is $\max_{i \in \mathcal{V}} \epsilon_i/3$, it follows that in this case, the obtained solution is $(\max_{i \in \mathcal{V}} \epsilon_i)$-optimal.

B. Data-Privacy

In this subsection, we show that the developed algorithm preserves the privacy of $p^0_i$ and investigate the privacy-preserving property through the notion of data-privacy [20]. We first define the information set $I^t_i$ used by the adversaries at time $t$ for state estimation. Let

$$I^t_i = I^{\text{own},t}_i \cup I^{\text{in},t}_i,$$

where

$$I^{\text{own},t}_i = \bigcup_{s=1}^{t} I^{\text{own},s}_i = \bigcup_{s=1}^{t} \{a_{i,s}^s, x_{i,s}^s\},$$

$$I^{\text{in},t}_i = \bigcup_{s \in S_t} I^{\text{in},s}_i = \bigcup_{s \in S_t} \{a_{i_j,s}^s, x_{i_j,s}^s|j \in N^{s}_i\}.$$

The set $S_t$ contains those numbers of iterations $s (s \leq t)$ when the adversaries have obtained the full knowledge of $I^{\text{in},s}_i$. Note that $I^{\text{in},s}_i$ consists of all the available information on the states and weights owned by and transmitted to agent $i$ up to the $t$-th iteration. Let $X$ be a random variable whose distribution and any other relevant information are unknown. Since $X$ can be any arbitrary value in its domain, it is reasonable to assume that the probability that an accurate enough estimation $\hat{X}$ of $X$ can be obtained is rather small [19], i.e.,

$$\Pr\{|\hat{X} - X| \leq \alpha\} \leq \gamma,$$

where $\alpha \geq 0$ and $\gamma \geq 0$ are small given constants. The bound $\gamma$ for the disclosure probability satisfies

$$\gamma \ll \max_{i \in \mathcal{E}} \int_{-\alpha}^{\alpha} f_{\theta_i(k)}(y)dy, \quad \forall k,$$

where $f_{\theta_i(k)}(y)$ is the probability density function (PDF) of the added noise $\theta_i(k)$.

Recall that we aim to preserve the privacy of $p^0_i \in \mathbb{R}^{m_i+1}$. Thanks to the blockwise insertions in (11), the adversaries are unaware of the exact value of $m_i$. They do know $m_i$, however, based on the received $p^0_i$. Hence, the estimation of $p^0_i$ consists of two parts: to estimate its components $p^0_i(k)$, where $k = 1, \ldots, m_i + 1$, and to infer that $p^0_i(k)$ is null for $k = m_i + 2, \ldots, m + 1$. Let $\alpha$ and $\alpha_k$ be the estimation accuracy of $p^0_i$ and each component $p^0_i(k)$, respectively, s.t.,

$$\sum_{k=1}^{m_i+1} \alpha_k = \alpha, \quad \alpha_k \in [0, \alpha], \quad \forall k = 1, \ldots, m_i + 1. \quad (23)$$

It follows that

$$\|\hat{p}_i - p^0_i\|_1 = \sum_{k=1}^{m_i+1} |\hat{p}_i(k) - p^0_i(k)| \leq \sum_{k=1}^{m_i+1} \alpha_k = \alpha,$$

Hence, we can sequentially consider the relationship between the estimation accuracy $\alpha_k$ and the disclosure probability $\beta_k$ of each component $p^0_i(k)$, and then synthesize them to obtain the result concerning $p^0_i$. Also, since $m_i$ is unknown and varies with $\epsilon_i$ and $f_i(x)$, it is viewed as a random variable by the adversaries. Let $F_{m_i,t}(-\cdot)$ be the cumulative distribution function of $m_i$ given $I^t_i$. The following theorem characterizes the effects of privacy preservation of the proposed algorithm.
Theorem 3. If Assumptions 3 and 4 hold, given $T_1$, D-CPOA achieves $(\alpha, \beta)$-data-privacy for $p^*_0$, where \{\alpha_k\} satisfies (23),
\[
\beta = \prod_{k=1}^{m+1} \beta_k, \quad \beta_k = \prod_{k=m+2}^{m+1} F_{m+1}(k-2),
\]
and
\[
h_1(\alpha_k) = p \max_{\nu \in \Theta^k} \int_{\nu+\alpha_k}^{\nu-\alpha_k} f_\theta(k)(y)dy + \gamma.
\]
Proof. The proof is provided in Appendix C.

Theorem 3 states that D-CPOA preserves the privacy of $p^*_0$. The effects of privacy preservation are evaluated through $(\alpha, \beta)$-data-privacy. The interpretation of $\beta$ in (24) is as follows. Note that $\beta$ is the product of a set of bounds $\beta_k$ for disclosure probabilities corresponding to the components $p^*_0(k)$, $\forall k = 1, \ldots, m+1$ (see (37)), and the probabilities of correctly identifying null components (see (38)). The bounds $\beta_k$ are derived via the law of total probability. Suppose that agent $i$ inserts its perturbed state $p^*_0(k)$ at time $s$. If the event that the adversaries know $I_i^{m,t}$ for time $t = s - 1$ and any time $t$ between $K_1 + 1$ and $K_2$ happens (the probability of which is not more than $p_{K_2-K_1+1}$), then the added noises $\theta_i(k)$ and states $p^*_0(k)$ can be perfectly inferred. Otherwise, the disclosure probability will not exceed $h_1(\alpha_k)$. The bounds $h_1(\alpha_k)$ are derived likewise based on whether the adversaries know $I_i^{m,s-1}$. If the adversaries know this information, then the maximum disclosure probability is
\[
\max_{\nu \in \Theta} \int_{\nu+\alpha_k}^{\nu-\alpha_k} f_\theta(k)(y)dy,
\]
which equals to the probability that the optimal distributed estimation falls into $[p^*_0(k) - \alpha_k, p^*_0(k) + \alpha_k]$ [20]. Otherwise, the disclosure probability is rather small since the adversaries own little relevant information of $p^*_0$. The probabilities of correct decision of null components are derived based on whether the index $k$ exceeds $m_k + 1$ (i.e., the dimension of $p^*_0$, which is viewed as a random variable by the adversaries). The design of blockwise insertions causes adversaries to additionally identify null components, thus further reducing the disclosure probability of $p^*_0$.

From (24), we know that for those $p^*_0$ of larger sizes (i.e., with larger $m_k$), $\beta$ will generally be smaller, which implies a higher degree of privacy preservation. In addition, $\beta$ increases with $\alpha_k$ but decreases with $K_2 - K_1$. These relationships support the intuitions that less accurate estimations can be acquired with higher probabilities, and more room for randomness leads to lower probabilities of privacy disclosure.

Remark 5. In this paper, we investigate the privacy-preserving property of D-CPOA by studying its degree of data-privacy for $p^*_0$. The reasons are twofold. First, this degree directly reflects the effectiveness of the incorporated privacy-preservation mechanism, as $p^*_0$ is exactly the initial value calling for protections in the iterations. Second, this degree is closely related to the effects of privacy preservation of $f_i(x)$.

In (6), if $\|\hat{p}_i - p^{0(m)}_i\|_1 \leq \alpha$, i.e., a fairly precise estimation $\hat{p}_i$ of $p^{0}_i$ is obtained, then $\forall x \in X = [a, b]$, we have
\[
|\hat{f}_i(x) - p^{0(m)}_i(x)| = \left| \sum_{k=0}^{m} (\hat{p}_i(k) - p^{0}_i(k)) I_k \left( \frac{2x - (a + b)}{b - a} \right) \right|
\]
\[
\leq \sum_{k=0}^{m} |\hat{p}_i(k) - p^{0}_i(k)| \cdot 1 = \|\hat{p}_i - p^{0}_i\|_1 \leq \alpha.
\]
By referring to (7), it follows that
\[
|\hat{f}_i(x) - f_i(x)| \leq \alpha + \epsilon_1,
\]
i.e., an accurate enough estimation $\hat{f}_i(x)$ of $f_i(x)$ is acquired. Nevertheless, to derive the requirement of closeness between $\hat{p}_i$ and $p^{0}_i$ from that between $\hat{f}_i(x)$ and $f_i(x)$ (or $p^{0(m)}_i(x)$) is very difficult due to the coupling of terms in (25). Hence, the characterization of the degree of data-privacy for $p^{0}_i$ is the main focus of this paper.

C. Discussions on Dependability

In this section, we discuss the dependability of the proposed algorithm, considering various requirements including privacy preservation and robustness to network imperfections. We summarize the comparisons of the performance of D-CPOA and other typical algorithms in Table I. The details are given as follows.

- Privacy Guarantee. We have shown in Theorem 3 that the consensus-based iterations of D-CPOA preserve the privacy of sensitive $p^*_0$ and analyzed the effects of preservation through $(\alpha, \beta)$-data-privacy. Next, we study such effects via differential privacy, which provides a strong privacy guarantee when in face of adversaries owning arbitrarily much auxiliary information [4, 21, 39]. We define the database $D$ and the randomized query output $M(D)$ as the set of initial states and the set of transmitted states of consensus protocols, i.e.,
\[
D = \{p^*_0|\forall i \in \mathcal{V}\}, \quad M(D) = \{x^+_t(t)|\forall t \in \mathbb{N}, i \in \mathcal{V}\},
\]
respectively. Based upon [21, 39], in our setting, a privacy-preserving consensus protocol is $\epsilon$-differentially private if
\[
\Pr \{M(D) \in O\} \leq e^\epsilon \Pr \{M(D') \in O\}
\]
holds for any $O \subseteq \text{range}(M)$ and $\sigma$-adjacent $D, D'$ satisfying
\[
\|p^*_0 - (p^{0}_i)\|_1 \leq \sigma, \quad \text{if } i = i_0,
\]
\[
0, \quad \text{if } i \neq i_0
\]
for all $i \in \mathcal{V}$, where $i_0$ is some element in $\mathcal{V}$. Note that we have used correlated noises (see (13)) to pursue the proximity of $p^*_0$ to the exact average $\hat{p}$ (see Lemma1), thus ensuring the accuracy of the obtained solutions (see Theorem3). Based on the impossibility result of simultaneously achieving exact average consensus and differential privacy [21, 39], we conclude that our algorithm is currently not $\epsilon$-differentially private. If we want to pursue differential privacy at the cost of losing some accuracy of the obtained solutions, we can add uncorrelated noises that satisfy the condition in [39, Theorem 4.3] (e.g., independent Laplace noises) to the transmitted states at every iteration. In that case, the proposed algorithm becomes
differentially private, though the obtained solutions will suffer from random errors due to the inexact convergence of its consensus-based iterations.

**Remark 6.** In [40], the authors present a mechanism that both ensures the accuracy and preserves the privacy of local objectives in distributed optimization. Though this work and [40] share an idea of decomposing local objectives via basis functions, there exist notable differences in the design and analysis of privacy preservation, the schemes of iterations and optimization, and the generality of results. First, we leverage the randomness in blockwise insertions of perturbed initial states to obtain a stronger privacy guarantee in terms of data-privacy [20]. Second, we employ gradient-free iterations where information on local proxies is exchanged and local polynomial optimization to obtain solutions, while [40] relies on distributed gradient descent [5] to reach convergence. Finally, we acquire arbitrarily precise estimates of globally optimal solutions of nonconvex problems. The problem setting and solution quality of this work are more general compared to those of [40].

- **Asynchrony.** We discuss the asynchronous extension of the proposed algorithm. Compared to synchronous models, asynchronous paradigms are more desirable in applications for its increased efficiency in handling uncoordinated computations and imperfect communication, e.g., transmission delays and packet drops. The design of consensus-based information dissemination presented in Algorithm 1 is synchronous. Its extension to cope with asynchrony is readily available and can benefit from the extensive research on asynchronous consensus protocols, including those allowing for random activations (e.g., gossip algorithms [41]), delays [42], packet drops [43] and all these issues [15]. In these protocols, the basic idea of proving convergence is to first transform asynchronous models to synchronous counterparts over augmented graphs, where virtual nodes and edges are added to facilitate the analysis, and then establish the convergence of synchronous models. All these asynchronous protocols converge deterministically to the average of initial values. If they are incorporated into the proposed algorithm, by Lemma 1 and Theorem 2, the accuracy of the obtained solutions can still be guaranteed. In addition, since the iterations of the developed algorithm are consensus-based and do not involve gradients, there is no need to select varying step-sizes in different circumstances of asynchrony.

**D. Complexity**

We present a lemma on the dependence of the degree $m_i$ of the local approximation $p_i^{(m_i)}(x)$ on the specified tolerance $\epsilon_1$ and the smoothness of the local objective function $f_i(x)$.

**Lemma 4** ([16]). If $f_i(x)$ and its derivatives through $f_i^{(v)}(x)$ are absolutely continuous and $f_i^{(v)}(x)$ is of bounded variation on $X_i$, then $m_i \sim O(\epsilon_1^{-1/v})$. If $f_i(x)$ is analytic on $X_i$, then $m_i \sim O(\ln(1/\epsilon_1))$.

Lemma 4 suggests that for functions that are smooth to some extent, approximations of moderate degrees (e.g., of the order of $10^1 \sim 10^2$) can serve as rather accurate representations. The following theorem describes the complexities of D-CPOA in terms of $m$ and $\epsilon$, which equal to $\max_{i \in V} m_i$ and $3\epsilon_1$, respectively. We measure the computational complexity via the order of flops [4], and use $F_0$ to denote the cost of flops in one evaluation of $f_i(x)$.

1. “scvx” refers to “strongly-convex” objective functions.
2. “ncvx” refers to “nonconvex” objective functions.
3. The convergence time is $O(\frac{1}{\epsilon^3})$, implying that both the complexities of evaluations of gradients (i.e., queries of the first-order oracle) and those of inter-agent communication are $O(\frac{1}{\epsilon^2})$. 4. “DP” stands for “differential privacy”. 5. There is a trade-off between accuracy and privacy. 6. This symbol means “same as above”. 7. In [9], the authors proposed a general strategy of function perturbation to ensure differential privacy. This strategy can be combined with any distributed convex constrained optimization algorithms to take effect. Hence, we place "✓" to some blocks in this row to imply feasibility. 8. Detailed discussions are provided in Sec. IV-C. 9. See Lemma 4 and Theorem 5 for details.
local objective functions, optimal solutions for problem (1) with objective functions to highlight the advantages brought by the E. Discussions on Multivariate Extensions algorithm brings no extra costs in terms of complexities. Hence, we conclude that the dependability of the proposed needed numbers of iterations (i.e., inter-agent communication). and thus they only change the values but not the orders of the of added noises. These actions are completed in finite time, i.e., in finite time, i.e., in iterations. Since the consensus-based protocol converges geometrically, the order of the total number of iterations (i.e., inter-agent communication) is of

\[ K_2 + \mathcal{O}\left(\log \frac{1}{\delta}\right) = \mathcal{O}\left(\log \frac{1}{\delta}\right) = \mathcal{O}\left(\log \frac{m}{\epsilon}\right), \]

where the required precision \( \delta \) is given by (16). The order of flops needed in this stage is of \( \mathcal{O}(m \log \frac{m}{\epsilon}) \). The results in the theorem follow from the above analysis.

The comparisons of the complexities of D-CPOA and those of CPCA [16] are shown in Table II. We observe that the complexities of these two algorithms are the same. The reasons are as follows. The major difference between the two algorithms lies in the stage of consensus-based information dissemination. In this stage, D-CPOA fulfills privacy preservation by utilizing the randomness of insertions of block-data and subtractions of added noises. These actions are completed in finite time, and thus they only change the values but not the orders of the needed numbers of iterations (i.e., inter-agent communication). Hence, we conclude that the dependability of the proposed algorithm brings no extra costs in terms of complexities.

E. Discussions on Multivariate Extensions

In this paper, we mainly consider problems with univariate objective functions to highlight the advantages brought by the

\[ \text{Algorithm} \quad \theta^m \text{order Oracles} \quad \text{Communication} \quad \text{PD Iterations} \]

| Alg  | \( \mathcal{O}(m) \) | \( \mathcal{O}\left(\log \frac{m}{\epsilon}\right) \) | \( \mathcal{O}\left(\sqrt{m \log \frac{1}{\epsilon}}\right) \) |
|------|------------------|------------------|------------------|
| CPCA | \( \mathcal{O}(m) \) | \( \mathcal{O}\left(\log \frac{m}{\epsilon}\right) \) | \( \mathcal{O}\left(\sqrt{m \log \frac{1}{\epsilon}}\right) \) |
| D-CPOA | \( \mathcal{O}(m) \) | \( \mathcal{O}\left(\log \frac{m}{\epsilon}\right) \) | \( \mathcal{O}\left(\sqrt{m \log \frac{1}{\epsilon}}\right) \) |

\( ^1 \) Compared with CPCA, D-CPOA generally requires more inter-agent communication to reach certain precision. This increase results from potential network imperfections and extra steps of insertions and subtractions, which may slow down the convergence rates. Nonetheless, the communication complexities of both algorithms are the same (see proof of Theorem 5).

Theorem 5. D-CPOA ensures that every agent obtains \( \epsilon \)-optimal solutions for problem (1) with \( \mathcal{O}(m) \) evaluations of local objective functions, \( \mathcal{O}\left(\log \frac{m}{\epsilon}\right) \) rounds of inter-agent communication, \( \mathcal{O}\left(\sqrt{m \log \frac{1}{\epsilon}}\right) \) iterations of primal-dual interior-point methods and \( \mathcal{O}\left(m \cdot \max(m^{3.5} \log \frac{1}{\epsilon}, F_0)\right) \) flops.

Proof. Note that the evaluations of local objective functions (i.e., queries of the zeroth-order oracle) are only performed in the stage of initialization, and the primal-dual interior-point method [38] is used to solve problem (20) in the stage of polynomial optimization. By referring to the proof of [16, Theorem 6], we know that for every agent, the orders of evaluations of local objective functions and primal-dual iterations are of \( \mathcal{O}(m) \) and \( \mathcal{O}\left(\sqrt{m \log \frac{1}{\epsilon}}\right) \), respectively, where \( m \) is the maximum degree of local approximations. Also, the orders of the required flops of these two stages are of \( \mathcal{O}(m \cdot \max(m, F_0)) \) and \( \mathcal{O}(m^{4.5} \log \frac{1}{\epsilon}) \), respectively.

In the stage of information dissemination, the insertions of block-data and subtractions of noises are completed in finite time, i.e., in \( K_2 \) iterations. Since the consensus-based protocol converges geometrically, the order of the total number of iterations (i.e., inter-agent communication) is of

\[ K_2 + \mathcal{O}\left(\log \frac{1}{\delta}\right) = \mathcal{O}\left(\log \frac{1}{\delta}\right) = \mathcal{O}\left(\log \frac{m}{\epsilon}\right), \]

where the required precision \( \delta \) is given by (16). The order of flops needed in this stage is of \( \mathcal{O}(m \log \frac{m}{\epsilon}) \). The results in the theorem follow from the above analysis.

The comparisons of the complexities of D-CPOA and those of CPCA [16] are shown in Table II. We observe that the complexities of these two algorithms are the same. The reasons are as follows. The major difference between the two algorithms lies in the stage of consensus-based information dissemination. In this stage, D-CPOA fulfills privacy preservation by utilizing the randomness of insertions of block-data and subtractions of added noises. These actions are completed in finite time, and thus they only change the values but not the orders of the needed numbers of iterations (i.e., inter-agent communication). Hence, we conclude that the dependability of the proposed algorithm brings no extra costs in terms of complexities.

E. Discussions on Multivariate Extensions

In this paper, we mainly consider problems with univariate objective functions to highlight the advantages brought by the idea of using polynomial approximation, e.g., achieving efficient optimization of nonconvex problems and readily allowing for enhancement to be dependable when diverse practical needs exist. We now discuss the possibility of multivariate extensions of the proposed idea. The differences will mainly lie in the stage of initialization and that of optimization of approximations. Specifically, let \( L_2(X) \) be the set of square-integrable functions over \( X \subset \mathbb{R}^n \) and \( f_i(x) \in L_2(X) \) be a general local objective. Then, there exists an orthonormal basis \( \{h_k(x)\}_{k \in \mathbb{N}} \) (e.g., orthonormalization of Taylor polynomials) and an arbitrarily precise approximation

\[ \hat{f}_i(x) = \sum_{k=1}^{m} c_k h_k(x) \]

for \( f_i(x) \), where \( \{c_k\}_{k=1}^{m} \) is the set of coefficients. Afterward, agents can exchange and update their local variables storing these coefficients (as in Sec. III-B) and acquire an approximation for the global objective function. Finally, they can locally optimize this approximation via the tools for polynomial optimization or for finding stationary points of general nonconvex functions [44], thus obtaining desired solutions. Nevertheless, the aforementioned idea of extensions calls for further investigation and more careful analysis and is still among our ongoing work.

V. NUMERICAL EVALUATIONS

In this section, we perform numerical experiments to illustrate the performance of D-CPOA. We consider a network with \( N = 20 \) agents. At each time \( t \), besides itself, every agent \( i \) has two out-neighbors. One belongs to a fixed cycle, and the other is chosen uniformly at random. Hence, \( \{G^t\} \) is 1-strongly-connected. We assume that all the local constraint sets are the same interval \( X = [-1, 1] \) and the local objective function \( f_i(x) \) of agent \( i \) is

\[ f_i(x) = \frac{a_i}{1 + e^{-x}} + b_i \log(1 + x^2), \]

where \( a_i \sim \mathcal{N}(10, 2) \) and \( b_i \sim \mathcal{N}(5, 1) \) are normally distributed. It follows that \( f_i(x) \) is nonconvex and Lipschitz continuous on \( X \). Chebfun toolbox [29] is used to construct Chebyshev polynomial approximations \( p_i(x) \) corresponding to all the local objective functions \( f_i(x) \).

The convergence of the proposed algorithm is shown in Fig. 1(a). In the experiment, we set \( K_1 = 10, K_2 = 20 \). We generate i.i.d. random noises \( \theta_i(k) \) from the uniform distribution \( U(-1, 1) \) and randomly select \( L \) from the discrete uniform distribution \( U\{1, K_2 - K_1\} \) to satisfy (13). In Fig. 1(a), the square markers on the blue line indicate how many numbers of iterations \( t \) of information dissemination have been performed, when certain precisions \( \epsilon \) are specified. The triangle markers on the orange line represent what the actual values of objective errors \( |f^* - f^*| \) are, when those numbers of iterations are completed. We observe that the relationship between \( \log \epsilon \) and \( t \) is roughly linear. This phenomenon results from the property of linear convergence of the consensus-based information dissemination in the developed algorithm.

The effects of privacy preservation are presented in Fig. 1(b). This figure demonstrates the relationships between
estimation accuracy $\alpha_k$ and bound $\beta_k$ for the disclosure probability for a single component $p^i_k$ when different types of noises $\theta_i(k)$ are used. These relationships are explicitly characterized by (37) in Appendix C. In the experiment, we set $K_1 = 10, K_2 = 20, p = 0.8$ and $\gamma = 10^{-6}$. We consider three types of noises that satisfy uniform, normal and Laplace distributions. We assume that the mean and variance of these noises are $0$ and $1$, respectively. We observe that $\beta_k$ increases with $\alpha_k$, which confirms the intuition that a less accurate estimate can be obtained with a higher probability. We also notice that uniformly distributed noises yield the smallest $\beta_k$ and thus the most effective preservation of $p^i_k$. This observation supports the conclusion in [20]. Note that the bound $\beta$ for the disclosure probability of $p^i_k$ is the product of all $\beta_k$ for $k = 1, \ldots, m_i + 1$ (see (24) in Sec. IV).

The degrees $m_i$ of local approximations constructed in this experiment roughly vary from 20 to 30 when the specified precision $\epsilon = 10^{-10}$. Hence, in this case, $\beta$ is an extremely small number given $\alpha_k$ and $\beta_k$ in the figure.

Specifically, we study the convergence rates of the consensus-based iterations incorporated with the proposed privacy-preserving mechanism. The initial states of agents are set as the vectors of coefficients of local approximations. The rest of the settings are the same as those in the study of the convergence of D-CPOA. We also implement SCDA in [22] and a differentially private consensus protocol [21], [39] for comparison, where uniformly distributed and Laplace distributed noises are added to every component of local variables, respectively. In all three protocols, the initially added noises are of zero mean and variance $1/3$. In the last two protocols, the variances of the added noises decay at a rate of $0.64$. The relationships between maximum deviations $\max_{i \in V} \|p^i_k - \bar{p}\|_1$ and numbers of iterations $t$ are shown in Fig. 1(c). It is observed that our protocol converges faster than SCDA to the exact average. The main reason is that we do not continuously add noises to local variables all along the interactions. Hence, the possible negative effects of noises on the convergence rates are mitigated. Also, the deviation of the differentially private consensus protocol does not converge to zero. This phenomenon reflects the fundamental trade-off between privacy and accuracy in this class of protocols.

The robustness of the consensus-based iterations is shown in Fig. 1(d). We consider cases where the aforementioned time-varying links between agents suffer from different rates of failure, which results from packet drops or delays exceeding certain thresholds. It is observed that the iterations still converge in these cases, thus ensuring the solution accuracy of the proposed algorithm. Nonetheless, the convergence rates tend to be slower as the link failure rates increase.

VI. RELATED WORK

There have been extensive researches on designing efficient distributed optimization algorithms, e.g., primal methods [5], [6], [13], [45] and dual-based methods [2], [7], [46]. The core idea of the primal methods is to combine consensus with gradient-based optimization algorithms, thus achieving consensual iterative convergence in the primal domain. Thanks to the development of gradient tracking [6], [13], [45], [47], which enables local agents to approximately track the gradients of the global objective function, the convergence rates of these distributed algorithms can nearly match that of the optimal centralized gradient-based algorithm [48]. The basic intuition of the dual-based methods is to express the consensus requirement as equality constraints, and then solve the dual problems of the equivalent reformulations or carry on primal-dual updates. These carefully constructed dual problems are decoupled, thus easily allowing for the distributed implementations of certain linearly convergent centralized optimization algorithms, e.g., ADMM [46]. For convex problems, distributed algorithms guarantee convergence to globally optimal points; for nonconvex problems, the convergence to stationary or locally optimal points is ensured [25], [47], [49]–[51].

The aforementioned work mainly centers on bridging the gap in terms of convergence behaviors between distributed and centralized optimization algorithms. To effectively deploy these distributed algorithms into applications, some specific issues need to be addressed. These issues include but are not limited to privacy preservation, time-varying and directed communication, asynchronous computations due to lack of coordination, transmission delays or packet drops.

Specifically, the privacy concern of distributed algorithms has received growing attention. Conventional approaches are based on the premise that exact local data is exchanged between agents. Nevertheless, if there exist adversaries that intentionally gather certain data necessary for estimation, the sensitive information of objective functions, constraints and local states can be disclosed [4]. To tackle this problem, a number of privacy-preserving consensus and distributed
optimization algorithms have been proposed. One typical approach based on the idea of message perturbation is to add random noises to the data transmitted within iterations. The perturbation of the critical data (e.g., states [17]–[19], [21], [22], gradients [4], [8] and functions [9]) limits its utility for yielding sensible estimations. Some work considers the use of uncorrelated Laplacian or Gaussian noises and develops various differentially private consensus [21], [52] and distributed optimization algorithms [4], [8], [9]. The differentially private mechanism equips these algorithms with strong privacy guarantees even against those adversaries owning arbitrarily much side information. Nonetheless, it also brings about the trade-off between privacy and accuracy [9], [21]. Other work thus turns to correlated noises and shows that exact average consensus [17]–[19], [22] or optimization [40] is reached. Another typical approach is to apply cryptographic techniques, e.g., homomorphic encryption. Related algorithms can be found in [53]–[55]. These cryptography-based methods are suitable if the requirements of trusted agents or shared keys/ secrets are satisfied, and the extra communication and computation burdens induced by encryption and decryption are acceptable.

In addition to the privacy concern, the robustness issues of distributed optimization have also been widely investigated. Time-varying and directed communication inhibits the efficient construction of doubly stochastic weight matrices, which are crucial for achieving convergence over undirected graphs. To overcome this challenge, push-sum-based algorithms [10], [25], [26] and push-pull-based algorithms [11], [12] are developed. The former combine the push-sum consensus protocol [24] with gradient-based methods and only require column stochastic weight matrices. The latter use one row stochastic and one column stochastic weight matrix to mix estimates of optimal solutions and trackers of average gradients, respectively. Algorithms that purely handle random transmission delays can be found in [3], [56], where the basic idea is to locally fuse the delayed information as soon as it arrives. To achieve asynchronous computations, gossip-type algorithms [12], [13] and those further allowing delays and packet drops [14], [15] have been developed.

Different from the aforementioned work, the proposed algorithm is based on the idea of using polynomial approximation and is equipped with effective mechanisms to meet diverse practical requirements concerning privacy and robustness. We show that the efficient distributed optimization of general nonconvex problems is achieved, and in the meantime the common issues of privacy-accuracy trade-off and step-size selections are avoided.

VII. CONCLUSION

In this paper, we proposed D-CPOA to solve a class of constrained distributed nonconvex optimization problems, considering the needs of privacy preservation and robustness to various network imperfections. We achieved exact convergence and effective preservation of the privacy of local objective functions by incorporating a new privacy-preserving mechanism for consensus-based iterations. The developed mechanism utilized the randomness in blockwise insertions of perturbed data and separate subtractions of added noise, and its privacy degree was explicitly characterized through $(\alpha, \beta)$-data-privacy. We ensured the robustness of the proposed algorithm by using the push-sum average consensus protocol as a basis for iterations, and discussed its extensions to help maintain the performance when diverse imperfections in network communication exist. We proved that the major benefits brought by the idea of using polynomial approximation were preserved, and the aforementioned demanding requirements were satisfied at the same time.

APPENDIX

A. Proof of Lemma 1

Proof. The proof consists of two steps. First, we prove that the limit value of $p_t^i \triangleq x_t^i / y_t^i (t \in \mathbb{N})$ is indeed $\bar{p}$, i.e.,

$$\lim_{t \to \infty} p_t^i = \bar{p}. \quad (26)$$

Then, we demonstrate that the meet of the stopping criterion (16) is a sufficient condition for (21).

- Step 1: Proof of the Limit Value

We consider the $k$-th component of the involved local variables, $\forall k = 1, \ldots, m$. Let

$$x^t \triangleq [x_1^t(k), \ldots, x_N^t(k)]^T, \quad \theta \triangleq [\theta_1(k), \ldots, \theta_N(k)]^T,$$

$$p^0 \triangleq [p_1^0(k), \ldots, p_N^0(k)]^T, \quad y^t \triangleq [y_1^t, \ldots, y_N^t]^T.$$

Note that if the $k$-th components of some $x^t_j, \theta_j$ and $p^0_j (j \in \mathcal{V})$ are null, they are regarded as 0 in the expressions. We investigate the effects of insertions (11) and subtractions (14) on the accuracy of the consensus-based updates in Algorithm 1 as follows.

We first consider the effect of insertions that happened in the first $K_1$ iterations. Let $t_k$ be the number of the iteration where agent $i$ inserts the perturbed state $\tilde{p}_i^0(k)$. Since $A^{\bar{r}}_k$ is column stochastic, from (11) and (12), we have

$$1^T x^{t+1} = 1^T A^{\bar{r}}_k x^{t} = 1^T x^{t} + 1^T \tilde{p}_i^0(k),$$

which means that the sum of the components of $x^t$ increases by $\tilde{p}_i^0(k)$. At the end of the $K_1$-th iteration, all the agents have inserted their perturbed initial states. Hence,

$$1^T x^{K_1} = 1^T x^{0} + \sum_{i \in \mathcal{V}} \tilde{p}_i^0(k) = \sum_{i \in \mathcal{V}} \tilde{p}_i^0(k) = 1^T (p^0 + \theta).$$

Then, we focus on the effect of subtractions happened between time $K_1 + 1$ and time $K_2$. Suppose that agent $i$ performs its first action of subtractions at the $t_1$-th iteration. From the column stochasticity of $A^t (t \in \mathbb{N})$ and (3), it is not difficult to obtain that

$$1^T x^{t_1} = 1^T A^{t_1-1} x^{t_1-1} = 1^T x^{t_1-1} = \ldots = 1^T x^{K_1}.$$

At the $t_1$-th iteration, we have

$$1^T x^{t_1+1} = 1^T x^{t_1} - \delta_i(k) = 1^T x^{K_1} - \delta_i(k),$$

which implies that the sum of the components of $x^t$ decreases by $\delta_i(k) = \theta_i(k)/L$. At the end of the $K_2$-th iteration, every
agent has completed its $L$ rounds of actions of subtracting the noises. It follows that
\[
1^\top x^{K_2} = 1^\top x^{K_1} - 1^\top \theta = 1^\top p^0.
\]
Since $y^t_i$ is constantly updated by (3), we have
\[
1^\top y^{K_2} = 1^\top A^{K_2-1} y^{K_2-1} = \ldots = 1^\top y^0.
\]
Later on, agents continue to update $x^t_i$ and $y^t_i$ by (3). Based on the convergence of the push-sum consensus protocol, we conclude that the exact average can still be achieved, i.e.,
\[
\lim_{t \to \infty} p^t_i = \lim_{t \to \infty} x^t_i = \frac{1^\top y^{K_2}}{1^\top y^0} = \frac{1^\top p^0}{1^\top y^0} = \frac{\sum_{j=1}^N p^0_j(k)}{N} = p(k).
\]
Note that this result holds for any $k = 1, \ldots, m$. Therefore, the limit value of $p^t_i$ is $\bar{p}$, i.e., (26) holds.

**Step 2: Proof of the Sufficiency**

Next, we verify the effectiveness of the stopping criterion (16). Note that $p^t_i = x^t_i/y^t_i$, $\forall t \in \mathbb{N}$. The push-sum-consensus-based update of $x^t_i$ in (3) can be transformed to
\[
p^{t+1}_i = \frac{\sum_{j=1}^N w^t_{ij} y^t_j}{\sum_{j=1}^N w^t_{ij}}, \quad \text{where} \quad w^t_{ij} = \frac{a^t_{ij} y^t_j}{y^t_{i+1}}.
\]
It follows from (3) and (4) that $W^t = (w^t_{ij})_{N \times N}$ is row stochastic, i.e., $\sum_{j=1}^N w^t_{ij} = 1$, $w^t_{ij} \in [0, 1]$, $\forall i, j = 1, \ldots, N$. Hence, we have
\[
p^{t+1}_i(k) = \sum_{j=1}^N w^t_{ij} p^t_j(k) \leq \max_{j \in V} \max_{j \in V} p^t_j(k) = \max_{j \in V} p^t_j(k), \quad \forall k = 1, \ldots, m, \forall t \in \mathbb{V}.
\]
Let $M^t(k) \triangleq \max_{i \in V} p^t_i(k)$, $m^t(k) \triangleq \min_{i \in V} p^t_i(k)$. It follows that
\[
M^{t+1}(k) \leq M^t(k), \quad m^{t+1}(k) \geq m^t(k).
\]
It has been proven that $\lim_{t \to \infty} p^t_i(k) = \bar{p}(k), \forall i \in V$. Hence,
\[
\lim_{t \to \infty} M^t(k) = \bar{p}(k), \quad \lim_{t \to \infty} m^t(k) = \bar{p}(k).
\]
Since the sequences of $(M^t(k))_{t \in \mathbb{N}}$ and $(m^t(k))_{t \in \mathbb{N}}$ are non-increasing and non-decreasing, respectively, we have
\[
m^t(k) \leq \bar{p}(k) \leq M^t(k), \quad \forall t \in \mathbb{N}.
\]
Note that the max/min consensus protocols converge in $U$ iterations. When agents terminate at time $K$, we have
\[
r^K_i(k) - s^K_i(k) = M^K(k) - m^K(k),
\]
where $K^t \triangleq K - U$. The meet of (16) implies that
\[
|p^K_i(k) - \bar{p}(k)| \leq M^K(k) - m^K(k) \leq r^K_i(k) - s^K_i(k) \leq \delta, \quad \forall i, k.
\]

**B. Proof of Theorem 2**

**Proof.** The proof is similar to that of Theorem 4 in [16]. We provide a sketch of the main steps here. The key idea is to prove the closeness between $p^K_i(x)$ and $f(x)$ on the entire $X = [a, b]$. Then, their optimal values are also close enough (see [16, Lemma 3]). Note that $p^K_i(x)$ and $p(x)$ are in the forms of (5) with their coefficients $\{c_j\}$ and $\{c'_j\}$ stored in $p^K_i$ and $\bar{p}$, respectively. It follows from (21) that
\[
|p^K_i(x) - p(x)| = \left| \sum_{j=0}^m (c_j - c'_j)T_j \left( \frac{2x - (a+b)}{b-a} \right) \right|
\]
\[
\leq \sum_{j=0}^m c_j - c'_j \cdot 1 \leq \sum_{j=0}^m \|p^K_i - \bar{p}\|_\infty
\]
\[
\leq \delta(m + 1) = \epsilon, \quad \forall x \in [a, b],
\]
where the first inequality is based on $|T_j(s)| \leq 1, \forall s \in [-1, 1]$. Note that $\bar{p}$ is the average of all $p^K_i$. Hence, $p(x)$ is also the average of all $p_i(x)$. Based on (7), we have
\[
|p(x) - f(x)| = \left| \frac{1}{N} \sum_{i=1}^N (p_{i}(m^*) - f_i(x)) \right|
\]
\[
\leq \frac{1}{N} \sum_{i=1}^N \left| (m^*) - f_i(x) \right| \leq \frac{1}{N} N\epsilon = \epsilon, \quad \forall x \in [a, b].
\]
Given that $\epsilon_1 = \epsilon_2 = \epsilon/3$, we have
\[
|p^K_i(x) - f(x)| \leq |p^K_i(x) - \bar{p}(x)| + |\bar{p}(x) - f(x)|
\]
\[
\leq \epsilon_1 + \epsilon_2 = \frac{2}{3} \epsilon, \quad \forall x \in [a, b].
\]
Let $p^*$ be the optimal value of $p^K_i(x)$ on $X = [a, b]$. It follows from [16, Lemma 3] that
\[
|p^* - f^*| \leq \frac{2}{3} \epsilon.
\]
Note that $p^* \leq f^* \leq p^* + \epsilon_3 = p^* + \frac{\epsilon}{3}$. Hence,
\[
f^* - \frac{2}{3} \epsilon \leq p^* \leq f^* \leq p^* + \frac{\epsilon}{3} \leq f^* + \epsilon,
\]
which leads to $|f^* - f^*| \leq \epsilon$.

We then characterize the distance between $x^*_p$ and $x^*_f$. Without loss of generality, we consider the case where $x^*_p$ is the single globally optimal point of problem (1). If there are multiple globally optimal points, we can perform a similar analysis by investigating the distance of $x^*_p$ to the set of all these globally optimal points. We consider a rather small solution accuracy $\epsilon$, i.e., a quite strict requirement on the accuracy of the obtained solution. It follows from (27) that
\[
f(x^*_p) \leq p^K_i(x^*_p) + \frac{2}{3} \epsilon = p^K_i(x^*_f) \leq f(x^*_f) + \frac{2}{3} \epsilon,
\]
which implies that
\[
f(x^*_p) \leq f(x^*_f) + \frac{4}{3} \epsilon.
\]
Hence, $x^*_p$ falls in the following sublevel set of $f(x)$
\[
S = \{x \in X | f(x) \leq f(x^*_f) + \frac{4}{3} \epsilon \}.
\]
Therefore, we have
\[
|x^*_p - x^*_f| \leq \text{diam}(S).
\]
C. Proof of Theorem 3

Proof. We first consider the estimation of \( p_i^0(k) \), where \( k = 1, \ldots, m_i + 1 \). Suppose that at the \( t_k \)-th iteration, agent \( i \) inserts the perturbed state \( p_i^0(k) \) by (11). Note that the estimation \( \hat{p}_i(k) \) of \( p_i^0(k) \) can be calculated at three types of time, i.e., before \( t_k \), at \( t_k \), and after \( t_k \). We discuss each of these cases in detail as follows.

• Case 1: At time \( t < t_k \), \( p_i^0(k) \) has not been inserted yet. What the adversaries have collected are either null values or combinations of the perturbed states of agent \( i \)'s neighbors. Since there is not any available information on \( p_i^0(k) \) that serves as a basis for the estimation, by (22), we have

\[
\Pr\{||\hat{p}_i(k) - p_i^0(k)|| \leq \alpha_k | I_i^t \} \leq \gamma.
\]

• Case 2: At time \( t = t_k \), \( p_i^0(k) \) is inserted. By Assumption 4, the probability that the adversaries acquire the full knowledge of \( I_i^m t_k \) is \( p \). If this is the case, based on (11) and (12), they can easily calculate \( \hat{p}_i(k) \) by

\[
\hat{p}_i(k) = p_i^0(k) + \theta_i(k).
\]

Hence, after an estimation \( \hat{\theta}_i(k) \) of \( \theta_i(k) \) is obtained, \( \hat{p}_i(k) \) is calculated by

\[
\hat{p}_i(k) = \hat{p}_i^0(k) - \hat{\theta}_i(k).
\]

Therefore, we have

\[
\Pr\{||\hat{p}_i(k) - p_i^0(k)|| \leq \alpha_k | I_i^t \} = \Pr\{||\hat{\theta}_i(k) - \theta_i(k)|| \leq \alpha_k | I_i^t \},
\]

\[
= \Pr\{\theta_i(k) \in [\hat{\theta}_i(k) - \alpha_k, \hat{\theta}_i(k) + \alpha_k] | I_i^t \}
\]

\[
= \int_{\hat{\theta}_i(k) - \alpha_k}^{\hat{\theta}_i(k) + \alpha_k} f_{\theta_i}(y) dy
\]

\[
\leq \max_{\nu \in \Theta} \int_{\nu - \alpha_k}^{\nu + \alpha_k} f_{\theta_i}(y) dy,
\]

where \( \hat{\theta}_i(k) \in \Theta \). However, if the adversaries can only access part of \( I_i^m t_k \), they are unable to calculate \( x_i^t(k) \) by (12) and then recover \( \hat{p}_i^0(k) \) by (28). Note that

\[
x_i^{t+}(k) = x_i^t(k) + p_i^0(k) = x_i^t(k) + \theta_i(k) + p_i^0(k).
\]

Hence, in this case, they need to obtain an estimation \( \hat{\theta}_i(k) \) of \( x_i^t(k) \) first, and then calculate \( \hat{p}_i(k) \) by

\[
\hat{p}_i(k) = x_i^{t+}(k) - \hat{\theta}_i(k).
\]

According to (12), \( x_i^t(k) \) is a linear combination of the states \( x_j^{(t_k-1)} \) for \( j \in N_i^m t_k \). These states are dependent on some \( p_i^0(k) \) and thus also dependent on some \( \theta_i(k) \), where \( l \in \gamma \). Note that the adversaries only have partial knowledge of \( I_i^m t_k \) and know part of these states. Hence, there exist certain independent random variables, i.e., \( \theta_i(k) \), of which the adversaries do not own any prior or relevant knowledge. As a result, by (22), it is hard to estimate \( x_i^{t}(k) \) with high precision. It follows that

\[
\Pr\{||\hat{p}_i(k) - p_i^0(k)|| \leq \alpha_k | I_i^t \} = \Pr\{||\hat{\theta}_i(k) - (x_i^t(k) + \theta_i(k))|| \leq \alpha_k | I_i^t \}
\]

\[
\leq \Pr\{\theta_i(k) - x_i^{t}(k) \in [\theta_i(k) - \alpha_k, \theta_i(k) + \alpha_k] | I_i^t, \theta_i(k) \} \leq \gamma,
\]

(30)

Combining (29) and (30) together, we have

\[
\Pr\{||\hat{p}_i(k) - p_i^0(k)|| \leq \alpha_k | I_i^t \} \leq p \max_{\nu \in \Theta} \int_{\nu - \alpha_k}^{\nu + \alpha_k} f_{\theta_i}(y) dy + \gamma
\]

(31)

\[
\equiv h_i(\alpha_k).
\]

• Case 3: At time \( t > t_k \), the adversaries can estimate \( p_i^0(k) \) either by the same rule that is adopted at time \( t = t_k \), or by the new rule based on the new information. In the former case, we still obtain (31). We now discuss the latter case in detail. We first consider the time \( t = t_k + 1 \). Note that

\[
x_i^{(t_k+1)}(k) = x_i^{t+1}(k)
\]

\[
= x_i^{t+}(k) + \frac{1}{a_{ii}} \left( \sum_{j \in N_i^m t_k \setminus \{i\}} a_{ij} x_j^{t+}(k) - \tau_{i,t_k+1}(k) \right)
\]

\[
= p_i^0(k) + \theta_i(k) + x_i^{t+}(k)
\]

\[
+ \frac{1}{a_{ii}} \left( \sum_{j \in N_i^m t_k \setminus \{i\}} a_{ij} x_j^{t+}(k) - \tau_{i,t_k+1}(k) \right)
\]

\[
= p_i^0(k) + \theta_i(k) + \theta_i(k),
\]

(32)

where \( \tau_{i,t_k}(k) = \zeta_i(k) \) if noises are subtracted at time \( t \), and \( \tau_{i,t_k}(k) = 0 \) otherwise. If the full knowledge of \( I_i^m t_k \) is available, the adversaries can not only collect all the \( x_j^t(k) \) for \( j \in N_i^m t_k \), but also accurately infer \( \tau_{i,t_k+1}(k) \) by

\[
\tau_{i,t_k+1}(k) = \sum_{j \in N_i^m t_k} a_{ij} x_j^{t+}(k) - x_i^{(t_k+1)}(k).
\]

Hence, \( \theta_i^0(k) \) is a deterministic constant. In this case, by using (32), we still have

\[
\Pr\{||\hat{p}_i(k) - p_i^0(k)|| \leq \alpha_k | I_i^{t+1} \} = \Pr\{||\hat{\theta}_i(k) - \theta_i(k)|| \leq \alpha_k | I_i^{t+1} \}.
\]

Next, we analyze the disclosure probability of \( \theta_i(k) \) given \( I_i^{t+1} \). The newly available information, i.e., the subtracted noise \( \zeta_i(k) \), allows for another means of inferring \( \theta_i(k) \). We now show that the resulting disclosure probability is rather small when \( L \) is drawn from an unknown distribution. Note that \( \zeta_i(k) = \theta_i(k) / L > \alpha_k \). Hence,

\[
\Pr\{||\hat{p}_i(k) - p_i^0(k)|| \leq \alpha_k | I_i^{t+1} \} = \Pr\{||\hat{\theta}_i(k) - \theta_i(k)|| \leq \alpha_k | I_i^{t+1} \}
\]

\[
= \Pr\{|\hat{\theta}_i(k) - \theta_i(k)| \leq \alpha_k | I_i^{t+1} \}
\]

\[
= \Pr\{|\hat{L} - L| \cdot \zeta_i(k) \leq \alpha_k | I_i^{t+1} \}
\]

\[
= \Pr\{L \leq L | \zeta_i(k)\}
\]

\[
\leq \gamma,
\]
where $\hat{L}$ is an estimation of $L$, and the last inequality follows from (22). Thus, the disclosure probability will not exceed the upper bound in (29), i.e.,
\[
\Pr\{\hat{p}_i(k) - p_i^0(k) \leq \alpha_k|I_k^{i+1}\} = \Pr\{\hat{\theta}_i(k) - \theta_i(k) \leq \alpha_k|I_k^{i+1}\} \\
\leq \max_{\nu \in \Theta} \int_{\nu - \alpha_k}^{\nu + \alpha_k} f_{\theta_i}(y)dy + \gamma.
\]
(34)

If the full knowledge of $I_{k}^{i,m}$ is unavailable, then $\theta_i'(k)$ contains those independent random variables whose relevant information is unknown to the adversaries. Specifically, if $t_k + 1 \leq K_2$, then those variables refer to certain added noises $\hat{\theta}_i(k)$ that are included in $x_i^{k+1}(k)$, where $l \in \mathcal{V}$. Otherwise, those variables refer to certain subtracted noises $\hat{\gamma}_i(k)$ for some $l \in \mathcal{V}$. Thus, it follows from (22) that
\[
\Pr\{\hat{p}_i(k) - p_i^0(k) \leq \alpha_k|I_k^{i+1}\} \leq \gamma.
\]
Combining (33) and (34), we have
\[
\Pr\{|\hat{p}_i(k) - p_i^0(k)| \leq \alpha_k|I_k^{i+1}\} = \max_{\nu \in \Theta} \int_{\nu - \alpha_k}^{\nu + \alpha_k} f_{\theta_i}(y)dy + \gamma
\]
(35)

A similar analysis can be performed for other arbitrary $t \geq t_k + 1, t \in \mathbb{N}$. However, for $t \geq K_2$, there exists an extreme case where the adversaries successfully obtain the full knowledge of $I_{k}^{i,m,t}$ at time $t = t_k + 1$ and also from time $t = K_1 + 1$ to time $t = K_2$. In this case, they can not only calculate $p_i^0(k)$ by (28), but also acquire $\tau_{i,t}(k)$ and perfectly infer $\theta_i(k)$ by
\[
\theta_i(k) = \sum_{t=K_1+1}^{K_2} \tau_{i,t}(k).
\]

Hence, the exact value of $p_i^0(k)$ can be obtained, and
\[
\Pr\{|\hat{p}_i(k) - p_i^0(k)| \leq \alpha_k|I_k^{K_2}\} = \Pr\{|\hat{\theta}_i(k) - \theta_i(k)| \leq \alpha_k|I_k^{K_2}\} = 1.
\]

The probability that such a case happens is $p_{K_2-K_1+1}$. Thus, for any $k = 1, \ldots, m_i + 1$ and $t \in \mathbb{N}$, we have
\[
\Pr\{|\hat{p}_i(k) - p_i^0(k)| \leq \alpha_k|I_k^{i+1}\} \leq \beta_k,
\]
(36)

where
\[
\beta_k = (1 - p_{K_2-K_1+1})h_i(\alpha_k) + p_{K_2-K_1+1}.
\]
(37)

Since $h_i(\alpha_k) \leq p + \gamma < 1$, $\beta_k$ is larger than the RHS of (31).

Then, we consider the inference on whether $p_i^0(k)$ is null for $k = m_i + 1, \ldots, m_i + 1$, i.e., whether $p_i^0$ is an $(m_i + 1)$-dimensional vector. Note that there is no action of insertions or subtractions corresponding to the aforementioned components. Hence, the adversaries will not find any inconsistency between $x_i^{(t)}(k)$ and $\sum_{j \in \mathcal{V}} a_{i,j}x_j^{(t)}(k)$, where $t \in \mathcal{S}_i$. Let this event be denoted by $A$. Once it occurs, the adversaries need to decide between the following two hypotheses
\[
\mathcal{H}_0 : p_i^0(k) = \text{null}, \quad \mathcal{H}_1 : p_i^0(k) \text{ is a nonzero number.}
\]

Based on the algorithmic design, we have
\[
\Pr\{A|\mathcal{H}_0\} = 1, \quad \Pr\{A|\mathcal{H}_1\} = (1 - p)^{L+1}.
\]

It follows from the maximum likelihood rule that the adversaries will always choose $\mathcal{H}_0$ when $A$ occurs. The probability that they successfully decide that $p_i^0(k)$ is null for $k = m_1 + 2, \ldots, m_1 + 1$ equals to
\[
\Pr\{m_1 + 1 \leq k - 1\} = F_{m_1|I_k}^{(k-2)}, \quad (38)
\]
Combining (36) and (38), we have
\[
\Pr\{|\hat{p}_i(k) - p_i^0(k)| \leq \alpha_k|I_k^{i+1}\} = \prod_{k=1}^{m_1+1} \Pr\{m_1 + 1 \leq k - 1\} \leq \beta_k \cdot \prod_{k=m_1+2}^{m_1+1} F_{m_1|I_k}^{(k-2)} = \beta. \quad \square
\]

REFERENCES

[1] Z. He, J. He, C. Chen, and X. Guan, “Constrained distributed nonconvex optimization over time-varying directed graphs,” in Proc. 59th IEEE Conf. Decis. Control, 2020, pp. 378–383.

[2] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, “Optimal algorithms for smooth and strongly convex distributed optimization in networks,” in Proc. 34th Int. Conf. Mach. Learning, 2017, pp. 3027–3036.

[3] T. Yang, J. Lu, D. Wu, J. Wu, G. Shi, Z. Meng, and K. H. Johansson, “A distributed algorithm for economic dispatch over time-varying directed networks with delays,” IEEE Trans. Ind. Electron., vol. 64, no. 6, pp. 5095–5106, 2016.

[4] S. Han, U. Topcu, and G. J. Pappas, “Differentially private distributed constrained optimization,” IEEE Trans. Autom. Control, vol. 62, no. 1, pp. 50–64, 2017.

[5] A. Nedić and A. Ozdaglar, “Distributed subgradient methods for multiagent optimization,” IEEE Trans. Autom. Control, vol. 54, no. 11, pp. 48–61, 2009.

[6] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: An exact first-order algorithm for decentralized consensus optimization,” SIAM J. Optim., vol. 20, no. 2, pp. 944–966, 2015.

[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, 2011.

[8] M. T. Hale and M. Egerstedt, “Cloud-enabled differentially private multiagent optimization with constraints,” IEEE Control Syst., vol. 5, no. 4, pp. 1693–1706, 2018.

[9] E. Nozari, P. Tallapragada, and J. Cortés, “Differentially private distributed convex optimization via functional perturbation,” IEEE Trans. Control Netw. Syst., vol. 5, no. 1, pp. 395–408, 2018.

[10] A. Nedíć, A. Olshovsky, and W. Shi, “Achieving geometric convergence for distributed optimization over time-varying graphs,” SIAM J. Optim., vol. 27, no. 4, pp. 2597–2633, 2017.

[11] R. Xin and U. A. Khan, “A linear algorithm for optimization over directed graphs with geometric convergence,” IEEE Contr. Syst. Lett., vol. 2, no. 3, pp. 315–320, 2018.

[12] S. Pu, W. Shi, J. Xu, and A. Nedíć, “Push-pull gradient methods for distributed optimization in networks,” IEEE Trans. Autom. Control, vol. 66, no. 1, pp. 1–16, 2021.

[13] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, “Convergence of asynchronous distributed gradient methods over stochastic networks,” IEEE Trans. Autom. Control, vol. 63, no. 2, pp. 434–448, 2018.

[14] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, “Decentralized consensus optimization with asynchrony and delays,” IEEE Trans. Signal Inf. Process. Netw., vol. 4, no. 2, pp. 293–307, 2018.
