Guaranteed Privacy of Distributed Nonconvex Optimization via Mixed-Monotone Functional Perturbations

Mohammad Khajenejad, *Member, IEEE* and Sonia Martínez, *Fellow, IEEE*

***Abstract*— In this paper, we introduce a new notion of guaranteed privacy for distributed nonconvex optimization algorithms. In particular, leveraging mixed-monotone inclusion functions, we propose a privacy-preserving mechanism which is based on deterministic, but unknown affine perturbations of the local objective functions. The design requires a robust optimization method to characterize the best accuracy that can be achieved by an optimal perturbation. This is used to guide the refinement of a guaranteed-private perturbation mechanism that can achieve a quantifiable accuracy via a theoretical upper bound that is independent of the chosen optimization algorithm. Finally, simulation results illustrate the accuracy-privacy trade-off and that our approach outperforms a benchmark differentially private distributed optimization algorithm in the literature.**

***Index Terms*— Distributed nonconvex optimization, Functional perturbations, Guaranteed privacy**

I. INTRODUCTION

DATA privacy and protection have become a critical concern in the management of cyber-physical systems (CPS) and their public trustworthiness. In such applications, malicious agents can expand their attack surface by extracting valuable information from the many physical, control, and communication components of the system, inflicting damage on the CPS and its users. Hence, a great effort is being devoted to design robust, data-secure control strategies for these systems [1]. Motivated by this, we aim to investigate an alternative design of privacy-preserving mechanisms, which can make the quantification of privacy, as well as the associated performance loss, both tractable and reasonable.

Literature Review. Among the many approaches to data security, one can distinguish privacy-aware methods that protect sensitive data from worst-case data breaches by adding random perturbations to it. However, this high data-resiliency comes at the cost of high performance loss, which either can be hard to quantify in practice, or is theoretically bounded by indices that are too large to be useful. A main approach in the literature to characterize this trade-off, while preserving privacy, is that of *differential privacy* [2].

This method was originally proposed for the protection of databases of individual records subject to public queries. In particular, a system processing sensitive inputs is made differentially private by randomizing its answers in such a way that the distribution over published outputs is not too sensitive to the data provided by any single participant. This notion has been extended to several areas in machine learning and regression [3]–[5], control (estimation, verification) [6], [7], multi-agent systems (consensus, message passing) [8]–[10], and optimization and games [11]–[13]. In particular, a notable privacy-preserving mechanism design approach in the literature is based on the idea of message perturbation, i.e., modifying an original non-private algorithm by having agents perturb the messages/outputs to their neighbors or a central coordinator with Laplace or Gaussian noise [8]–[10]. This approach benefits from working with the original objective functions, however, it suffers from a steady-state accuracy error, since for fixed design parameters, the algorithm's output does not correspond the true optimizer in the absence of noise [14]. To overcome this shortcoming, encryption-based privacy-preserving algorithms for some classes of functions were introduced in; e.g., [15], [16].

Another notable approach to privacy relies on functional perturbation, with the idea of having agent(s) independently perturb their objective function(s) in a differentially private way and then participating in a centralized (or distributed) algorithm [3], [4], [14]. The work in [3] proposed a differentially private classifier by perturbing the objective function with a linear finite-dimensional function. However, only the privacy of the underlying finite-dimensional parameter set—and not the entire objective functions—is preserved. A sensitivity analysis-based differentially private algorithm was designed in [4] by perturbing the Taylor expansion of the cost function, where, unfortunately, the functional space had to be restricted to the space of quadratic functions. Similar perturbations were proposed in [5] but without ensuring the smoothness and convexity of the perturbed function. In addition, none of [3]–[5] provided a systematic way to study the effect of added noise on the global optimizer. To address this issue, [14] suggested specific functional perturbations such that the difference between the probabilities of events corresponding to any pair of data sets is bounded by a function of the distance between the data sets, at the expense of a trade-off between the privacy and the accuracy of the mechanism. Further, the works in [7], [17] provided theoretically proven numerical methods to quantify differential privacy in high probability in estimation and verification, respectively. However, differential

Manuscript received Sep. 16, 2022; revised Nov. 21, 2022; accepted Dec. 6, 2022. This work is partially supported by ONR and AFOSR grants ONR-N00014-19-1-2471 and FA-9550-19-1-0235.

M. Khajenejad and S. Martínez are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, USA (email: mkhajenejad, soniamd@ucsd.edu)

privacy requires the slight change of the *statistics* of the output of the perturbed function or message if the objective function or the message sent from one agent changes. This satisfies privacy only in a probabilistic, and not a guaranteed sense. To bridge this gap, we aim to investigate an alternative deterministic approach to privacy-preserving mechanism design for distributed optimization, and also to relax the convexity assumption needed in most of the existing works.

Contributions. We start by introducing a novel notion of guaranteed privacy. This notion applies to deterministic, but unknown, functional perturbations of an optimization problem, and characterizes privacy in terms of how close the ranges of two sets of functions in a vicinity are. This notion of privacy is stronger than that of differential privacy in the sense that it guarantees that the changes of the true function ranges are small through deterministic, but unknown, functional perturbations. In contrast, functions are perturbed stochastically by a differential privacy approach. By exploiting the differentiability and local Lipschitzness of the objective functions, we propose a novel perturbation mechanism that relies on the mixed-monotone inclusion functions of the problem objectives. We then characterize the *best accuracy* that can be achieved by an optimal perturbation, and use this to guide the refinement of a guaranteed-private perturbation mechanism that can achieve a quantifiable accuracy via a theoretical upper bound. The design requires a robust optimization approach, and restricts privacy to functions in a given vicinity. Simulations are used to illustrate the level of the tightness of the accuracy upper bounds, as well as the accuracy-privacy compromise.

II. Preliminaries

In this section, we introduce basic notation, as well as preliminary concepts and results used in the sequel.

Notation. $\mathbb{R}^n, \mathbb{R}^{n\times p}, \mathbb{D}_n, \mathbb{R}^n_{\geq 0}$, and $\mathbb{R}^n_{>0}$ denote the n-dimensional Euclidean space, and the sets of n by p matrices, diagonal n by n matrices, and nonnegative and positive vectors in $\mathbb{R}^n$, respectively. Also, $\mathbf{0}_{n\times p}, I_n$ and $\mathbf{0}_n$ denote the zero matrix in $\mathbb{R}^{n\times p}$, the identity matrix in $\mathbb{R}^{n\times n}$, and the zero vector in $\mathbb{R}^n$, respectively. Further, for $D \subseteq \mathbb{R}^n$, $L_2(D)$ denotes the set of square-integrable measurable functions over D. Given $M \in \mathbb{R}^{n\times p}$, $M^\top$ represents its transpose, M_{ij} denotes M's entry in the i^{th} row and the j^{th} column, $M^\oplus \triangleq \max(M, \mathbf{0}_{n\times p})$, $M^\ominus = M^\oplus - M$ and $|M| \triangleq M^\oplus + M^\ominus$. Finally, for $a, b \in \mathbb{R}^n, a \leq b$ means $a_i \leq b_i, \forall i \in \{1, \ldots, n\}$.

Definition 1 (Hyper-intervals): A (hyper-)interval $\mathcal{I} \triangleq [\underline{z}, \overline{z}] \subset \mathbb{R}^n$, or an n-dimensional interval, is the set of all real vectors $z \in \mathbb{R}^n$ that satisfy $\underline{z} \leq z \leq \overline{z}$. Moreover, we call $\text{diam}(\mathcal{I}) \triangleq \|\overline{z} - \underline{z}\|_\infty \triangleq \max_{i\in\{1,\cdots,n\}} |\overline{z}_i - \underline{z}_i|$ the *diameter* or *interval width* of $\mathcal{I}$. Finally, $\mathbb{IR}^n$ denotes the space of all n-dimensional intervals, i.e., *interval vectors*.

Definition 2 ($\mathcal{V}$-Adjacent Sets of Functions): Given any normed vector space $(\mathcal{V}; \|\cdot\|_\mathcal{V})$ with $\mathcal{V} \subseteq L_2(D)$, two sets of functions $F \triangleq \{f_1, \ldots, f_n\}, F' \triangleq \{f'_1, \ldots, f'_n\} \subset L_2(D)$ are called $\mathcal{V}$*-adjacent* if there exists $i_0 \in \{1, \ldots, n\}$ such that

$$f_{i_0} - f'_{i_0} \in \mathcal{V}, \quad \text{and} \quad f_i = f'_i, \text{ for all other } i \neq i_0 \,.$$

III. Problem Formulation

Consider a group of N agents communicating over a network. Each agent $i \in \{1, \ldots, N\}$ has a local objective function $\hat{f}_i : D \to \mathbb{R}$, where $D \subset \mathbb{R}^n$ has a nonempty interior. Consider the problem

$$\min_{x\in\mathcal{X}_0} \hat{f}(x) \triangleq \textstyle\sum_{i=1}^N \hat{f}_i(x) \ \text{ s.t. } G(x) \leq 0, H(x) = b, x \in \mathcal{X}_0, \quad (1)$$

where the mappings $G : D \to \mathbb{R}^m, H : D \to \mathbb{R}^s$ are convex, the vector $b \in \mathbb{R}^s$ are known to all agents and $\mathcal{X}_0 \triangleq [\underline{x}_0, \overline{x}_0]$ is assumed to be an interval in $\mathbb{R}^n$. By applying a *penalty* method [18, Ch.5], the problem in (1) is equivalent to

$$\min_{x\in\mathcal{X}_0} f(x) \triangleq \textstyle\sum_{i=1}^N f_i(x), \quad (2)$$

where, we assume the following:

Assumption 1 (Locally Lipschitz & Differentiable Objectives): Each f_i is differentiable, locally Lipschitz in its domain and only known to agent i. Moreover, upper and lower uniform bounds for its Jacobian matrix, $\overline{J}^f_i, \underline{J}^f_i \in \mathbb{R}^{1\times n}$ are only known to agent i. Moreover, the problem constraint set $\mathcal{X}_0$ is globally known to each agent[1].

Note that we do not restrict any $f_i, i \in \{1, \ldots, n\}$, to be convex, nor twice-continuously differentiable. The problem objective is to define a mechanism so that agents can solve (2) via a distributed, private algorithm.

It is known that even if f_i and/or its gradient may not be directly shared among agents, an adversary may be able to infer it by compounding side information with network communications. This problem has been addressed via the notion of differential privacy and the design of differentially-private algorithms, e.g., in [11]–[14], for convex objective functions (cf. Section I for a thorough review of the relevant work). Differential privacy applies a random, additive perturbation to either the algorithm input data or its output, so that the result becomes very close to that of the same process when applied to data in a vicinity. Here, our problem data refers to the problem objective functions, and the novel to-be-designed privacy mechanism consists of applying deterministic additive perturbations characterized by intervals—but unknown to the adversary. By doing so, the approximated range of the objective functions is also an interval, and privacy can be measured by how close these intervals are for functions in a vicinity. We formalize this via a mapping $\mathcal{M}$, as follows, and provide one such mapping in the next section.

Definition 3 (Guaranteed Privacy): Let $\mathcal{M} : L_2(D)^N \times \mathbb{IR}^n \to \mathbb{IR}^N$ be a deterministic interval-valued map from the function space $L_2(D)^N$ to the space of intervals in $\mathbb{R}^N$. Given $\mathcal{X} \triangleq [\underline{x}, \overline{x}] \in \mathbb{IR}^n$ and a "privacy gap" $\epsilon \in \mathbb{R}_{>0}$, the map $\mathcal{M}$ is ϵ*-guaranteed private* with respect to the vicinity $\mathcal{V}$, if for any two $\mathcal{V}$-adjacent sets of functions $F \triangleq \{f_i\}_{i=1}^N$ and $F' \triangleq \{f'_i\}_{i=1}^N$ that (at most) differ in their i_0^{th} element and any interval $\mathcal{I} \in \mathbb{IR}^N$ such that $\mathcal{M}(F, \mathcal{X}) \subseteq \mathcal{I}$, one has

$$\text{diam}(\mathcal{M}(F', \mathcal{X}) \cap \mathcal{I}) \leq e^{\epsilon\|f_{i_0} - f'_{i_0}\|_\mathcal{V}} \text{diam}(\mathcal{M}(F, \mathcal{X})). \quad (3)$$

[1]Here, the "hard" constraint set $\mathcal{X}_0$ is used to model commonly known constraints that originate from, e.g., reasonable bounds of the problem. We assume that other local constraints are treated in a "soft" manner, and have been accounted for as part of the local agents' objectives. Since local constraints are typically subject to privacy considerations, sharing them with others to construct a common constraint set would require an additional privacy-preserving algorithm. Hard constraints will be addressed in future work by considering local perturbations to hard constraint sets.

It is worth to reemphasize the difference between the notions of guaranteed privacy (introduced above) and differential privacy utilized in most of the work in the literature, e.g., [11], [12], [14]. When differential privacy is considered, the *statistics* of the output of $\mathcal{M}$, i.e., the *probability* of the value of $\mathcal{M}$ belonging to some set changes only relatively slightly if the objective function of one agent changes, and the change is in $\mathcal{V}$ (cf. [14, Definition III.1] for more details). On the other hand, when guaranteed privacy is considered, instead of introducing randomness to disguise the objective function, we implement a range perturbation of $\mathcal{V}$-adjacent functions as defined above, with the end goal of robustifying the optimization problem (2) in a controlled manner by an ϵ gap. This being said, note that our goal is to design a new mechanism (or mapping) $\mathcal{M}$ that preserves the ϵ-guaranteed privacy of the objective functions with respect to the problem solution, regardless of the distributed (nonconvex) optimization algorithm chosen to arrive at it. In this case, $\mathcal{M}$ can be interpreted as a preventive action on the set of local functions F, which will guarantee the privacy of *any* distributed nonconvex optimization algorithm applied to solve (2). So, our problem can be cast as follows:

Problem 1: Given the program in (2), design a mechanism $\mathcal{M}$ that maintains the privacy of any convergent distributed nonconvex optimization algorithm in the sense of Definition 3. In other words, $\mathcal{M}$ is an ϵ-guaranteed privacy-preserving map with some desired $\epsilon > 0$. Further, the mechanism's guarantee on accuracy improves as the level of privacy decreases.

IV. Guaranteed Privacy-Preserving Functional Perturbation

In this section, we introduce our proposed strategy to design a guaranteed privacy-preserving mechanism (or mapping) for distributed nonconvex optimization. The main idea is to perturb the true objective function by deterministic but unknown *linear additive perturbation functions* in a distributed manner, such that the privacy is preserved, regardless of the utilized optimization algorithm. Then, we show that the level of privacy can be estimated by computing the over-approximation of the range of the true and perturbed functions using mixed-monotone inclusion functions. For the sake of completeness, we start by briefly recapping the notions of inclusion and decomposition functions, as well as mixed-monotonicity that will be used throughout our main results.

Definition 4 (Inclusion Functions): [19, Chapter 2.4] Consider a function $g : \mathbb{R}^n \to \mathbb{R}^N$. The interval function $[g] : \mathbb{IR}^n \to \mathbb{IR}^N$ is an inclusion function for g, if

$$\forall \mathcal{X} \in \mathbb{IR}^n, g(\mathcal{X}) \subseteq [g](\mathcal{X}),$$

where $g(\mathcal{X})$ denotes the true range of g for the domain $\mathcal{X}$.

Proposition 1 (JSS Decomposition): [20, Corollary 2] Consider a function $g : \mathcal{X} \triangleq [\underline{x}, \overline{x}] \in \mathbb{IR}^n \to \mathbb{R}$ and suppose $\forall x \in \mathcal{X}, J^g(x) \in [\underline{J}^g, \overline{J}^g]$, where $J^g(x)$ is the gradient of g at x and $\underline{J}^g, \overline{J}^g$ are known row vectors in $\mathbb{R}^{1\times n}$. Then, g can be decomposed into the sum of an affine mapping, parameterized by a row vector $m \in \mathbb{R}^{1\times m} \in \mathbf{M}_g$, and a remainder mapping $h : \mathcal{X} \to \mathbb{R}$, as follows:

$$g(x) = h(x) + mx, \quad \forall x \in \mathcal{X}, \text{ where} \tag{4}$$

$$\mathbf{M}_g \triangleq \{m \in \mathbb{R}^{1\times m} | m_j = \overline{J}^g_j \text{ or } m_j = \underline{J}^g_j,\ 1 \le j \le n\}. \tag{5}$$

Further, h is a *Jacobian sign-stable* (JSS) [21] mapping in $\mathcal{X}$ by construction, i.e., its Jacobian vector entries have constant sign over $\mathcal{X}$. Therefore, for each $j \in \{1, \ldots, n\}$, either of the following hold: $J^h_j(x) \ge 0, \forall x \in \mathcal{X}$, or $J^h_j(x) \le 0, \forall x \in \mathcal{X}$, where $J^h(x)$ denotes the Jacobian vector of h at $x \in \mathcal{X}$.

Proposition 2 (Mixed-Monotone Inclusion Functions): [22, Proposition 4] Given the assumptions in Proposition 1,

$$[g](\mathcal{X}) = [h_d(\underline{x}, \overline{x}) + m^{\oplus}\underline{x} - m^{\ominus}\overline{x}, h_d(\overline{x}, \underline{x}) + m^{\oplus}\overline{x} - m^{\ominus}\underline{x}], \tag{6}$$

with $h_d(x_1, x_2) \triangleq h(Bx_1 + (I_n - B)x_2)$, for any ordered $x_1, x_2 \in \mathcal{X}$, i.e., $x_1 \le x_2$ or $x_2 \le x_1$, is an inclusion function for g. We refer to this inclusion as the *mixed-monotone inclusion function* of g. Further, $B \in \mathbb{D}_n$ is a binary diagonal matrix that identifies the vertex of the interval $[x_1, x_2]$ (or $[x_2, x_1]$) that minimizes (or maximizes) the JSS function h in the case that $x_1 \le x_2$ (or $x_2 \le x_1$), and can be computed as: $B = \text{diag}(\max(\text{sgn}(\overline{J}^g), \mathbf{0}_{1,n}))$. Finally, h_d is tight, i.e.,

$$\underline{h}_{\mathcal{X}} \triangleq h_d(\underline{x}, \overline{x}) = \min_{x\in\mathcal{X}} h(x),\ \ \overline{h}_{\mathcal{X}} \triangleq h_d(\overline{x}, \underline{x}) = \max_{x\in\mathcal{X}} h(x).$$

From now on, $[g](\mathcal{X})$ denotes the mixed-monotone inclusion function of g on $\mathcal{X}$, unless otherwise specified.

A. Guaranteed Privacy-Preserving Mechanism

We are ready to introduce a guaranteed privacy-preserving map (or mechanism) through the following theorem.

Theorem 1 (Privacy of Functional Perturbation): Let $F = \{f_i(x)\}_{i=1}^N = \{h_i(x) + m_i x\}_{i=1}^N$ be the JSS decompositions of the set of functions $f_i : \mathcal{X}_0 \triangleq [\underline{x}_0, \overline{x}_0] \in \mathbb{IR}^n \to \mathbb{R}, i \in \{1, \ldots, n\}$, based on Proposition 1. Suppose Assumption 1 holds, $\mathcal{X}_0$ is not a singleton, $\tilde{m}_i^\top \in \mathbb{R}^n$, and

$$\epsilon_i \triangleq \beta(f_i, \tilde{m}_i, \mathcal{X}_0, \delta_i) \triangleq (\frac{1}{\delta_i}) \min_{m\in\mathbf{M}_{f_i}} \ln\left(\frac{\Delta^{h_i}_{\mathcal{X}_0} + |\hat{m}_i|\Delta + 2\delta_i}{\Delta^{h_i}_{\mathcal{X}_0} + |\hat{m}_i|\Delta}\right), \tag{7}$$

where $\Delta \triangleq \overline{x}_0 - \underline{x}_0, \hat{m}_i \triangleq m_i + \tilde{m}_i,\ \Delta^{h_i}_{\mathcal{X}_0} \triangleq \overline{h}_i - \underline{h}_i$ and $\mathbf{M}_{f_i}, \overline{h}_i, \underline{h}_i$ are given in Propositions 1 and 2. Then, the mapping $\mathcal{M} : L_2(D)^N \times \mathcal{X}_0 \to \mathbb{IR}^N$ defined as

$$\begin{aligned} &\mathcal{M}(F, \mathcal{X}_0) = [\mathcal{G}](\mathcal{X}_0), [\mathcal{G}] \triangleq [[g_1]^\top, \ldots, [g_N]^\top]^\top, \\ &g_i(x) \triangleq f_i(x) + \tilde{m}_i x,\ \forall x \in \mathcal{X}_0, \forall i \in \{1, \ldots, n\} \end{aligned} \tag{8}$$

satisfies ϵ-guaranteed privacy where $\epsilon = \max_{i\in\{1,\ldots,n\}} \epsilon_i$, with respect to the vicinity:

$$\mathcal{V} \triangleq \{f'_i \in L_2(D) | \forall i \in \{1, \ldots, N\}, |f'_i(x) - f_i(x)| \le \delta_i, \forall x \in \mathcal{X}_0\}. \tag{9}$$

We call $\{\tilde{m}_i\}_{i=1}^N$ the "perturbation slopes" and f'_i a function in the "δ_i-vicinity" of f_i, throughout the paper.

From Theorem 1, the privacy gap $\epsilon_i = \beta(f_i, \tilde{m}_i, \mathcal{X}_0, \delta_i)$ is a decreasing function of δ_i, i.e., the smaller ϵ is, the harder it will be to distinguish the solution to problems with functions in a vicinity $\mathcal{V}$. Note that, this result shows that by making δ_i large, i.e., by allowing the distance between the perturbed and the true function become larger, we can make ϵ_i small and thus increase privacy. However, this will directly impact the distance between the optimizers of the corresponding problems, which results in a loss of the quality of the solution—in the sense of being close to the original one. This intuitively characterizes a trade-off between privacy and accuracy, which will be discussed later in Theorem 2.

Proof: With a slight abuse of notation, let $\mathcal{M}(f_i, \mathcal{X}_0)$ denote the i^{th} argument of the interval vector $\mathcal{M}(F, \mathcal{X}_0)$. It follows from (8) and Propositions 1 and 2 that

$$\begin{aligned}\mathcal{M}(f_i, \mathcal{X}_0)=[\underline{h}_i+m_i^{\oplus}\underline{x}_0-m_i^{\ominus}\overline{x}_0, \overline{h}_i+m_i^{\oplus}\overline{x}_0-m_i^{\ominus}\underline{x}_0]\Rightarrow \\ \text{diam}(\mathcal{M}(f_i, \mathcal{X}_0)) = \Delta_{\mathcal{X}_0}^{h_i} + |\hat{m}_i|\Delta.\end{aligned} \tag{10}$$

On the other hand, (9) implies that for any f_i' in the vicinity of f_i, $-\delta_i + f_i(x) \leq f_i'(x) \leq \delta_i + f_i(x), \forall x \in \mathcal{X}_0$. By adding the perturbation functions $\tilde{m}_i x$ and using the JSS decomposition of f_i for an arbitrary $m_i \in \mathbf{M}_{f_i}$, we obtain $-\delta_i+h_i(x)+m_i x+\tilde{m}_i x \leq f_i'(x)+\tilde{m}_i x\leq\delta_i+h_i(x)+m_i x+\tilde{m}_i x$, which implies:

$$\mathcal{M}(f_i', \mathcal{X}_0)\subseteq [-\delta_i+\underline{h}_i+\hat{m}_i^{\oplus}\underline{x}_0-\hat{m}_i^{\ominus}\overline{x}_0, \delta_i+\overline{h}_i+\hat{m}_i^{\oplus}\overline{x}_0-\hat{m}_i^{\ominus}\underline{x}_0].$$

This, together with the fact $\text{diam}(\mathcal{M}(f_i', \mathcal{X}_0) \cap \mathcal{I}) \leq \text{diam}(\mathcal{M}(f_i', \mathcal{X}_0))$ for any interval $\mathcal{I}$, results in

$$\text{diam}(\mathcal{M}(f_i', \mathcal{X}_0) \cap \mathcal{I}) \leq 2\delta_i + \Delta_{\mathcal{X}_0}^{h_i} + |\hat{m}_i|\Delta. \tag{11}$$

Finally, it follows from (7) , (10) and (11) that

$$\frac{\text{diam}(\mathcal{M}(f_i', \mathcal{X}_0) \cap \mathcal{I})}{\text{diam}(\mathcal{M}(f_i, \mathcal{X}_0))}\leq \min_{m\in\mathbf{M}_{f_i}} \frac{2\delta_i + \Delta_{\mathcal{X}_0}^{h_i} + |\hat{m}_i|\Delta}{\Delta_{\mathcal{X}_0}^{h_i} + |\hat{m}_i|\Delta}= e^{\alpha_i},$$

with $\alpha_i \triangleq \epsilon_i\delta_i$, which implies $\text{diam}(\mathcal{M}(f_i', \mathcal{X}_0) \cap \mathcal{I}) \leq e^{\epsilon_i\delta_i}\text{diam}(\mathcal{M}(f_i, \mathcal{X}_0))$. Taking the maximum of both sides on i returns the results in (3). ■

B. Tractable Computation of a Functional Perturbation

According to (7), an important factor that affects the privacy gap, in addition to δ_i, is the choice of the *perturbation slope*, $\tilde{m}_i$. In this subsection, we investigate what the best choice for $\tilde{m}_i$ is. In the next section, we further discuss how to leverage the choice of δ_i so that privacy is still ensured for functions in a particular vicinity.

It is reasonable to choose the perturbation slopes in such a way that the difference between the minimum of the perturbed function and the true function is reduced as much as possible. An approximated upper bound of this difference can be obtained by leveraging mixed-monotone inclusion functions (cf. Proposition 2). Given the common nonlinear terms h, this bound can be minimized by reducing the difference of the linear terms for all possible subintervals of the initial interval domain $\mathcal{X}_0$. This results into the following robust optimization:

$$\tilde{m}^*= \underset{\tilde{m}\in\mathbb{R}^{1\times n},\forall[\underline{x},\overline{x}]\subseteq[\underline{x}_0,\overline{x}_0]}{\arg\min} |(\hat{m}^{\oplus}-m^{\oplus})\underline{x}-(\hat{m}^{\ominus}-m^{\ominus})\overline{x}|, \tag{12}$$

where $\hat{m} \triangleq m+\tilde{m}$. Note that the choice of $\tilde{m}$ through (12) is not necessarily optimal in the sense that it minimizes the privacy gap or maximizes the accuracy of the chosen optimization method to solve the problem. However, given that our goal is to choose the perturbation slope *independently* of the chosen optimization method, it is reasonable to use (12). Moreover, a significant advantage of designing the perturbation slope through (12) is that it provides us with a *tractable approach* to obtain $\tilde{m}$, which is done via the transformation of (12) into a linear program (LP), discussed in the following lemma.

Lemma 1 (Tractable Computation of Perturbation Slopes): The robust optimization problem in (12) can be equivalently reformulated to the following linear program:

$$\begin{aligned}&\min_{\xi\in\mathbb{R}^{2n+1},p_1,p_2\in\mathbb{R}^{3n}} \begin{bmatrix}\mathbf{0}_{2n}^\top & 1\end{bmatrix}\xi \\ &\text{s. t.} \quad \Lambda\xi \leq l, \ p_1^\top d \leq 0, \ p_2^\top d \leq 0, \\ &\quad \Gamma^\top p_1 = \xi, \ -\Gamma^\top p_2 = \xi, p_1 \geq \mathbf{0}_{3n}, \ p_2 \geq \mathbf{0}_{3n},\end{aligned} \tag{13}$$

where $d \triangleq \begin{bmatrix}\overline{x}_0^\top & \underline{x}_0^\top & \mathbf{0}_n^\top\end{bmatrix}^\top, l \triangleq \begin{bmatrix}m^{\oplus} & m^{\ominus} & 0\end{bmatrix}^\top$,

$$\Gamma\triangleq\begin{bmatrix}\mathbf{0}_{n\times n} & -I_n & \mathbf{0}_n \\ -I_n & \mathbf{0}_{n\times n} & \mathbf{0}_n \\ I_n & I_n & \mathbf{0}_n\end{bmatrix}, \Lambda\triangleq\begin{bmatrix}-I_n & \mathbf{0}_{n\times n} & \mathbf{0}_n \\ \mathbf{0}_{n\times n} & -I_n & \mathbf{0}_n \\ \mathbf{0}_n^\top & \mathbf{0}_n^\top & -1\end{bmatrix}.$$

$$\text{Moreover,} \qquad \tilde{m}^* = (\xi^*)^\top \begin{bmatrix}I_n & -I_n & \mathbf{0}_n^\top\end{bmatrix}^\top, \tag{14}$$

where $\tilde{m}^*$ and ξ^* are solutions to the robust optimization in (12) and the LP in (13), respectively.

Proof: First, note that the robust program in (12) can be equivalently written as follows, with $\hat{m} \triangleq m + \tilde{m}$:

$$\begin{aligned}&\min_{\{\tilde{m}\in\mathbb{R}^{1\times n},\theta\geq0\}} \theta \\ &\text{s.t.} -\theta\leq(\hat{m}^{\oplus}-m^{\oplus})\underline{x}-(\hat{m}^{\ominus}-m^{\ominus})\overline{x}\leq\theta, \forall[\underline{x}, \overline{x}] \subseteq [\underline{x}_0, \overline{x}_0].\end{aligned}$$

In turn, by considering the change of variables $\eta \triangleq \hat{m}^{\oplus} - m^{\oplus}, \rho \triangleq \hat{m}^{\ominus} - m^{\ominus}, \xi \triangleq [\eta \ \rho \ \theta]^\top, a_1 \triangleq [\underline{x}^\top - \overline{x}^\top - 1]^\top, a_2 \triangleq [-\underline{x}^\top \overline{x}^\top - 1]^\top$, the latter can be reformulated as:

$$\begin{aligned}&\min_{\{\xi\}} \begin{bmatrix}\mathbf{0}_{2n}^\top & 1\end{bmatrix}\xi \\ &\text{s. t.}\, \Lambda\xi\leq l, [a_1 \ a_2]^\top\xi \leq \mathbf{0}_2, \forall a_1, a_2 \ \text{ s. t.}\, \Gamma a_1\leq d, -\Gamma a_2\leq d,\end{aligned}$$

with c, Γ, Λ, d and l given under (13). Furthermore, by [23, Section 1.2.1], the above robust LP can be equivalently cast as the regular LP in (13). Finally, with $\xi^* = [\eta^* \ \rho^* \ \theta^*]^\top$ being a solution to (13), $(\xi^*)^\top \begin{bmatrix}I_n & -I_n & \mathbf{0}_n^\top\end{bmatrix}^\top = \eta^*-\rho^* = \hat{m}^{*\oplus} - m^{\oplus} - (\hat{m}^{*\ominus} - m^{\ominus}) = \hat{m}^* - m = \tilde{m}^*$. ■

C. Guaranteed Private Mechanism & Accuracy Analysis

In this subsection, and in view of the results of the previous section, we slightly modify our privacy mechanism, and investigate its accuracy when applied to a distributed optimization setting. In particular, we show that, regardless of the distributed and convergent optimization algorithm employed, a perturbation of the problem objective functions via the map of the form of Theorem 1 ensures guaranteed privacy, while remaining reasonably accurate. In other words, we show that there is a computable and reasonably tight upper bound for the error caused by perturbations, regardless of the chosen optimization algorithm. To do so, we require that each agent $i \in \{1, \ldots, N\}$ computes the function g_i, where

$$\forall x\in\mathcal{X}_0, g(x)\triangleq\textstyle\sum_{i=1}^N g_i(x), g_i(x)=f_i(x) + \tilde{m}_i x, \tag{15}$$

and, as we explain next, $\tilde{m}_i$ is constrained by a mild condition that is characterized by $\tilde{m}_i^* = (\xi_i^*)^\top \begin{bmatrix}I_n & -I_n & \mathbf{0}_n^\top\end{bmatrix}^\top$. Moreover, ξ_i^* solves (13) after replacing m with m_i. After this process, agents implement *any* distributed optimization algorithm with the modified objective functions $\{g_i\}_{i=1}^N$. Let

$$\mathbb{X}_g = \arg\min_{x\in\mathcal{X}_0} \sum_{i=1}^N g_i(x) \triangleq \{\tilde{x}^*\in\mathcal{X}_0 \,|\, g(\tilde{x}^*) \le g(x), \forall x\in\mathcal{X}_0\},$$
$$\mathbb{X}_f = \arg\min_{x\in\mathcal{X}_0} \sum_{i=1}^N f_i(x) \triangleq \{x^*\in\mathcal{X}_0 \,|\, f(x^*) \le f(x), \forall x\in\mathcal{X}_0\}, \quad (16)$$

denote the set of possible outputs of the distributed algorithm, and the *set* of optimizers of the original problem (1), respectively. The following theorem characterizes the accuracy and privacy of the corresponding perturbation introduced in Section IV-B, in terms of providing tractable upper bounds for the errors incurred when using them.

Theorem 2 (Private Mechanism & Its Accuracy): Consider a group of N agents that aim to collectively solve the distributed nonconvex optimization (2). Suppose Assumption 1 holds, denote $\Delta \triangleq \overline{x}_0 - \underline{x}_0$, and define

$$\delta_i^* \triangleq \max(|\tilde{m}_i^{*\oplus}\overline{x}_0 - \tilde{m}_i^{*\ominus}\underline{x}_0|, |\tilde{m}_i^{*\oplus}\underline{x}_0 - \tilde{m}_i^{*\ominus}\overline{x}_0|), \quad (17)$$

with $\tilde{m}_i^*$ given in (15). Then,
(i) For any $\delta_i \ge \delta_i^*$, the family $G = \{g_i\}_{i=1}^N$ with an arbitrary perturbation $\tilde{m}_i$ such that $\tilde{m}_i\Delta \le \delta_i^*$, belongs to a $\delta \ge \max_i \delta_i^*$-vicinity of the family $F = \{f_i\}_{i=1}^N$. Moreover, the mapping $\mathcal{M}$ given in Theorem 1 for this class of perturbations, is $\epsilon = \max_{i\in\{1,\ldots,N\}} \epsilon_i$-guaranteed private, where $\epsilon_i = \beta(f_i, \tilde{m}_i, \mathcal{X}_0, \delta_i)$.
(ii) The (worst-case) accuracy error satisfies:

$$e(\{f_i\}_{i=1}^N, \{\tilde{m}_i\}_{i=1}^N, \mathcal{X}_0) \triangleq \max_{x^*\in\mathbb{X}_f, \tilde{x}^*\in\mathbb{X}_g} \|x^* - \tilde{x}^*\|_\infty \le \text{UB}, \quad (18)$$

where the upper bound UB can be computed as:

$$\begin{aligned} \text{UB} = &\max_{\{y\in\mathcal{X}_0, z\in\mathcal{X}_0, \theta\in\mathbb{R}_{\ge 0}\}} \theta \\ &\text{s.t.} -\theta\mathbf{1}_n \le y - z \le \theta\mathbf{1}_n, \tilde{m}_i(y-z) \le 0, 1\le i\le N, \end{aligned} \quad (19)$$

and $\mathbb{X}_f, \mathbb{X}_g$ are given in (16).

Proof: To prove **(i)**, note that by (15) and [24, Lemma 1], $|g_i(x) - f_i(x)| = \tilde{m}_i x \in [\tilde{m}_i^{\oplus}\underline{x}_0 - \tilde{m}_i^{\ominus}\overline{x}_0, \tilde{m}_i^{\oplus}\overline{x}_0 - \tilde{m}_i^{\ominus}\underline{x}_0]$, $\forall x \in \mathcal{X}_0$, implying that $|g_i(x) - f_i(x)| \le |\tilde{m}_i|\Delta \le \delta_i^* \triangleq \max(|\tilde{m}_i^{*\oplus}\overline{x}_0 - \tilde{m}_i^{*\ominus}\underline{x}_0|, |\tilde{m}_i^{*\oplus}\underline{x}_0 - \tilde{m}_i^{*\ominus}\overline{x}_0|) \le \delta_i$, $\forall x \in \mathcal{X}_0$. Then, **(i)** follows from applying Theorem 1 on each f_i.

To prove **(ii)**, first note that for any $(x^*, \tilde{x}^*) \in \mathbb{X}_f \times \mathbb{X}_g$:

$$\begin{aligned} f(x^*) &= h(x^*) + mx^* \le f(\tilde{x}^*) = f(\tilde{x}^*) + \tilde{m}\tilde{x}^* - \tilde{m}\tilde{x}^* \\ &= g(\tilde{x}^*) - \tilde{m}\tilde{x}^* \le g(x^*) - \tilde{m}\tilde{x}^* = h(x^*) + mx^* + \tilde{m}x^* - \tilde{m}\tilde{x}^*, \end{aligned}$$

where the first and second inequalities follow from the fact that x^* and $\tilde{x}^*$ are minimizers of f and g, respectively. Therefore, given any $(x^*, \tilde{x}^*) \in \mathbb{X}_f \times \mathbb{X}_g$ and any perturbation slope, it is necessary that $\tilde{m}(x^* - \tilde{x}^*) \le 0$. By this and defining $\theta = \|x^* - \tilde{x}^*\|_\infty$, (18) is equivalent to

$$\begin{aligned} e(\{f_i\}_{i=1}^N, \{\tilde{m}_i\}_{i=1}^N, \mathcal{X}_0) = &\max_{\{y\in\mathbb{X}_f, z\in\mathbb{X}_g, \theta\in\mathbb{R}_{\ge 0}\}} \theta \\ &\text{s.t.} -\theta\mathbf{1}_n \le y - z \le \theta\mathbf{1}_n, \tilde{m}_i(y-z) \le 0, 1\le i\le N. \end{aligned} \quad (20)$$

Finally, comparing (19) and (20) indicates that the optimal value of the former is an upper bound for the optimal value of the latter since the feasible set of the latter is a subset of the one for the former, i.e., $\mathbb{X}_f \times \mathbb{X}_g \subseteq \mathcal{X}_0 \times \mathcal{X}_0$. ■

As a consequence of Theorem 2, since the choice of $\tilde{m}_i$ is almost arbitrary (constrained by the mild condition $\tilde{m}_i\Delta \le \delta_i^*$), then, it is very unlikely for an adversary to know/guess $\tilde{m}_i$. From this perspective, the process remains private, with the level of privacy given in (7), while the entire process remains accurate with an error less than the upper bound given in (19). Furthermore, $\tilde{m}_i\Delta \le \delta_i^*$ can be interpreted as a significantly weaker counterpart of the required conditions on the perturbation noise in differential privacy, e.g., in [14]. It is also worth emphasizing that, though it might be conservative depending on objective function and constraints, the computed accuracy (19) provides an upper bound for the accuracy error *regardless of the chosen algorithm*. This can be interpreted as an additional degree of resiliency of the proposed privacy-preserving mechanism against perturbing the selected optimization algorithms.

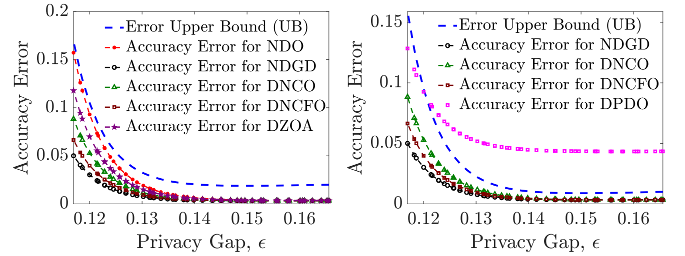

Fig. 1: **Left**: theoretical accuracy error upper bound (UB) computed according to (19) in Theorem 2 with the perturbation slope $\tilde{m}^*$ obtained by solving the LP in 13, as well as true accuracy error $\|x^* - \tilde{x}^*\|_\infty$ for 50 randomly sampled perturbations $\tilde{m}$ from the normal distribution $\mathcal{N}(\tilde{m}^*, 1)$ obtained by applying the nonconvex distributed optimization algorithms NDO [25], NDGD [26], DNCO [27], DNCFO [28] and DZOA [29]. **Right**: comparison of the guaranteed privacy errors and upper bound with the one from a differential private distributed optimization (DPDO) [8] algorithm.

V. ILLUSTRATIVE EXAMPLE

To illustrate the effectiveness of our approach, we considered a nonconvex distributed optimization example from [25], which is in the from of (2), with $n = 1, N = 3$ and $\mathcal{X}_0 = [-10, 10]$, where $f_1(x) = (x^3 - 16x)(x+2)$, $f_2(x) = (0.5x^3 + x^2)(x-4)$ and $f_3(x) = (x+2)^2(x-4)$, with the global optimizer $x^* = [2.62\ 2.62\ 2.62]^\top$. We implemented the following five algorithms[2]: the *nonconvex distributed optimization* (NDO) proposed in [25], the *nonconvex decentralized gradient descent* (NDGD) approach in [26], the *distributed nonconvex constrained optimization* (DNCO) method introduced in [27], the *distributed nonconvex first-order optimization* (DNCFO) algorithm in [28], and the *distributed zero-order algorithm* (DZOA) from [29].

Using Lemma 1, a set of tractable perturbation slopes was obtained as $\tilde{m}^* = \{\tilde{m}_1^*, \tilde{m}_2^*, \tilde{m}_3^*\} = \{0.52, 0.73, 0.38\}$ through solving the LP in (13). The ϵ-guaranteed privacy gaps of the mechanism defined in Theorem 2 when using $\tilde{m}_i^*$ are given by $\{\epsilon_1^*, \epsilon_2^*, \epsilon_3^*\} = \{0.14, 0.32, 0.68\}$, respectively. Further, to study the compromise between privacy and accuracy, we randomly picked 50 samples of $\tilde{m}$ chosen from the normal distribution $\mathcal{N}(\tilde{m}^*, 1)$ and applied Theorem 2, aiming to compute the corresponding $\epsilon = \max_{\{i=1,\ldots,3\}} \epsilon_i$ and

[2] Similar to [25], the communication networks for two consecutive time slots are chosen to be the following two graph combinations: $(1 \leftrightarrow 2, 2 \leftrightarrow 3), (1 \leftrightarrow 2, 1 \leftrightarrow 3)$ or $(1 \leftrightarrow 2, 1 \leftrightarrow 3), (1 \leftrightarrow 2, 2 \leftrightarrow 3)$. In other words, for each two consecutive time slots, either nodes 1 & 2, as well as 2 & 3 are connected and then it switches to 1 & 2, as well as 1 & 3, or vice versa.

the worst-case accuracy error, i.e., $e(\{f_i\}_{i=1}^3, \{\tilde{m}_i\}_{i=1}^3, \mathcal{X}_0)$ (cf. (18)), as well as the theoretical error upper bound UB (cf. (19)) for each sampled $\tilde{m}$. For illustration, in the left plot in Figure 1, the computed values for the privacy gap (ϵ) are sorted in an ascending order along the horizontal axis, which results in descending (decreasing) corresponding errors, after algorithms converge. As can be observed, at the highest privacy ($\epsilon = 0.116$), we obtain the lowest accuracy (i.e., highest accuracy error) which still can be very tightly approximated with the theoretical upper bound $\text{UB} = 0.153$ for all the optimization algorithms errors. Moreover, eventually, when privacy is the lowest ($\epsilon = 0.155$), all the algorithms converge to the lowest accuracy error ($e = 0.01$), which again can be reasonably over-approximated by the corresponding theoretical upper bound $\text{UB} = 0.025$. Moreover, as can be seen, the error upper bound is independent of the chosen optimization method and is bounding all error sequences, noting that the different convergence results depend on the type of nonconvex method employed. Next, to compare our notion of privacy with a benchmark perturbation-based approach, we considered the *differentially-private distributed optimization* (DPDO) algorithm introduced in [8]. Note that this algorithm (as well as all others in the literature to our best knowledge), is only applicable to (strongly) convex objective functions. To make it applicable to the considered objective function from [25], we restricted the problem to a subset of the initial domain, i.e., $\mathcal{X}_0' = [1.5, 4]$ where strong convexity holds. After applying DPDO, and as can be seen from the right plot in Figure 1, we observe the following. First, the corresponding solution error given the DPDO algorithm is more conservative than the guaranteed privacy-based error upper bound for ϵ greater than some small threshold. Second, the guaranteed-privacy errors are all bounded by the estimated upper bound (only a subset of them are plotted in the right figure for the sake of clarity), and this bound decreases consistently to small values as the privacy gap ϵ increases. This clearly illustrates that not only is guaranteed privacy applicable in nonconvex settings as opposed to differential privacy, but also that the bound provided by guaranteed privacy is an improvement over the error bounds provided by DPDO in convex cases.

VI. Conclusion and Future Work

This paper introduced a novel notion of guaranteed privacy for a broad class of differentiable locally Lipschitz nonconvex distributed optimization problems. We showed how this property holds for a deterministic type of perturbation mechanisms, which exploit the Jacobian sign-stability of the problem objective functions. Furthermore, using robust optimization techniques, a tractable approach was provided to further restrict the mechanism in a way that allows for the quantification of the accuracy bounds of the method. In particular, these bounds were shown to be decreasing with respect to the privacy gap, as illustrated through simulations. Future work will consider designing privacy-preserving estimation, verification and resource/task allocation algorithms in networked CPS.

References

[1] J. Cortés, G.E. Dullerud, S. Han, J. Le Ny, S. Mitra, and G.J. Pappas. Differential privacy in control and network systems. In *Proc. IEEE Conf. Decis. Control*, pages 4252–4272, 2016.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[3] K. Chaudhuri, C. Monteleoni, and A.D. Sarwate. Differentially private empirical risk minimization. *J. Mach. Learn. Research*, 12(3), 2011.

[4] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional mechanism: regression analysis under differential privacy. *Proceedings of the VLDB Endowment*, 5(11), 2012.

[5] R. Hall, A. Rinaldo, and L. Wasserman. Differential privacy for functions and functional data. *The Journal of Machine Learning Research*, 14(1):703–727, 2013.

[6] Y. Wang, Z. Huang, S. Mitra, and G.E. Dullerud. Differential privacy in linear distributed control systems: Entropy minimizing mechanisms and performance tradeoffs. *IEEE Transactions on Control of Network Systems*, 4(1):118–130, 2017.

[7] Y. Han and S. Martínez. A numerical verification framework for differential privacy in estimation. *IEEE Control Syst. Lett.*, 6:1712–1717, 2021.

[8] Z. Huang, S. Mitra, and N. Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 international conference on distributed computing and networking*, pages 1–10, 2015.

[9] S. Han, U. Topcu, and G. Pappas. Differentially private distributed constrained optimization. *IEEE Trans. Autom. Control*, 62(1):50–64, 2016.

[10] M.T. Hale and M. Egerstedty. Differentially private cloud-based multi-agent optimization with constraints. In *2015 American Control Conference (ACC)*, pages 1235–1240. IEEE, 2015.

[11] T. Ding, S. Zhu, J. He, C. Chen, and X. Guan. Differentially private distributed optimization via state and direction perturbation in multiagent systems. *IEEE Tran. Autom. Control*, 67(2):722–737, 2021.

[12] Q. Li, R. Heusdens, and M.G. Christensen. Privacy-preserving distributed optimization via subspace perturbation: A general framework. *IEEE Transactions on Signal Processing*, 68:5983–5996, 2020.

[13] M. Ye, G. Hu, L. Xie, and S. Xu. Differentially private distributed Nash equilibrium seeking for aggregative games. *IEEE Trans. Autom. Control*, 67(5):2451–2458, 2021.

[14] E. Nozari, P. Tallapragada, and J. Cortés. Differentially private distributed convex optimization via functional perturbation. *IEEE Transactions on Control of Network Systems*, 5(1):395–408, 2016.

[15] Y. Lu and M. Zhu. Privacy-preserving distributed optimization using homomorphic encryption. *Automatica*, 96:314–325, 2018.

[16] C. Fioravanti, G. Oliva, S. Panzieri, and C.N. Hadjicostis. Private consensus using chaotic oscillator-based encryption. In *2022 30th Mediterranean Conf. on Control and Automation*, pages 927–932.

[17] Y. Wang, H. Sibai, M. Yen, S. Mitra, and Dullerud G.E. Differentially private algorithms for statistical verification of cyber-physical systems. *IEEE Open Journal of Control Systems*, 1:294–305, 2022.

[18] D. Bertsekas, A. Nedic, and A. Ozdaglar. *Convex analysis and optimization*, volume 1. Athena Scientific, 2003.

[19] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. Applied interval analysis. *ed: Springer, London*, 2001.

[20] M. Khajenejad and S.Z. Yong. Tight remainder-form decomposition functions with applications to constrained reachability and interval observer design. *IEEE Trans. Autom. Control, accepted, arXiv preprint arXiv:2103.08638*, 2022.

[21] L. Yang, O. Mickelin, and N. Ozay. On sufficient conditions for mixed monotonicity. *IEEE Trans. Autom. Control*, 64(12):5080–5085, 2019.

[22] M. Khajenejad, F. Shoaib, and S.Z. Yong. Guaranteed state estimation via indirect polytopic set computation for nonlinear discrete-time systems. In *Proc. IEEE Conf. Decis. Control*, pages 6167–6174, 2021.

[23] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*, volume 28. Princeton university press, 2009.

[24] D. Efimov, T. Raïssi, S. Chebotarev, and A. Zolghadri. Interval state observer for nonlinear time varying systems. *Automatica*, 49(1):200–205, 2013.

[25] T. Tatarenko and B. Touri. Nonconvex distributed optimization. *IEEE Trans. Autom. Control*, 62(8):3744–3757, 2017.

[26] J. Zeng and W. Yin. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66(11):2834–2848, 2018.

[27] G. Scutari and Y. Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Math. Prog.*, 176(1):497–544, 2019.

[28] H. Sun and M. Hong. Distributed nonconvex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. *IEEE Tran. on Signal processing*, 67(22):5912–5928, 2019.

[29] Y. Tang, J. Zhang, and N. Li. Distributed zero-order algorithms for nonconvex multi-agent optimization. *IEEE Trans. Control Net. Systs.*, 8(1):269–281, 2020.