Local differential privacy for multi-agent distributed optimal power flow

Dobbe, Roel; Pu, Ye; Zhu, Jingge; Ramchandran, Kannan; Tomlin, Claire

DOI
10.1109/ISGT-Europe47291.2020.9248851

Publication date
2020

Document Version
Accepted author manuscript

Published in
Proceedings of 2020 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe 2020

Citation (APA)
Dobbe, R., Pu, Y., Zhu, J., Ramchandran, K., & Tomlin, C. (2020). Local differential privacy for multi-agent distributed optimal power flow. In *Proceedings of 2020 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe 2020* (pp. 265-269). Article 9248851 (IEEE PES Innovative Smart Grid Technologies Conference Europe; Vol. 2020-October). IEEE. https://doi.org/10.1109/ISGT-Europe47291.2020.9248851

Local Differential Privacy for Multi-Agent Distributed Optimal Power Flow

Roel Dobbe*†, Ye Pu*, Jingge Zhu*, Kannan Ramchandran, Claire Tomlin

***Abstract*—Real-time data-driven optimization and control problems over networks, such as in traffic or energy systems, may require sensitive information of participating agents to calculate solutions and decision variables. Adversaries with access to coordination signals may potentially decode information on individual agents and put privacy at risk. We use the Inexact Alternating Minimization Algorithm to instantiate local differential privacy for distributed optimization, addressing situations in which individual agents need to protect their individual data, in the form of optimization parameters, from all other agents and any central authority. This mechanism allows agents to customize their own privacy level based on local needs and parameter sensitivities. The resulting algorithm works across a large family of convex distributed optimization problems. We implement the method on a distributed optimal power flow problem that aims to prevent overload on critical branches in a radial network.**

I. Introduction

Advances in sensing and computing enable various infrastructures, such as traffic or energy networks, to perform optimization and control problems in real-time throughout a network. Often the scale of such problems desires a distributed implementation that can be solved quickly enough to allow for high frequency control actions. To enable this, a network may be split up into sub-networks governed by different agents, who exchange their local optimization variables with neighbors and/or a central operator to iteratively solve the optimization problem. Exchanging optimization variables between agents and the changes therein may reveal private information, such as whether someone is home and what kind of appliances someone is using [8]. In addition, there is growing understanding that secondary information may be inferred from the communicated variables, including the parameters used in the local objective and constraints, which may reveal sensitive information such as prices and capacity [7].

To make matters more challenging, different agents may be competing with each other to serve an operator with their service. Knowing the control capacity of and prices negotiated by other players can help in negotiating with the operator and leads to strategic behavior and untruthful communication, which harms the quality of solution to the distributed optimization problem. As such, both privacy needs and commercial dynamics may motivate the development of agent-to-agent distributed optimization algorithms that can mask sensitive information in objectives and constraints.

In recent years, various privacy-preserving algorithms have been proposed for distributed optimization and control problems, using various privacy metrics. The differential privacy framework [6] has gained most attention, and is particularly lauded for its robustness to auxiliary side information that an adversary might have to complement information gained from a particular algorithm, providing stronger privacy guarantees than other existing metrics. The framework assumes a setting in which sensitive information is stored in a database by a trustworthy curator, which can provide answers to external queries. A system is made "differentially private" by randomizing its answers in such a way that the distribution over published outputs is not too sensitive to changes in the stored data. These perturbations can be designed to make it provably difficult for an adversary to make inferences about individual records from the published outputs.

In the setting of distributed optimization, each agent is its own curator managing its own locally private information and communication of its optimization variables to neighboring agents or a central operator. In order to preserve differential privacy, each curator has to ensure that the output of queries, that is the communicated variables, remain approximately unchanged if local parameters relating to its objective or constraints are modified.

Related Work

This work complements an existing and rapidly growing body of literature on incorporating *differential privacy* into resource allocation and, most relevant here, in distributed optimization, control and networked systems. A recent elaborate tutorial paper by Cortés et al. [3] covers differential privacy for *distributed* optimization, and distinguishes between message-perturbing and objective-perturbing strategies for distributed optimization. In the first category, coordination messages are perturbed with noise before sent, either to neighbors or a central node, depending on the specific algorithm. Huang et al. [10] proposed a technique for disguising private information in the local objective function, and Han et al. [7] considered problems where the private information is encoded in the individual constraints. In the second category, each agent's objective function parameters are perturbed with noise in a differentially private manner, which guarantees differential privacy at the functional level and is preferred for systems with asymptotically stable dynamics [11]. This is the only work

Roel Dobbe is with the Department of Technology, Policy and Management at Delft University of Technology. Ye Pu and Jingge Zhu are with the Department of Electrical and Electronic Engineering at University of Melbourne. Kannan Ramchandran and Claire Tomlin are with the Department of Electrical Engineering and Computer Sciences at UC Berkeley. *: These authors contributed equally to this paper. †:Corresponding author: r.i.j.dobbe@tudelft.nl.

we found developing different levels of privacy for individual agents, however no analysis of such scenarios is done.

The above works are selective in that these consider privacy-preserving mechanisms for either constraints, objectives or initial states. An exception is the work Hsu et al. [9] on LPs, which can handle both private objectives and constraints.

Local differential privacy was formally defined in [5] for general statistical problems to denote situations in which "data remains private even from the statistician or learner". We develop an equivalent definition for distributed optimization, addressing *situations in which individual agents need to protect their individual data, in the form of optimization parameters, from other agents and any central authority*. We acknowledge recent work, which proposes an algorithm that can address local differential privacy via functional perturbations [11]. This account does not formally motivate, define or analyze local differential privacy, which we focus on in this manuscript. In addition, it scopes local privacy protection to parameters in individual objective functions, leaving aside parameters in constraints. Similarly, other works also limit protection to either individual objective functions [10] or individual constraint [7].

Contributions

Motivated by personal privacy and commercial secrecy concerns in distributed optimization of electricity networks, we investigate the problem of *preserving differential privacy of local objectives and constraints in distributed constrained optimization with agent-to-agent communication*. Building on previous works on privacy-aware distributed optimization via message perturbation [7], [10], we develop and analyze the notion of *local differential privacy*.

Our formulation enables us to *develop privacy guarantees for both local objective function parameters and local constraint parameters*. More specifically, the proposed algorithm solves a general class of convex optimization problems where each agent has a local objective function and a local constraint, and agents communicate with neighbors/adjacent agents, not necessarily including a central authority.

We show that the private optimization algorithm can be formulated as an instance of the Inexact Alternating Minimization Algorithm (IAMA) for distributed optimization [12]. This algorithm allows provable convergence under computation and communication errors. This property is exploited to provide privacy by injecting noise large enough to hide sensitive information, while small enough to exploit the convergence properties of IAMA. We derive and analyze the trade-off between the privacy level and sub-optimality of the algorithm, providing insight in the complexities of implementing differential privacy mechanisms. This trade-off between sub-optimality and differential privacy allows us to determine a *privacy budget* that captures the allowable cumulative variance of noise injected throughout the network that achieves a desired level of (sub-)optimality.

II. Preliminaries and Problem Statement

In this section, we consider a distributed optimization problem on a network of M sub-systems (nodes). The sub-systems communicate according to a fixed undirected graph $G = (\mathcal{V}, \mathcal{E})$. The vertex set $\mathcal{V} = \{1, 2, \cdots, M\}$ represents the sub-systems and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ specifies pairs of sub-systems that can communicate. If $(i, j) \in \mathcal{E}$, we say that sub-systems i and j are neighbors, and we denote by $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$ the set of the neighbors of sub-system i. Note that $\mathcal{N}_i$ includes i. The cardinality of $\mathcal{N}_i$ is denoted by $|\mathcal{N}_i|$. We use a vector v_i to denote the local variable of subsystem i and v_i can be of different dimensions for different i. The collection of these local variables is denoted as $v = [v_1^T, \cdots, v_M^T]^T$. Furthermore, the concatenation of the local variable v_i of sub-system i and the variables of its neighbors $v_j, j \in \mathcal{N}_i$ is denoted by z_i. With appropriate selection matrices E_i and F_{ji}, the variables have the following relationship: $z_i = E_i v$ and $v_i = F_{ji} z_j$, $j \in \mathcal{N}_i$, which implies the relation between the local variable v_i and the global variable v, i.e. $v_i = F_{ji} E_j v$, $j \in \mathcal{N}_i$. With the notations above, we consider the following distributed optimization problem:

Problem II.1 (Distributed Optimization)**.**

$$\min_{z,v} \quad \sum_{i=1}^{M} f_i(z_i) \tag{1}$$

$$s.t. \quad z_i \in \mathbb{C}_i, \quad z_i = E_i v, \quad i = 1, 2, \cdots, M\,, \tag{2}$$

where f_i is the local cost function for node i which is assumed to be strongly convex with a convexity modulus $\rho_{f_i} > 0$, and to have a Lipschitz continuous gradient with a Lipschitz constant $L(\nabla f_i) > 0$. The local constraint $\mathbb{C}_i$ is assumed to be a convex set which represents a convex local constraint on z_i, i.e. the concatenation of the variables of sub-system i and the variables of its neighbors.

The above problem formulation is fairly general and can represent a large class of problems in practice. In particular it includes the following quadratic programming problem, which we study as a particular instance in our applications.

Problem II.2 (Distributed Quadratic Problem)**.**

$$\min_{z,v} \quad \sum_{i=1}^{M} z_i^T H_i z_i + h_i^T z_i \tag{3}$$

$$s.t. \quad C_i z_i \leq c_i, \quad z_i = E_i v, \quad i = 1, 2, \cdots, M\,,$$

where $H_i \succ 0$. In particular, we will assume that the smallest eigenvalue of H_i satisfies $\lambda_{min}(H_i) := \lambda_{min}^{(i)} > 0$.

A. Local Differential Privacy

We present definitions and properties for differential privacy. Let $\mathcal{P}$ be a set containing some elements from a space $\mathcal{X}$. In the language of differential privacy, the set will be called a *database*, and the elements therein represent private information of individual users. More concretely, in the context of distributed optimization (Problem II.1), this information comprises the private parameters that constitute the local objective $f_i(\cdot)$ and constraints $\mathbb{C}_i$. Given two data bases $\mathcal{P}, \mathcal{P}'$, let adj : $\mathcal{X}^{|\mathcal{P}|} \times \mathcal{X}^{|\mathcal{P}'|} \mapsto [0, \infty)$ denote a metric that encodes the adjacency or distance between two databases. A mechanism or algorithm $\mathcal{A}$ is a mapping from $\mathcal{X}^{|\mathcal{P}|}$ to some set denoting its output space.

In the scenario of our interest, there are multiple agents involved in the algorithm, and each is only concerned with its *own privacy*. In other words, individual agent does not care nor trust other agents in the network. To this end, we extend the notion of algorithm $\mathcal{A}(\mathcal{P})$ to a distributed algorithm $\mathcal{A}(\mathcal{P}_1, \dots, \mathcal{P}_M)$ in a network with M agents, where $\mathcal{P}_i$ is itself a database which denotes the private parameters of agent i. The outputs of the mechanism are the message exchanged between nodes in the network over the time horizon of iterations. This mechanism induces M local mechanisms $\mathcal{A}_1(\mathcal{P}_1, \dots, \mathcal{P}_M), \dots, \mathcal{A}_M(\mathcal{P}_1, \dots, \mathcal{P}_M)$, each executed by one agent. The output of one local mechanism $\mathcal{A}_i$ is the message sent out by node i, i.e. range$(\mathcal{A}_i) \subseteq$ range$(\mathcal{A})$. It is important to realize that although one local mechanism, say $\mathcal{A}_i$, does not necessarily have direct access to the input/database $\mathcal{P}_j, j \neq i$ of other nodes, the output of $\mathcal{A}_i$ could still be affected by $\mathcal{P}_j, j \neq i$ because of the interactions among different nodes. For this reason, we explicitly write $\mathcal{P}_1, \dots, \mathcal{P}_M$ as input to all local mechanisms.

We now let each agent i specify its own level of privacy ϵ_i. To formalize this specification, we require a definition:

Definition II.3 (Local Differential Privacy)**.** *Consider a (global) mechanism $\mathcal{A}$ for a network with M nodes, and a local mechanisms $\mathcal{A}_i, i \in \{1, \dots, M\}$ induced by $\mathcal{A}$. We say that the mechanism $\mathcal{A}$ is ϵ_i-differentially locally private for node i, if for any $\mathcal{S}_i \in$ range$(\mathcal{A}_i)$, it satisfies that*

$$\frac{\mathbb{P}\{\mathcal{A}_i(\mathcal{P}_1, \dots, \mathcal{P}_i, \dots, \mathcal{P}_M) \in \mathcal{S}_i\}}{\mathbb{P}\{\mathcal{A}_i(\mathcal{P}_1, \dots, \mathcal{P}'_i, \dots, \mathcal{P}_M) \in \mathcal{S}_i\}} \leq e^{\epsilon_i}, \qquad (4)$$

where adj$(\mathcal{P}_i, \mathcal{P}'_i) \leq 1$. Moreover, if $\mathcal{A}$ is ϵ_i-differentially locally private for node i, $i = 1, \dots, M$, then we say that the mechanism $\mathcal{A}$ is $(\epsilon_1, \dots, \epsilon_M)$-differentially private.

Algorithm 1 Differentially private distributed algorithm

```
Require: Initialize $\mu_i^0 = 0 \in \mathbb{R}^{z_i}$, $\tau^0 = \min_{1 \leq i \leq M}\{\rho_{f_i}\}$ and $\tau^k = \frac{1}{\tau^0 k}$
  for $k = 1, 2, \cdots$ do
    1: $z_i^k = \text{argmin}_{z_i \in \mathbb{C}_i}\{f_i(z_i) + \langle \mu_i^{k-1}, -z_i \rangle\} + \delta_i^k$
    2: Send $z_i^k$ to all the neighbors of agent $i$.
    3: $v_i^k = \frac{1}{|\mathcal{N}_i|}\sum_{j \in \mathcal{N}_i} F_{ji} z_j^k$.
    4: Send $v_i^k$ to all the neighbors of agent $i$.
    5: $\mu_i^k = \mu_i^{k-1} + \tau^k(E_i v^k - z_i^k)$
  end for
```

In [4], the authors provide a detailed exposé and analysis of the algorithm used to solve Problem II.2 in a way that instantiates and satisfies the local differential privacy criteria in Definition II.3. Here we provide a high-level description for Algorithm 1. At each iteration k, each agent first solves a local version of the minimization problem based on the current dual variable μ_i^{k-1}. In Step 2, each agent sends its local solution to all the neighbours after adding noise for the sake of privacy. In Step 3, each agent computes the average of the solutions received from its neighbouring sub-systems and updates its local variable v_i^k. This step is crucial for this distributed algorithm as it drives the agents to come to a consensus of the global optimal solution. After sending out the local variable to all its neighbours in Step 4, each agent updates its local dual variable in Step 5.

III. Application: Distributed Optimal Power Flow

This section presents a simplified optimal power flow (OPF) problem that inspires the proposed control approach. We consider the setting of a radial distribution feeder, and consider the flow of real power on its branches. We formulate the power flow model and the OPF objectives and develop the distributed OPF problem according to the quadratic problem, as defined in (3). We then discuss the parameters that are subject to privacy requirements and interpret trade-offs.

A. Simplified Optimal Power Flow

Solving the simplified OPF problem requires a model of the electric grid describing both topology and impedances. This information is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ denoting the set of all buses (nodes) in the network, and $\mathcal{E}$ the set of all branches (edges). For ease of presentation and without loss of generality, here we introduce part of the linearized power flow equations over *radial* networks, also known as the *LinDistFlow* equations [2]. In such a network topology, each bus j has one upstream parent bus $\{i \mid (i, j) \in \mathcal{E}\}$ and potentially multiple downstream child buses $\{k \mid (j, k) \in \mathcal{E}\}$. By $\mathcal{D}_j$ we denote the set of all buses downstream of branch (i, j). We assume losses in the network to be negligible and model the power flowing on a branch as the sum of the downstream net load:

$$P_{ij} \approx \sum_{k \in \mathcal{D}_j} \{p_k^{\text{c}} - p_k^{\text{g}} + u_k\} \qquad (5)$$

In this model, capital P_{ij} represents real power flow on a branch from node i to node j for all branches $(i, j) \in \mathcal{E}$, lower case p_i^c is the real power consumption at node i, and p_i^g is its real power generation. This nodal consumption and generation is assumed to be uncontrollable. In addition, we consider controllable nodal injection u_i, available at a subset of nodes $i \in \mathcal{C} \subset \mathcal{V}$ that have a Distributed Energy Resource (DER). In this case study, we aim to prevent overload of real power flow over certain critical branches in an electric network. This aim is formulated through constraints

$$\begin{aligned} \sum_{k \in \mathcal{D}_j} \{p_k^{\text{c}} - p_k^{\text{g}} + u_k\} - \overline{P_{ij}} &\leq 0, \\ \underline{P_{ij}} - \sum_{k \in \mathcal{D}_j} \{p_k^{\text{c}} - p_k^{\text{g}} + u_k\} &\leq 0, \forall (i, j) \in \mathcal{E}_{\text{safe}}, \end{aligned} \qquad (6)$$

$\mathcal{E}_{\text{safe}} \subset \mathcal{E}$ denotes a subset of branches for which power flow limitations are defined, $\overline{P_{ij}}, \underline{P_{ij}}$ denoting the upper and lower power flow bounds on branch $(i,j) \in \mathcal{E}_{\text{safe}}$. In addition, each controlled node i is ultimately limited by the local capacity on total apparent power capacity,

$$\underline{u}_i \le u_i \le \overline{u}_i \,, \forall i \in \mathcal{C} \,. \tag{7}$$

We consider a scenario in which the operator negotiates different prices for different capacities, potentially at different points in time, with different third party DER owners. Let u_i refer to the real power used for the optimization scheme from agent i, and π_i denotes the quadratic price per procuring a kWatt from agent i for the time period that the set points are implemented (typically in the order of minutes). The optimal power flow determines the control setpoints that minimizes an economic objective subject to operational constraints.

$$\begin{aligned} \min_{u_i\,, i \in \mathcal{C}} \quad & \sum_{i \in \mathcal{C}} \pi_i (u_i)^2 \,, \\ \text{s.t.} \quad & (6)\,,(7)\,. \end{aligned} \tag{8}$$

The OPF problem (8) can be recast as an instance of the quadratic distributed optimization problem (3). First, note that the objective is quadratic in the optimization variables u_i, and separable per node. Second, for all nodes $i \in \mathcal{V}$, the capacity box constraints (7) are linear and fully local. The safety constraints (6) require communication to and computation by a central trusted node. To ensure strong convexity of the local problems, the economic cost objectives are shared between each agent i and the central trusted node. Hence, respectively for $\forall i \in \mathcal{C}_{\setminus\{0\}}$ and the central node 0, the objectives read

$$f_i(u_i) = \frac{\pi_i}{2}(u_i)^2 \,, \;\; f_0(z_0) = \sum_{i \in \mathcal{C}} \frac{\pi_i}{2}(u_i)^2 \,. \tag{9}$$

As such, this distributed problem assumes a star-shaped communication structure, in which the a centrally trusted node receives all $u_i, p_i^{\text{c}}, p_i^{\text{g}}$ from the agents. The agents retrieve iterates of u_i from the central node and compute a simple problem with only economic cost and a local capacity constraint.

B. Private Information in Distributed OPF

We consider assigning privacy requirements to two sets of parameters; the prices π_i that the DSO charges to different agents in the network, and the capacities $\underline{u}_i, \overline{u}_i$ available to all agents $i \in \mathcal{C}$. Together, these parameters provide important strategic insight into the commercial position of each agent. An operator may charge different prices for different levels of commitment or for the varying value that the operator gets from the actions of a specific agent at specific time periods or places in the network. In a natural commercial context, the operator may have an interest to hide the prices to other agents. In addition, in a negotiation setting, a strategic agent may want to find out the capacity available by other agents in the network to adjust its bid to the operator, so as to be the first or only agent to be considered, which could lead to asymmetric and potentially unfair bidding situations. As such, in order to give all agents with capacity a fair chance to participate, there is value in hiding the capacity (and price) parameters.

To formulate this as an instance of local differential privacy, we need to define the adjacency metric for all considered parameters. In the case of both prices and capacity, this is achieved by considering the maximum range in which these parameters are expected to lie. The distance metric proposed is the ℓ_1-norm. Given this metric, we need to define a proper adjacency relation, which determines the maximum change in a single parameter that we aim to hide with the differentially private algorithm.

Definition III.1. *(Adjacency Relation for Distributed OPF): For any parameter set $\mathcal{P} = \{f_i(\pi_i), \mathbb{C}_i(\overline{u}_i, \underline{u}_i)\}$ and $\mathcal{P}' = \{f_i'(\pi_i'), \mathbb{C}_i'(\overline{u}_i', \underline{u}_i')\}$, we have adj$(\mathcal{P}, \mathcal{P}') \le 1$ if and only if there exists $i \in [M]$ such that*

$$|\pi_i - \pi_i'| \le \delta\pi \,, \quad |\overline{u}_i - \overline{u}_i'| \le \delta\overline{u} \,, \quad |\underline{u}_i - \underline{u}_i'| \le \delta\underline{u} \,, \tag{10}$$

and $\pi_j = \pi_j'$, $\overline{u}_j = \overline{u}_j'$, $\underline{u}_j = \underline{u}_j'$ for all $j \neq i$.

By setting $\delta\pi$, $\delta\overline{u}$ and $\delta\underline{u}$ respectively as the maximum price offered per unit of energy (i.e. $\overline{\pi}$ if $\pi_i \in [0, \overline{\pi}]$) and the maximum capacity in the network (i.e. $\arg\max_{i \in \mathcal{C}} \overline{u}_i$), we ensure that all parameters in the network are properly covered by the definition.

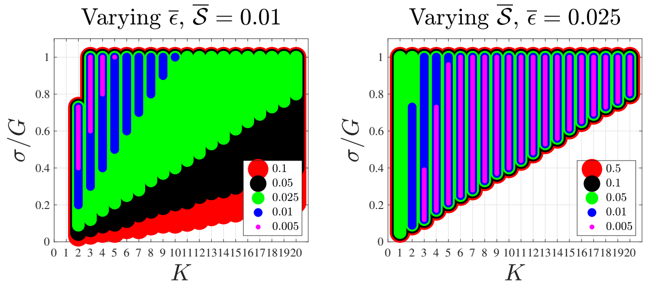

Figure 1: Feasible parameter sets (ν, K) for varying levels of $\overline{\epsilon}$ (left) and $\overline{\mathcal{S}}$ (right), setting $\sigma_i = \sigma$, $\forall i \in \mathcal{C}$.

C. Interpreting Trade-off Between Differential Privacy and Suboptimality

In [4, Section IV-C], we derive and discuss the trade-off relationship between specified levels of suboptimality $\bar{\mathcal{S}}$ and differential privacy $\overline{\epsilon}_i$ based on [4, Theorem III.1] and [4, Theorem III.11], captured in [4, Equation (36)].

$$\frac{K}{\nu_i} \le \frac{u_{i,\max}}{\Theta_i}\overline{\epsilon}_i \,, \quad \frac{M + \sum_{i=1}^{M} \nu_i^2}{K} \le \frac{\overline{\mathcal{S}}}{4}\left(\frac{\pi_{\max}}{u_{\max}}\right)^2 \,, \tag{11}$$

For the purpose of analysis, we assume that $\sigma_i = \sigma$, $\forall i \in \mathcal{C}$, which yields

$$\frac{K}{\nu} \le \frac{u_{i,\max}}{\Theta_i}\overline{\epsilon}_i \,, \quad \frac{1+\nu^2}{K} \le \frac{\overline{\mathcal{S}}}{4M}\left(\frac{\pi_{\max}}{u_{\max}}\right)^2 \,. \tag{12}$$

Here, K the number of iterations of the algorithm, and $\nu := \frac{\sigma}{u_{\max}}$ is the normalized noise-to-signal ratio with $u_{\max} = \max_{i \in \mathcal{C}} \max(|\underline{u}_i|, |\overline{u}_i|)$. Figure 1 shows the feasible set for varying levels of specifications $(\overline{\epsilon}, \overline{\mathcal{S}})$. The first equation shows that the ratio of the number of iterations to the normalized noise needs to be sufficiently small, capped by the specified privacy level $\overline{\epsilon}_i$ and the agent's maximum capacity. It also shows the effect of the sensitivity on this trade-off. The latter equation shows that with increasing number of agents M injecting noise, we need more iterations to achieve the same level of suboptimality. Similarly, if the maximum capacity u_{max} of the agents increases or the maximum price π_{max} decreases, we require more iterations or lower noise variance to maintain the same level of suboptimality.

D. Numerical Results

Numerical results are obtained for the distributed optimal power flow problem. We first evaluate the sensitivity using the sampling-based method in [4, Section III-C] for a smaller size problem. We choose both parameters α and β in [4, Lemma III.4] to be 1.6×10^{-2}, and the sample size to be $n = 3000$. This gives a (lower bound on) sensitivity $\Theta_i = 1.8439$. Using [4, Lemma III.6], we can find an upper bound of the sensitivity $\Theta_i \leq 2$. We can see that the solution given by the sample-based method appears tight to the upper-bound.

We then implement our method on the larger simplified OPF problem for the IEEE 13 Node Test Feeder [1]. We focus on single-phase power flow (aggregated over three phases), and do not model the voltages. We consider each node an agent in the network communicating with neighboring agents connected through an electric wire. All agents have various capacities with $|\underline{u}_i|, |\overline{u}_i| \leq 0.5\text{kW}\ \forall i \in \mathcal{C}$. The prices for all agents vary as $\pi_i \in [10, 100]$ cents/(kW)^2. The critical branches are $\mathcal{E}_{\text{safe}} = \{(650, 632), (632, 645), (632, 671)\}$, with with capacity limits $\overline{P_{ij}} = [3, 0.3, 2]$ kW and $\underline{P_{ij}} = [-3, -0.3, -2]$ kW for the three branches respectively.

In Fig. 2, we demonstrate the convergence performance of Algorithm 1 in [4] for solving the distributed optimization problem in Problem II.2, originating from the OPF problem. We assume that Agent 3 aims at protecting its local matrix h_3 and adds noise to its local solutions in the distributed optimization algorithm. The blue line shows the averaged performance of Algorithm 1 over 300 samples (experiments), where the errors are generated randomly according to a zero-mean Laplace distribution with the variance equal to $\sigma_i = 0.1$. The black line shows the performance of the exact algorithm, for which the errors are set to be zero. We can observe that as the number of iterations K increases, the average difference $\|z^k - z^\star\|$ generated by Algorithm 1 and the deference generated by the exact algorithm $\|z^k - z^\star\|$ decrease for both the cases, however, the convergence speed of Algorithm 1 becomes slower and sub-linear, which supports the findings in Theorem III.11 in [4].

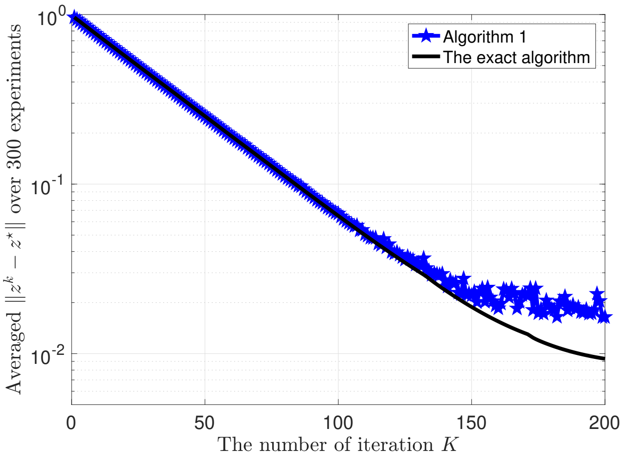

Figure 2: Convergence performance for Algorithm 1 in [4], averaged over 300 experiments (blue) versus the no-error scenario (black).

IV. Conclusions and Future Work

In this paper, we developed local ϵ-differential privacy for distributed optimization, applied to distributed optimal power flow. The method builds on recent advances in inexact alternating minimization algorithm (IAMA). Exploiting IAMA's convergence properties under the existence of errors in communication and computation, we showed one can add noise to agent-to-agent communication in a way that preserves privacy in the specifications of user objectives and constraints while still guaranteeing convergence. The method extends current approaches for differential privacy in distributed optimization by allowing privacy for both objectives and constraints and customization of privacy specifications for individual agents.

References

[1] IEEE Distribution Test Feeders, 2017.

[2] M. Baran and F. Wu. Optimal capacitor placement on radial distribution systems. *IEEE Transactions on Power Delivery*, 4(1):725–734, Jan. 1989.

[3] J. Cortes, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas. Differential privacy in control and network systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4252–4272, Dec. 2016.

[4] R. Dobbe, Y. Pu, J. Zhu, K. Ramchandran, and C. Tomlin. Customized Local Differential Privacy for Multi-Agent Distributed Optimization. *arXiv preprint arXiv:1806.06035*, 2018.

[5] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 429–438. IEEE, 2013.

[6] C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, Aug. 2014.

[7] S. Han, U. Topcu, and G. J. Pappas. Differentially Private Distributed Constrained Optimization. *IEEE Transactions on Automatic Control*, 62(1):50–64, Jan. 2017.

[8] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, Dec. 1992.

[9] J. Hsu, A. Roth, T. Roughgarden, and J. Ullman. Privately solving linear programs. In *International Colloquium on Automata, Languages, and Programming*, pages 612–624. Springer, 2014.

[10] Z. Huang, S. Mitra, and N. Vaidya. Differentially Private Distributed Optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, ICDCN '15, pages 4:1–4:10, New York, NY, USA, 2015. ACM.

[11] 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, 2018.

[12] Y. Pu, M. N. Zeilinger, and C. N. Jones. Inexact fast alternating minimization algorithm for distributed model predictive control. pages 5915–5921, Los Angeles, CA, USA, Dec. 2014.