Differentially Private Distributed Constrained Optimization

Shuo Han, *Member, IEEE*, Ufuk Topcu, and George J. Pappas, *Fellow, IEEE*

***Abstract*—Many resource allocation problems can be formulated as an optimization problem whose constraints contain sensitive information about participating users. This paper concerns a class of resource allocation problems whose objective function depends on the aggregate allocation (i.e., the sum of individual allocations); in particular, we investigate distributed algorithmic solutions that preserve the privacy of participating users. Without privacy considerations, existing distributed algorithms normally consist of a central entity computing and broadcasting certain public coordination signals to participating users. However, the coordination signals often depend on user information, so that an adversary who has access to the coordination signals can potentially decode information on individual users and put user privacy at risk. We present a distributed optimization algorithm that preserves differential privacy, which is a strong notion that guarantees user privacy regardless of any auxiliary information an adversary may have. The algorithm achieves privacy by perturbing the public signals with additive noise, whose magnitude is determined by the sensitivity of the projection operation onto user-specified constraints. By viewing the differentially private algorithm as an implementation of stochastic gradient descent, we are able to derive a bound for the suboptimality of the algorithm. We illustrate the implementation of our algorithm via a case study of electric vehicle charging. Specifically, we derive the sensitivity and present numerical simulations for the algorithm. Through numerical simulations, we are able to investigate various aspects of the algorithm when being used in practice, including the choice of step size, number of iterations, and the trade-off between privacy level and suboptimality.**

***Index Terms*—Data privacy, distributed algorithms, optimization methods.**

I. Introduction

ELECTRIC vehicles (EVs), including pure electric and hybrid plug-in vehicles, are believed to be an important component of future power systems [18]. Studies predict that the number of EVs in the U.S. can reach approximately 1.7 million by year 2020 (assuming a conservative annual growth rate of 20%) [3]. By that time, EVs will become a significant load on the power grid [5], [30], which can lead to undesirable effects such as voltage deviations if charging of the vehicles is uncoordinated.

The key to reducing the impact of EVs on the power grid is to coordinate their charging schedules, which is often cast as a constrained optimization problem with the objective of minimizing the peak load, power loss, or load variance [7], [28]. Due to the large number of vehicles, computing an optimal schedule for all vehicles can be very time consuming if the computation is carried out on a centralized server that collects demand information from users. Instead, it is more desirable that the computation is distributed to individual users. Among others, Ma *et al.* [21] proposed a distributed charging strategy based on the notion of valley-filling charging profiles, which is guaranteed to be optimal when all vehicles have identical (i.e., homogeneous) demand. Gan *et al.* [13] proposed a more general algorithm that is optimal for nonhomogeneous demand and allows asynchronous communication.

In order to solve the constrained optimization problem of scheduling in a distributed manner, the server is required to publish certain public information that is computed based on the tentative demand collected from participating users. Charging demand often contains private information of the users. As a simple example, zero demand from a charging station attached to a single home unit is a good indication that the home owner is away from home. Note that the public coordination signal is received by everyone including potential adversaries whose goal is to decode private user information from the public signal, so that it is desirable to develop solutions for protecting user privacy.

It has been long recognized that *ad hoc* solutions such as anonymization of user data are inadequate to guarantee privacy due to the presence of public side information. A famous case is the reidentification of certain users from an anonymized dataset published by Netflix, which is an American provider of on-demand Internet streaming media. The dataset was provided for hosting an open competition called the Netflix Prize for finding the best algorithm to predict user ratings on films. It has been reported that certain Netflix subscribers can be identified from the anonymized Netflix prize dataset through auxiliary information from the Internet Movie Database (IMDb) [22]. As such, providing rigorous solutions to preserving privacy has

Manuscript received November 14, 2014; revised July 16, 2015, January 26, 2016, and January 28, 2016; accepted February 10, 2016. Date of publication March 11, 2016; date of current version December 26, 2016. This work was supported in part by the National Science Foundation under Grant CNS-1239224 and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Recommended by Associate Editor I. V. Kolmanovsky.

S. Han and G. J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: hanshuo@seas.upenn.edu; pappasg@seas.upenn.edu).

U. Topcu is with the Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, Austin, TX 78712 USA (e-mail: utopcu@utexas.edu).

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

Digital Object Identifier 10.1109/TAC.2016.2541298

become an active area of research. In the field of systems and control, recent work on privacy includes, among others, filtering of streaming data [20], smart metering [26], traffic monitoring [4], and privacy in stochastic control [32].

Recently, the notion of *differential privacy* proposed by Dwork and her collaborators has received attention due to its mathematically rigorous formulation [10]. Compared to other privacy solutions such as k-anonymity [29] and secure multiparty computation [1], differential privacy is robust to arbitrary auxiliary information that an adversary may have and thus provides a stronger privacy guarantee. The original setting assumes that the sensitive user information is held by a trustworthy party (often called *curator* in related literature), and the curator needs to answer external queries (about the sensitive user information) that potentially come from an adversary who is interested in learning information belonging to some user. For example, in EV charging, the curator is the central server that aggregates user information, and the queries correspond to public coordination signals. Informally, preserving differential privacy requires that the curator must ensure that the results of the queries remain approximately unchanged if data belonging to any single user are modified. In other words, the adversary should know little about any single user's information from the results of queries. A recent survey on differential privacy can be found in [9]; there is also a recent textbook on this topic written by Dwork and Roth [11].

Contributions: Motivated by the privacy concerns in EV charging and recent advances in differential privacy, in this paper, we investigate the problem of *preserving differential privacy in distributed constrained optimization*. We present a differentially private distributed algorithm for solving a class of constrained optimization problems. The objective function of the problem needs to be convex and Lipschitz continuously differentiable, and it should only depend on the aggregate allocation (i.e., the sum of individual allocations); the constraints need to be convex and separable. The privacy guarantee of our algorithm is proved using the adaptive composition theorem. We show that the private optimization algorithm can be viewed as an implementation of stochastic gradient descent [24]. Based on previous results on stochastic gradient descent [27], we are able to derive a bound for the suboptimality of our algorithm and reveal the trade-off between privacy and performance of the algorithm.

We illustrate the applicability of this general framework of differentially private distributed constrained optimization in the context of EV charging. To this end, we begin by computing the *sensitivity* of the public signal with respect to changes in private information. Specifically, this requires analyzing the sensitivity of the projection operation onto the user-specified constraints. Although such sensitivity can be difficult to compute for a general problem, using tools in optimization theory, we are able to derive an explicit expression of the sensitivity for the EV charging example. Through numerical simulations, we show that our algorithm is able to provide strong privacy guarantees with little loss in performance when the number of participating users (i.e., vehicles) is large.

Related Work: There is a large body of research work on incorporating differential privacy into resource allocation problems. A part of the work deals with indivisible resources (or equivalently, games with discrete actions), including the work by, among others, Kearns *et al.* [19], Rogers and Roth [25], and Hsu *et al.* [15]. Our paper focuses on the case of *divisible* resources and where private information is contained in the *constraints* of the allocation problem.

In the work of differentially private resource allocation, it is a common theme that the coordination signals are randomly perturbed to avoid revealing private information of the users, such as in the work by Huang *et al.* [17], Hsu *et al.* [16], and our previous work on differentially convex optimization with piecewise affine objectives [14]. Huang *et al.* [17] study the problem of differentially private distributed convex optimization in which the private user information is encoded in the individual cost functions, whereas in our setting the private user information is encoded in the individual constraints. The recent work by Hsu *et al.* [16] on privately solving linear programs is closely related to our work, since their setting also assumes that the private information is contained in the (affine) constraints. Our work can be viewed as a generalization of their setting by extending the form of objective functions and constraints. In particular, the objective function can be any convex and Lipschitz continuously differentiable function that depends on the aggregate allocation, and the constraints need only to be convex and separable. For illustration, we show how to implement the algorithm for a particular set of affine constraints motivated by EV charging.

Paper Organization: The paper is organized as follows. Section II introduces the necessary background on (nonprivate) distributed optimization and, in particular, projected gradient descent. Section III reviews the results in differential privacy and gives a formal problem statement of differentially private distributed constrained optimization. Section IV gives an overview of the main results of the paper. Section V describes a differentially private distributed algorithm that solves a general class of constrained optimization problems. We also study the trade-off between privacy and performance by analyzing the suboptimality of the differentially private algorithm. In Section VI, we illustrate the implementation of our algorithm via a case study of EV charging. In particular, we compute the sensitivity of the projection operation onto user-specified constraints, which is required for implementing our private algorithm. Section VII presents numerical simulations on various aspects of the algorithm when being used in practice, including choice of step size, number of iterations, and the trade-off between privacy level and performance.

II. Background: Distributed Constrained Optimization

A. Notation

Denote the ℓ_p-norm of any $x \in \mathbb{R}^n$ by $\|x\|_p$. The subscript p is dropped in the case of the ℓ_2-norm. For any nonempty convex set $\mathcal{C} \subset \mathbb{R}^n$ and $x \in \mathbb{R}^n$, denote by $\Pi_{\mathcal{C}}(x)$ the projection operator that projects x onto $\mathcal{C}$ in the ℓ_2-norm. Namely, $\Pi_{\mathcal{C}}(x)$ is the solution of the following constrained least-squares problem:

$$\min_{\hat{x}}. \quad \|\hat{x} - x\|^2 \qquad \text{s.t.} \quad \hat{x} \in \mathcal{C}. \tag{1}$$

It can be shown that problem (1) is always feasible and has a unique solution so that $\Pi_{\mathcal{C}}$ is well-defined. For any function f

(not necessarily convex), denote by $\partial f(x)$ the set of subgradients of f at x

$$\partial f(x) := \left\{ g : f(y) \geq f(x) + g^T(y-x) \text{ for all } y \right\}.$$

When f is convex and differentiable at x, the set $\partial f(x)$ becomes a singleton set whose only element is the gradient $\nabla f(x)$. For any function f, denote its range by range(f). For any differentiable function f that depends on multiple variables including x, denote by $\partial_x f$ the partial derivative of f with respect to x. For any $\lambda > 0$, denote by Lap(λ) the zero-mean Laplace probability distribution such that the probability density function of a random variable X obeying the distribution Lap(λ) is $p_X(x) = (1/2\lambda)\exp(-|x|/\lambda)$. The vector consisting all ones is written as $\mathbf{1}$. The symbol $\preceq$ is used to represent elementwise inequality: for any $x, y \in \mathbb{R}^n$, we have $x \preceq y$ if and only if $x_i \leq y_i$ for all $1 \leq i \leq n$. For any positive integer n, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$.

B. Distributed Constrained Optimization

Before discussing privacy issues, we first introduce the necessary background on distributed constrained optimization. We consider a constrained optimization problem over n variables $r_1, r_2, \ldots, r_n \in \mathbb{R}^T$ in the following form:

$$\begin{aligned} \min_{\{r_i\}_{i=1}^n} \quad & U\left(\sum_{i=1}^n r_i\right) \\ \text{s.t.} \quad & r_i \in \mathcal{C}_i, \quad i \in [n]. \end{aligned} \tag{2}$$

Throughout the paper, we assume that the objective function $U : \mathbb{R}^T \to \mathbb{R}$ in problem (2) is differentiable and convex, and its gradient ∇U is L-Lipschitz in the ℓ_2-norm, i.e., there exists $L > 0$ such that

$$\|\nabla U(x) - \nabla U(y)\| \leq L\|x - y\| \quad \text{for all } x, y.$$

The set $\mathcal{C}_i$ is assumed to be convex for all $i \in [n]$. For resource allocation problems, the variable r_i and the constraint set $\mathcal{C}_i$ are used to capture the allocation and constraints on the allocation for user/agent i.

Algorithm 1 Distributed projected gradient descent (with a fixed number of iterations)

```
Input: $U$, $\{\mathcal{C}_i\}_{i=1}^n$, $K$, and step sizes $\{\alpha_k\}_{k=1}^K$.
Output: $\{r_i^{(K+1)}\}_{i=1}^n$.
Initialize $\{r_i^{(1)}\}_{i=1}^n$ arbitrarily. For $k = 1, 2, \ldots, K$, repeat:
  1) Compute $p^{(k)} := \nabla U(\sum_{i=1}^n r_i^{(k)})$.
  2) For $i \in [n]$, update $r_i^{(k+1)}$ according to
        $r_i^{(k+1)} := \Pi_{\mathcal{C}_i}\left(r_i^{(k)} - \alpha_k p^{(k)}\right).$   (3)
```

The optimization problem (2) can be solved iteratively using projected gradient descent, which requires computing the gradients of U with respect to each r_i and projecting the gradients onto the feasible set at each iteration. When the objective function only depends on the sum $\sum_{i=1}^n r_i$, it can be shown that $\nabla_{r_i} U$ is identical for all i, and we have $\nabla_{r_i} U(\sum_{i=1}^n r_i) = \nabla U(\sum_{i=1}^n r_i)$. As a consequence, the computational complexity of the projected gradient descent is dominated by the projection operation and grows with n. For practical applications, the number n can be quite large, so that it is desirable to distribute the projection operation to individual users. A distributed version of the projected gradient descent method applied to problem (2) is shown in Algorithm 1. The algorithm guarantees that the output converges to the optimal solution as $K \to \infty$ with proper choice of step sizes $\{\alpha_k\}_{k=1}^K$ (see [13] for details on how to choose α_k).

III. Problem Formulation

A. Privacy in Distributed Constrained Optimization

In many applications, the specifications of $\mathcal{C}_i$ may contain sensitive information that user i wishes to keep undisclosed from the public. In the framework of differential privacy, it is assumed that an adversary can potentially collaborate with some users in the database in order to learn about other user's information. Under this assumption, the distributed projected descent algorithm (Algorithm 1) can lead to possible loss of privacy of participating users for reasons described below. It can be seen from Algorithm 1 that $\mathcal{C}_i$ affects $r_i^{(k)}$ through (3) and consequently also $p^{(k)}$. Since $p^{(k)}$ is broadcast publicly to every charging station, with enough side information (such as collaborating with some participating users), an adversary who is interested in learning private information about some user i may be able to infer information about $\mathcal{C}_i$ from the public signals $\{p^{(k)}\}_{k=1}^K$. We will later illustrate the privacy issues in the context of EV charging.

B. Differential Privacy

Our goal is to modify the original distributed projected gradient descent algorithm (Algorithm 1) to preserve *differential privacy*. Before giving a formal statement of our problem, we first present some preliminaries on differential privacy. Differential privacy considers a set (called *database*) D that contains private user information to be protected. For convenience, we denote by $\mathcal{D}$ the universe of all possible databases of interest. The information that we would like to obtain from a database D is given by $q(D)$ for some mapping q (called *query*) that acts on D. In differential privacy, preserving privacy is equivalent to hiding changes in the database. Formally, changes in a database can be defined by a symmetric binary relation between two databases called an *adjacency* relation, which is denoted by Adj$(\cdot,\cdot)$; two databases D and D' that satisfy Adj(D, D') are called adjacent databases.

Definition 1 (Adjacent Databases): Two databases $D = \{d_i\}_{i=1}^n$ and $D' = \{d_i'\}_{i=1}^n$ are said to be *adjacent* if there exists $i \in [n]$ such that $d_j = d_j'$ for all $j \neq i$.

A *mechanism* that acts on a database is said to be differentially private if it is able to ensure that two adjacent databases are nearly indistinguishable from the output of the mechanism.

Definition 2 (Differential Privacy [10]): Given $\epsilon \geq 0$, a mechanism M preserves ϵ-differential privacy if for all

$\mathcal{R} \subseteq \operatorname{range}(M)$ and all adjacent databases D and D' in $\mathcal{D}$, it holds that

$$\mathbb{P}\left(M(D) \in \mathcal{R}\right) \le e^{\epsilon} \mathbb{P}\left(M(D') \in \mathcal{R}\right). \tag{4}$$

The constant ϵ indicates the level of privacy: smaller ϵ implies higher level of privacy. The notion of differential privacy promises that an adversary cannot tell from the output of M with high probability whether data corresponding to a single user in the database have changed. It can be seen that any nonconstant differentially mechanism is necessarily *randomized*, i.e., for a given database, the output of such a mechanism obeys a certain probability distribution. Finally, although it is not explicitly mentioned in Definition 2, a mechanism needs to be an approximation of the query of interest in order to be useful. For this purpose, a mechanism is normally defined in conjunction with some query of interest; a common notation is to include the query q of interest in the subscript of the mechanism as M_q.

C. Problem Formulation: Differentially Private Distributed Constrained Optimization

Recall that our goal of preserving privacy in distributed optimization is to protect the user information in $\mathcal{C}_i$, even if an adversary can collect all public signals $\{p^{(k)}\}_{k=1}^K$. To mathematically formulate our goal under the framework of differential privacy, we define the database D as the set $\{\mathcal{C}_i\}_{i=1}^n$ and the query as the K-tuple consisting of all the gradients $p = (p^{(1)}, p^{(2)}, \ldots, p^{(K)})$. Without loss of generality, we consider the case where $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ belong to a family of sets parameterized by $\beta \in \mathbb{R}^s$. Namely, there exists a parameterized set $\mathcal{C}$ such that for all $i \in [n]$, we can write $\mathcal{C}_i = \mathcal{C}(\beta_i)$ for some $\beta_i \in \mathbb{R}^s$. We also assume that there exists a metric $\rho : \mathbb{R}^s \times \mathbb{R}^s \to \mathbb{R}_+$, so that we can define the distance $\rho_{\mathcal{C}}(\mathcal{C}_i, \mathcal{C}_i')$ between any $\mathcal{C}_i = \mathcal{C}(\beta_i)$ and $\mathcal{C}_i' = \mathcal{C}(\beta_i')$ using the metric ρ as

$$\rho_{\mathcal{C}}\left(\mathcal{C}_i, \mathcal{C}_i'\right) := \rho\left(\beta_i, \beta_i'\right).$$

For any given $\delta\mathcal{C} \in \mathbb{R}_+$, we define and use throughout the paper the following adjacency relation between any two databases D and D' in the context of distributed constrained optimization.

Definition 3 (Adjacency Relation for Constrained Optimization): For any databases $D = \{\mathcal{C}_i\}_{i=1}^n$ and $D' = \{\mathcal{C}_i'\}_{i=1}^n$, it holds that $\operatorname{Adj}(D, D')$ if and only if there exists $i \in [n]$ such that $\rho_{\mathcal{C}}(\mathcal{C}_i, \mathcal{C}_i') \le \delta\mathcal{C}$, and $\mathcal{C}_j = \mathcal{C}_j'$ for all $j \ne i$.

The constant $\delta\mathcal{C}$ is chosen based on the privacy requirement, i.e., the kind of user activities that should be kept private. Using the adjacency relation described in Definition 3, we state in the following the problem of designing a differentially private distributed algorithm for constrained optimization.

Problem 4 (Differentially Private Distributed Constrained Optimization): Find a randomized mechanism M_p that approximates the gradients $p = (p^{(1)}, p^{(2)}, \ldots, p^{(K)})$ (defined in Algorithm 1) and preserves ϵ-differential privacy under the adjacency relation described in Definition 3. Namely, for any adjacent databases D and D', and any $\mathcal{R} \subseteq \operatorname{range}(M_p)$, the mechanism M_p should satisfy

$$\mathbb{P}\left(M_p(D) \in \mathcal{R}\right) \le e^{\epsilon} \mathbb{P}\left(M_p(D') \in \mathcal{R}\right).$$

D. Example Application: EV Charging

In EV charging, the goal is to charge n vehicles over a horizon of T time steps with minimal influence on the power grid. For simplicity, we assume that each vehicle belongs to one single user. For any $i \in [n]$, the vector $r_i \in \mathbb{R}^T$ represents the charging rates of vehicle i over time. In the following, we will denote by $r_i(t)$ the tth component of r_i. Each vehicle needs to be charged a given amount of electricity $E_i > 0$ by the end of the scheduling horizon; in addition, for any $t \in [T]$, the charging rate $r_i(t)$ cannot exceed the maximum rate $\bar{r}_i(t)$ for some given constant vector $\bar{r}_i \in \mathbb{R}^T$. Under these constraints on r_i, the set $\mathcal{C}_i$ is described as follows:

$$0 \preceq r_i \preceq \bar{r}_i, \qquad \mathbf{1}^T r_i = E_i. \tag{5}$$

The tuple $(\bar{r}_i, E_i)$ is called the *charging specification* of user i. Throughout the paper, we assume that $\bar{r}_i$ and E_i satisfy

$$\mathbf{1}^T \bar{r}_i \ge E_i \quad \text{for all } i \in [n] \tag{6}$$

so that the constraints (5) are always feasible.

The objective function U in problem (2) quantifies the influence of a charging schedule $\{r_i\}_{i=1}^n$ on the power grid. We choose U as follows for the purpose of minimizing load variance:

$$U\left(\sum_{i=1}^n r_i\right) = \frac{1}{2}\left\| d + \sum_{i=1}^n \frac{r_i}{m} \right\|^2. \tag{7}$$

In (7), m is the number of households, which is assumed proportional to the number of EVs, i.e., there exists γ such that $n/m = \gamma$; then, the quantity $\sum_{i=1}^n r_i/m$ becomes the aggregate EV load per household. The vector $d \in \mathbb{R}^T$ is the base load profile (per household) incurred by loads in the power grid other than EVs, so that $U(\sum_{i=1}^n r_i)$ quantifies the variation of the total load including the base load and EVs. It can be verified that U is convex and differentiable, and ∇U is Lipschitz continuous.

The set $\mathcal{C}_i$ (defined by $\bar{r}_i$ and E_i) can be associated with personal activities of the owner of vehicle i in the following way. For example, $\bar{r}_i(t) = 0$ may indicate that the owner is temporarily away from the charging station (which may be colocated with the owner's residence) so that the vehicle is not ready to be charged. Similarly, $E_i = 0$ may indicate that the owner is not actively using the vehicle so that the vehicle does not need to be charged.

We now illustrate why publishing the exact gradient $p^{(k)}$ can potentially lead to a loss of privacy. The gradient $p^{(k)}$ can be computed as $p^{(k)} = (1/m)(d + \sum_{i=1}^n r_i/m)$. Recall that the goal of differential privacy is to provide a strong privacy guarantee in the presence of any auxiliary information that an adversary may have. In the worst case, an adversary may be able to collaborate with all but one user i and obtain $r_j^{(k)}$ for all $j \ne i$ in the database. Then, the adversary can infer $r_i^{(k)}$ exactly from $p^{(k)}$, even though user i did not reveal his $r_i^{(k)}$ to the adversary. After obtaining $r_i^{(k)}$, the adversary can obtain information on $\mathcal{C}_i$ by, for example, computing $E_i = \mathbf{1}^T r_i^{(k)}$.

The adjacency relation in the case of EV charging is defined as follows. Notice that, in the case of EV charging, the parameter β_i that parameterizes the set $\mathcal{C}_i$ is given by $\beta_i = (\bar{r}_i, E_i)$, in which $(\bar{r}_i, E_i)$ is the charging specifications of user i as defined in (5).

Definition 5 (Adjacency Relation for EV Charging): For any databases $D = \{\mathcal{C}_i(\bar{r}_i, E_i)\}_{i=1}^n$ and $D' = \{\mathcal{C}'_i(\bar{r}'_i, E'_i)\}_{i=1}^n$, we have $\mathrm{Adj}(D, D')$ if and only if there exists $i \in [n]$ such that

$$\|\bar{r}_i - \bar{r}'_i\|_1 \le \delta r, \qquad |E_i - E'_i| \le \delta E \tag{8}$$

and $\bar{r}_j = \bar{r}'_j$, $E_j = E'_j$ for all $j \neq i$.

In terms of choosing δE and δr, one useful choice for δE is the maximum amount of energy an EV may need; this choice of δE can be used to hide the event corresponding to whether a user needs to charge his vehicle.

IV. Overview of Main Results

A. Results for General Constrained Optimization Problems

In Section V, we present the main algorithmic result of this paper, a differentially private distributed algorithm (Algorithm 2) for solving the constrained optimization problem (2). The constant Δ that appears in the input of Algorithm 2 is defined as

$$\Delta := \max_{i \in [n]} \max \Big\{ \big\| \Pi_{\mathcal{C}_i}(r) - \Pi_{\mathcal{C}'_i}(r) \big\| : r \in \mathbb{R}^T,\ \mathcal{C}_i \text{ and } \mathcal{C}'_i \text{ satisfy } \rho_{\mathcal{C}}(\mathcal{C}_i, \mathcal{C}'_i) \le \delta\mathcal{C} \Big\}. \tag{9}$$

In other words, Δ can be viewed as a bound on the global ℓ_2-sensitivity of the projection operator $\Pi_{\mathcal{C}_i}$ to changes in $\mathcal{C}_i$ for all $i \in [n]$. Later, we will illustrate how to compute Δ using the case of EV charging.

Algorithm 2 Differentially private distributed projected gradient descent

```
Input: $U$, $L$, $\{\mathcal{C}_i\}_{i=1}^n$, $K$, $\{\alpha_k\}_{k=1}^K$, $\eta \ge 1$, $\Delta$, and $\epsilon$.
Output: $\{\hat{r}_i^{(K+1)}\}_{i=1}^n$.
Initialize $\{r_i^{(1)}\}_{i=1}^n$ arbitrarily. Let $\hat{r}_i^{(1)} = r_i^{(1)}$ for all $i \in [n]$ and $\theta_k = (\eta+1)/(\eta+k)$ for $k \in [K]$.
For $k = 1, 2, \ldots, K$, repeat:
    1) If $k = 1$, then set $w_k = 0$; else draw a random vector $w_k \in \mathbb{R}^T$ from the distribution (proportional to) $\exp(-(2\epsilon\|w_k\|/K(K-1)L\Delta))$
    2) Compute $\hat{p}^{(k)} := \nabla U(\sum_{i=1}^n r_i^{(k)}) + w_k$.
    3) For $i \in [n]$, compute:
        $r_i^{(k+1)} := \Pi_{\mathcal{C}_i}\left(r_i^{(k)} - \alpha_k \hat{p}^{(k)}\right)$
        $\hat{r}_i^{(k+1)} := (1-\theta_k)\hat{r}_i^{(k)} + \theta_k r_i^{(k+1)}$.
```

Compared to the (nonprivate) distributed algorithm (Algorithm 1), the key difference in Algorithm 2 is the introduction of random perturbations in the gradients (step 2) that convert $p^{(k)}$ into a noisy gradient $\hat{p}^{(k)}$. The noisy gradients $(\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(K)})$ can be viewed as a randomized mechanism M_p that approximates the original gradients $p = (p^{(1)}, p^{(2)}, \ldots, p^{(K)})$. In Section V, we will prove that the noisy gradients (as a mechanism) $M_p := (\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(K)})$ *preserve ϵ-differential privacy* and hence solve Problem 4.

Theorem 6: Algorithm 2 ensures that $M_p := (\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(K)})$ preserves ϵ-differential privacy under the adjacency relation given by Definition 3.

Algorithm 2 can be viewed as an instance of stochastic gradient descent that terminates after K iterations. We will henceforth refer to Algorithm 2 as *differentially private distributed projected gradient descent*. The step size α_k is chosen as $\alpha_k = c/\sqrt{k}$ for some $c > 0$. The purpose of the additional variables $\{\hat{r}_i^{(k)}\}_{k=1}^K$ is to implement the polynomial-decay averaging method in order to improve the convergence rate, which is a common practice in stochastic gradient descent [27]; introducing $\{\hat{r}_i^{(k)}\}_{k=1}^K$ does not affect privacy. The parameter $\eta \ge 1$ is used for controlling the averaging weight θ_k. Details on choosing η can be found in Shamir and Zhang [27].

Like most iterative optimization algorithms, stochastic gradient descent only converges in a probabilistic sense as the number of iterations $K \to \infty$. In practice, the number of iterations is always finite, so that it is desirable to analyze the suboptimality for a finite K. In Section V, we provide an analysis on the expected suboptimality of Algorithm 2.

Theorem 7: The expected suboptimality of Algorithm 2 after K iterations is bounded as follows:

$$\mathbb{E}\left[U\left(\sum_{i=1}^n \hat{r}_i^{(K+1)} \right) - U^* \right] \le \mathcal{O}\left(\eta\sqrt{n}\rho \left(\frac{G}{\sqrt{K}} + \frac{\sqrt{2}TK^{\frac{3}{2}}L\Delta}{2\epsilon} \right) \right) \tag{10}$$

where U^* is the optimal value of problem (2), and

$$\rho = \max\left\{ \sqrt{\sum_{i=1}^n \|r_i\|^2} : r_i \in \mathcal{C}_i,\ i \in [n] \right\}$$

$$G = \max\left\{ \left\| \nabla U\left(\sum_{i=1}^n r_i \right) \right\| : r_i \in \mathcal{C}_i,\ i \in [n] \right\}.$$

Theorem 7 reveals an important trade-off in choosing the number of iterations K when running the differentially private optimization algorithm (Algorithm 2). If K is too small, then it will affect the convergence of gradient descent. On the other hand, if K is too large, then the amount of noise required by differential privacy will be too large and affect convergence as well.

B. Results for the Case of EV Charging

Having presented and analyzed the algorithm for a general distributed constrained optimization problem, we then illustrate how Algorithm 2 can be applied to the case of EV charging in Section VI. In particular, we demonstrate how to compute Δ in the case of EV charging. Theorem 8 shows that Δ can be bounded by δr and δE that appear in (8).

Theorem 8: Consider the example of EV charging (as described in Section III-D). For any $i \in [n]$, the global ℓ_2-sensitivity of the projection operator $\Pi_{\mathcal{C}_i(\bar{r}_i, E_i)}$ with respect to changes in $(\bar{r}_i, E_i)$ is bounded by

$$\Delta \le 2\delta r + \delta E$$

where δr and δE are specified in the adjacency relation given by (8).

The suboptimality analysis given in Theorem 7 can be further refined in the case of EV charging. The special form of U given by (7) allows obtaining an upper bound on suboptimality as given in Corollary 9 below.

Corollary 9: For the cost function U given by (7), the expected suboptimality of Algorithm 2 is bounded as follows:

$$\mathbb{E}\left[U\left(\sum_{i=1}^{n} \hat{r}_i^{(K+1)}\right) - U^*\right] \le \mathcal{O}\left(\eta T^{\frac{1}{4}} \left(\frac{\Delta}{n\epsilon}\right)^{\frac{1}{4}}\right). \qquad (11)$$

This upper bound shows the trade-off between privacy and performance. As ϵ decreases, more privacy is preserved but at the expense of increased suboptimality. On the other hand, this increase in suboptimality can be mitigated by introducing more participating users (i.e., by increasing n), which coincides with the common intuition that it is easier to achieve privacy as the number of users n increases.

V. Differentially Private Distributed Projected Gradient Descent

In this section, we give the proof that the modified distributed projected gradient descent algorithm (Algorithm 2) preserves ϵ-differential privacy. In the proof, we will extensively use results from differential privacy such as the Laplace mechanism and the adaptive sequential composition theorem.

A. Review: Results From Differential Privacy

The introduction of additive noise in step 2 of Algorithm 2 is based on a variant of the widely used Laplace mechanism in differential privacy. The Laplace mechanism operates by introducing additive noise according to the ℓ_p-sensitivity $(p \ge 1)$ of a numerical query $q : \mathcal{D} \to \mathbb{R}^m$ (for some dimension m), which is defined as follows.

Definition 10 (ℓ_p-Sensitivity): For any query $q : \mathcal{D} \to \mathbb{R}^m$, the ℓ_p-*sensitivty* of q under the adjacency relation Adj is defined as

$$\Delta_q := \max\left\{\|q(D) - q(D')\|_p : D, D' \in \mathcal{D} \text{ s.t. } \mathrm{Adj}(D, D')\right\}.$$

Note that the ℓ_p-sensitivity of q does not depend on a specific database D. In this paper, we will use the Laplace mechanism for bounded ℓ_2-sensitivity.

Proposition 11 (Laplace Mechanism [10]): Consider a query $q : \mathcal{D} \to \mathbb{R}^m$ whose ℓ_2-sensitivity is Δ_q. Define the mechanism M_q as $M_q(D) := q(D) + w$, where w is an m-dimensional random vector whose probability density function is given by $p_w(w) \propto \exp(-\epsilon\|w\|/\Delta_q)$. Then, the mechanism M_q preserves ϵ-differential privacy.

As a basic building block in differential privacy, the Laplace mechanism allows construction of the differentially private distributed projected gradient descent algorithm described in Algorithm 2 through *adaptive sequential composition*.

Proposition 12 (Adaptive Seqential Composition [11]): Consider a sequence of mechanisms $\{M_k\}_{k=1}^{K}$, in which the output of M_k may depend on $M_1, M_2, \ldots, M_{k-1}$ as described below:

$$M_k(D) = M_k\left(D, M_1(D), M_2(D), \ldots, M_{k-1}(D)\right).$$

Suppose $M_k(\cdot, a_1, a_2, \ldots, a_{k-1})$ preserves ϵ_k-differential privacy for any $a_1 \in \mathrm{range}(M_1), \ldots, a_{k-1} \in \mathrm{range}(M_{k-1})$. Then, the K-tuple mechanism $M := (M_1, M_2, \ldots, M_K)$ preserves ϵ-differential privacy for $\epsilon = \sum_{k=1}^{K} \epsilon_k$.

B. Proof That Algorithm 2 Preserves ϵ-Differential Privacy

Using the adaptive sequential composition theorem, we can show that Algorithm 2 preserves ϵ-differential privacy. We can view the K-tuple mechanism $M_p := (\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(K)})$ as a sequence of mechanisms $\{\hat{p}^{(k)}\}_{k=1}^{K}$. The key is to compute the ℓ_2-sensitivity of $p^{(k)} := \nabla U(\sum_{i=1}^{n} r_i^{(k)})$, denoted by $\Delta^{(k)}$, when the outputs of $\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(k-1)}$ are given, so that we can obtain a differentially private mechanism $\hat{p}^{(k)}$ by applying the Laplace mechanism on $p^{(k)}$ according to $\Delta^{(k)}$.

Lemma 13: In Algorithm 2, when the outputs of $\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(k-1)}$ are given, the ℓ_2-sensitivity of $p^{(k)} := \nabla U(\sum_{i=1}^{n} r_i^{(k)})$ satisfies $\Delta^{(k)} = (k-1)L\Delta$.

Proof: See Appendix I. ■

With Lemma 13 at hand, we now show that Algorithm 2 preserves ϵ-differential privacy (Theorem 6, Section IV).

Proof (of Theorem 6): For any $k \in [K]$, when the outputs of $\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(k-1)}$ are given, we know from Proposition 11 that $\hat{p}^{(k)}$ preserves ϵ_k-differential privacy, where ϵ_k satisfies $\epsilon_1 = 0$ and for $k > 1$

$$\frac{\epsilon_k}{\Delta^{(k)}} = \frac{2\epsilon}{K(K-1)L\Delta}.$$

Use the expression of $\Delta^{(k)}$ from Lemma 13 to obtain

$$\epsilon_k = \frac{2(k-1)\epsilon}{K(K-1)}.$$

Using the adaptive sequential composition theorem, we know that the privacy of $M_p := (\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(K)})$ is given by $\sum_{k=1}^{K} \epsilon_k = (2\epsilon/K(K-1)) \sum_{k=1}^{K}(k-1) = \epsilon$, which completes the proof. ■

C. Suboptimality Analysis: Privacy–Performance Trade-Off

As a consequence of preserving privacy, we only have access to noisy gradients $\{\hat{p}^{(k)}\}_{k=1}^{K}$ rather than the exact gradients $\{p^{(k)}\}_{k=1}^{K}$. Recall that the additive noise w_k in step 2 of Algorithm 2 has zero mean. In other words, the noisy gradient $\hat{p}^{(k)}$ is an unbiased estimate of $p^{(k)}$, which allows us to view Algorithm 2 as an instantiation of stochastic gradient descent. As we mentioned in Section IV, it is in fact a variant of stochastic gradient descent that uses polynomial-decay averaging for better convergence. The stochastic gradient descent method (with polynomial-decay averaging), which is described in Algorithm 3, can be used for solving the following optimization problem:

$$\min_{x}. \quad f(x) \qquad \text{s.t.} \quad x \in \mathcal{X}$$

where $x \in \mathbb{R}^m$ and $\mathcal{X} \subset \mathbb{R}^m$ for certain dimensions m. Proposition 14 (due to Shamir and Zhang [27]) gives an upper bound of the expected suboptimality after finitely many steps for the stochastic gradient descent algorithm (Algorithm 3).

Algorithm 3 Stochastic gradient descent with polynomial-decay averaging

```
Input: $f$, $\mathcal{X}$, $K$, $\{\alpha_k\}_{k=1}^{K}$, and $\eta \geq 1$.
Output: $\hat{x}^{(K+1)}$.
Initialize $x^{(1)}$ and $k = 1$. Let $\hat{x}^{(1)} = x^{(1)}$ and $\theta_k = (\eta + 1)/(\eta + k)$ for $k \in [K]$.
For $k = 1, 2, \dots, K$, repeat:
    1) Compute an unbiased subgradient $\hat{g}_k$ of $f$ at $x^{(k)}$, i.e.,
       $\mathbb{E}[\hat{g}_k] \in \partial f(x^{(k)})$.
    2) Update $x^{(k+1)} := \Pi_{\mathcal{X}}(x^{(k)} - \alpha_k \hat{g}_k)$ and $\hat{x}^{(k+1)} :=$
       $(1 - \theta_k)\hat{x}^{(k)} + \theta_k x^{(k+1)}$.
```

Proposition 14 (Shamir and Zhang [27]): Suppose $\mathcal{X} \subset \mathbb{R}^m$ is a convex set and $f : \mathbb{R}^m \to \mathbb{R}$ is a convex function. Assume that there exist ρ and $\widehat{G}$ such that $\sup_{x,x' \in \mathcal{X}} \|x - x'\| \leq \rho$ and $\max_{1 \leq k \leq K} \mathbb{E}\|\hat{g}_k\|^2 \leq \widehat{G}^2$ for $\{\hat{g}_k\}_{k=1}^{K}$ given by step 1 of Algorithm 3. If the step sizes are chosen as $\alpha_k = c/\sqrt{k}$ for some $c > 0$, then for any $K > 1$, it holds that

$$\mathbb{E}\left(f(\hat{x}^{(K+1)}) - f^*\right) \leq \mathcal{O}\left(\frac{\eta(\rho^2/c + c\widehat{G}^2)}{\sqrt{K}}\right) \tag{12}$$

where $f^* = \inf_{x \in \mathcal{X}} f(x)$.

A tighter upper bound can be obtained from (12) by optimizing the right-hand side of (12) over the constant c.

Corollary 15: Under the same setting as Proposition 14, the suboptimality bound for Algorithm 3 is given by

$$\mathbb{E}\left(f(\hat{x}^{(K+1)}) - f^*\right) \leq \mathcal{O}\left(\eta\frac{\rho\widehat{G}}{\sqrt{K}}\right) \tag{13}$$

which is achieved by choosing $c = \rho/\widehat{G}$.

By applying Corollary 15, we are able obtain the bound of suboptimality for Algorithm 2 as given by Theorem 7.

Proof (of Theorem 7): In order to apply Corollary 15, we need to compute ρ and $\widehat{G}$ for Algorithm 2. The constant ρ can be obtained as

$$\rho = \max\left\{\sqrt{\sum_{i=1}^{n} \|r_i\|^2} : r_i \in \mathcal{C}_i,\ i \in [n]\right\}.$$

Recall from Section II-B that, for all $i \in [n]$, the gradient of the objective function with respect to $r_i^{(k)}$ is identical, and $\nabla_{r_i^{(k)}} U(\sum_{i=1}^{n} r_i^{(k)}) = \nabla U(\sum_{i=1}^{n} r_i^{(k)})$. As a consequence, an unbiased stochastic gradient $\hat{g}_k$ of the objective function $U(\sum_{i=1}^{n} r_i^{(k)})$ with respect to $(r_1^{(k)}, r_2^{(k)}, \dots, r_n^{(k)})$ is given by $\hat{g}_k = [\hat{p}^{(k)}, \hat{p}^{(k)}, \dots, \hat{p}^{(k)}]$, which is formed by repeating $\hat{p}^{(k)}$ for n times. Using the definition $\widehat{G}^2 := \max_k \mathbb{E}\|\hat{g}_k\|^2$, we have $\widehat{G}^2 = n \cdot \max_k \mathbb{E}\|\hat{p}^{(k)}\|^2$. Substituting the expression of $\hat{p}^{(k)}$ into $\widehat{G}$, we have

$$\begin{aligned}\widehat{G} &= \sqrt{n} \cdot \max_{k \in [K]} \sqrt{\|p^{(k)}\|^2 + \mathbb{E}\|w_k\|^2}\\ &\leq \sqrt{n} \cdot \max_{k \in [K]} \left\{\|p^{(k)}\| + \sqrt{\mathbb{E}\|w_k\|^2}\right\}\\ &\leq \sqrt{n}\left(G + \sqrt{2}TK^2L\Delta/2\epsilon\right)\end{aligned}$$

where in the last step we have used the fact that

$$\begin{aligned}\mathbb{E}\|w_k\|^2 &= \text{var}\|w_k\|^2 + (\mathbb{E}\|w_k\|)^2\\ &= T\left(\frac{\Delta^{(k)}}{\epsilon_k}\right)^2 + T^2\left(\frac{\Delta^{(k)}}{\epsilon_k}\right)^2\\ &\leq 2T^2\left(\frac{\Delta^{(k)}}{\epsilon_k}\right)^2\\ \frac{\Delta^{(k)}}{\epsilon_k} &= \frac{K(K-1)L\Delta}{2\epsilon} \leq \frac{K^2L\Delta}{2\epsilon}.\end{aligned}$$

Substitute the expression of $\widehat{G}$ into (13) to obtain the result. ■

As K increases, the first term in (10) decreases, whereas the second term in (10) increases. It is then foreseeable that there exists an optimal choice of K that minimizes the expected suboptimality.

Corollary 16: By choosing $K = (\sqrt{2}G\epsilon/3TL\Delta)^{1/2}$, the expected suboptimality of Algorithm 2 is bounded as follows:

$$\mathbb{E}\left[U\left(\sum_{i=1}^{n} \hat{r}_i^{(K+1)}\right) - U^*\right] \leq \mathcal{O}\left(\eta T^{\frac{1}{4}} n^{\frac{1}{2}} \rho (G^3 L\Delta/\epsilon)^{\frac{1}{4}}\right) \tag{14}$$

where U^*, ρ, and G are given by Theorem 7.

Proof: The result can be obtained by optimizing the right-hand side of (10) over K. ■

However, since it is generally impossible to obtain a tight bound for ρ and $\widehat{G}$, optimizing K according to Corollary 16 usually does not give the best K in practice; numerical simulation is often needed in order to find the best K for a given problem. We will demonstrate how to choose K optimally later using numerical simulations in Section VII.

VI. Sensitivity Computation: The Case of EV Charging

So far, we have shown that Algorithm 2 (specifically, the mechanism M_p consisting of the gradients $(\hat{p}^{(1)}, \hat{p}^{(2)}, \dots, \hat{p}^{(K)})$ that are broadcast to every participating user) preserves ϵ-differential privacy. The magnitude of the noise w_k introduced to the gradients depends on Δ, which is the sensitivity of the projection operator $\Pi_{\mathcal{C}_i}$ as defined in (9). In order to implement Algorithm 2, we need to compute Δ explicitly. In the next, we will illustrate how to compute Δ using the case of EV charging as an example. We will give an expression for

Δ that depends on the constants δr and δE appearing in the adjacency relation (8). Since δr and δE are part of the privacy requirement, one can choose Δ accordingly once the privacy requirement has been determined.

A. Overview

The input of Algorithm 2 includes the constant Δ as described by (9), which bounds the global ℓ_2-sensitivity of the projection operator $\Pi_{\mathcal{C}_i(\bar{r}_i, E_i)}$ with respect to changes in $(\bar{r}_i, E_i)$. In this section, we will derive an explicit expression of Δ for the case of EV charging. Using tools in sensitivity analysis of optimization problems, we are able to establish the relationship between Δ and the constants δr and δE that appear in the adjacency relation (8) used in EV charging.

Recall that for any $r \in \mathbb{R}^T$, the output of the projection operation $\Pi_{\mathcal{C}_i(\bar{r}_i, E_i)}(r)$ is the optimal solution to the constrained least-squares problem

$$\begin{aligned} \min_{r_i}.\quad & \frac{1}{2}\|r_i - r\|^2 \\ \text{s.t.}\quad & 0 \preceq r_i \preceq \bar{r}_i, \qquad \mathbf{1}^T r_i = E_i. \end{aligned} \tag{15}$$

Define the ℓ_2-sensitivity for a fixed r as

$$\Delta_r := \max_{i\in[n]} \max \left\{ \left\| \Pi_{\mathcal{C}_i(\bar{r}_i . E_i)}(r) - \Pi_{\mathcal{C}_i(\bar{r}_i', E_i')}(r) \right\| : \right. \\ \left. (\bar{r}_i, E_i) \text{ and } (\bar{r}_i', E_i') \text{ satisfy (8)} \right\}.$$

It can be verified that $\Delta = \max_{r\in\mathbb{R}^T} \Delta_r$. In the following, we will establish the relationship between Δ_r and $(\delta r, \delta E)$; we will also show that Δ_r does not depend on the choice of r, so that $\Delta = \Delta_r$ for any $r \in \mathbb{R}^T$. For notational convenience, we consider the following least-squares problem:

$$\begin{aligned} \min_{x}.\quad & \frac{1}{2}\|x - x_0\|^2 \\ \text{s.t.}\quad & 0 \preceq x \preceq a, \qquad \mathbf{1}^T x = b \end{aligned} \tag{16}$$

where x_0, a, and b are given constants. If we let

$$x_0 = r_0, \qquad a = \bar{r}_i, \qquad b = E_i$$

then problem (16) is mapped back to problem (15). We also have $\mathbf{1}^T a \geq b$ based on the assumption as described in (6). Denote the optimal solution of problem (16) by $x^*(a, b)$. Since our purpose is to derive an expression for Δ_r when r is fixed, we also treat x_0 as fixed and have dropped the dependence of x^* on x_0. Our goal is to bound the global solution sensitivity with respect to changes in a and b, i.e.,

$$\|x^*(a', b') - x^*(a, b)\| \tag{17}$$

for any (a, b) and (a', b'). We will proceed by first bounding the *local solution sensitivities* $\partial_a x^*$ and $\partial_b x^*$ with respect to a and b. Then, we will obtain a bound on the global solution, sensitivity (17) through integration of $\partial_a x^*$ and $\partial_b x^*$.

B. Local Solution Sensitivity of Nonlinear Optimization Problems

We begin by reviewing existing results on computing local solution sensitivity of nonlinear optimization problems. Consider a generic nonlinear optimization problem parametrized by $\theta \in \mathbb{R}$ described as follows:

$$\begin{aligned} \min_{x\in\mathbb{R}^n}.\quad & f(x;\theta) \\ \text{s.t.}\quad & g_i(x;\theta) \leq 0, \quad i \in [p] \\ & h_j(x;\theta) = 0, \quad j \in [q] \end{aligned} \tag{18}$$

whose Lagrangian can be expressed as

$$L(x, \lambda, \nu; \theta) = f(x;\theta) + \sum_{i=1}^{p} \lambda_i g_i(x;\theta) + \sum_{j=1}^{q} \nu_j h_j(x;\theta)$$

where λ and ν are the Lagrange multipliers associated with constraints $\{g_i\}_{i=1}^p$ and $\{h_j\}_{j=1}^q$, respectively. If there exists a set $\Theta \subset \mathbb{R}$ such that the optimal solution is unique for all $\theta \in \Theta$, then the optimal solution of problem (18) can be defined as a function $x^* : \Theta \to \mathbb{R}^n$. This condition on the uniqueness of optimal solution holds for problem (16), since the objective function therein is strictly convex.

Denote by λ^* and ν^* the optimal Lagrange multipliers. Under certain conditions described in Proposition 17 below, the partial derivatives $\partial_\theta x^*$, $\partial_\theta \lambda^*$, and $\partial_\theta \nu^*$ exist; these partial derivatives will also be referred to as *local solution sensitivity* of problem (18).

Proposition 17 (Fiacco [12])*:* Let (x^*, λ^*, ν^*) be the primal-dual optimal solution of problem (18). Suppose the following conditions hold.

1) x^* is a locally unique optimal primal solution.
2) The functions f, $\{g_i\}_{i=1}^p$, and $\{h_j\}_{j=1}^q$ are twice continuously differentiable in x and differentiable in θ.
3) The gradients $\{\nabla g_i(x^*) : g_i(x^*) = 0,\ i \in [p]\}$ of the active constraints and the gradients $\{\nabla h_j(x^*) : j \in [q]\}$ are linearly independent.
4) Strict complementary slackness holds: $\lambda_i^* > 0$ when $g_i(x^*, \theta) = 0$ for all $i \in [p]$.

Then the local sensitivity $(\partial_\theta x^*, \partial_\theta \lambda^*, \partial_\theta \nu^*)$ of problem (18) exists and is continuous in a neighborhood of θ. Moreover, $(\partial_\theta x^*, \partial_\theta \lambda^*, \partial_\theta \nu^*)$ is uniquely determined by the following:

$$\begin{aligned} &\nabla^2 L \cdot \partial_\theta x^* + \sum_{i=1}^{p} \nabla g_i \cdot \partial_\theta \lambda_i^* + \sum_{j=1}^{q} \nabla h_j \cdot \partial_\theta \nu_j^* + \partial_\theta(\nabla L) = 0 \\ &\lambda_i \nabla g_i \cdot \partial_\theta x^* + g_i \partial_\theta \lambda_i^* + \lambda_i^* \partial_\theta g_i = 0, \quad i \in [p] \\ &\nabla h_j \cdot \partial_\theta x^* + \partial_\theta h_j = 0, \quad j \in [q]. \end{aligned}$$

C. Solution Sensitivity of the Distributed EV Charging Problem

We begin by computing the local solution sensitivities $\partial_a x^*$ and $\partial_b x^*$ for problem (16) using Proposition 17. After $\partial_a x^*$ and $\partial_b x^*$ are obtained, the global solution sensitivity (17) can be obtained through integration of the local sensitivity. For convenience, we compute the global solution sensitivity in a and b separately and combine the results in the end in the proof of Theorem 8.

One major difficulty in applying Proposition 17 is that it requires strict complementary slackness, which unfortunately does not hold for all values of a and b. We will proceed by deriving the local solution sensitivities assuming that strict complementary slackness holds. Later, we will show that strict complementary slackness only fails at finitely many locations on the integration path, so that the integral remains unaffected.

We shall proceed by computing the solution sensitivity for a and b separately. First of all, we assume that a is fixed and solve for the global solution sensitivity of x^* in b, defined as

$$\|x^*(a,b') - x^*(a,b)\| \, . \tag{19}$$

When strict complementary slackness holds, the following lemma gives properties of the local solution sensitivity of problem (16) with respect to b.

Lemma 18 (Local Solution Sensitivity in b): When strict complementary slackness holds, the local solution sensitivity $\partial_b x^*$ of problem (16) satisfies

$$\partial_b x^* \succeq 0 \qquad \text{and} \qquad \mathbf{1}^T \partial_b x^* = 1.$$

Proof: See Appendix II-A. ■

The following lemma shows that the condition is only violated for a finite number of values of b, so that it will still be possible to obtain the global sensitivity (19) through integration.

Lemma 19: The set of possible values of b in problem (16) for which strict complementary slackness fails to hold is finite.

Proof: See Appendix II-B. ■

The implication of Lemma 19 is that the local solution sensitivity $\partial_b x^*$ exists everywhere except at finitely many location. It is also possible to show that the optimal solution $x^*(a,b)$ is continuous in b (cf. Berge [2, p. 116], Dantzig *et al.* [6], or Tropp *et al.* [31, App. I]); the continuity of $x^*(a,b)$ in b and together with Lemma 19 imply that $\partial_b x^*$ is Riemann integrable so that we can obtain the global solution sensitivity through integration.

Proposition 20 (Global Solution Sensitivity in b): For any a, b, and b' that satisfy $\mathbf{1}^T a \geq b$ and $\mathbf{1}^T a \geq b'$, we have

$$\|x^*(a,b') - x^*(a,b)\|_1 = |b' - b|.$$

Proof: Without loss of generality, assume $b' > b$. Since $x^*(a,b)$ is continuous in b, and the partial derivative $\partial_b x^*$ exists except at finitely many points according to Lemma 19, we know that $\partial_b x_i^*$ is Riemann integrable for all $i \in [T]$, so that

$$x_i^*(a,b') - x_i^*(a,b) = \int_b^{b'} \partial_b x_i^*(a,b)\, db$$

according to the fundamental theorem of calculus. Using the fact $\partial_b x_i^* \geq 0$ as given by Lemma 18, we have

$$x_i^*(a,b') - x_i^*(a,b) \geq 0.$$

Then, we have

$$\begin{aligned}\|x^*(a,b') - x^*(a,b)\|_1 &= \sum_{i=1}^T |x_i^*(a,b') - x_i^*(a,b)| \\ &= \sum_{i=1}^T (x_i^*(a,b') - x_i^*(a,b)) \\ &= \int_b^{b'} \mathbf{1}^T \partial_b x^*(a,b)\, db.\end{aligned}$$

Using the fact $\mathbf{1}^T \partial_b x^* = 1$ given by Lemma 18, we obtain

$$\|x^*(a,b') - x^*(a,b)\|_1 = b' - b = |b' - b|.$$

■

Having computed the solution sensitivity in b, we now assume that b is fixed and solve for the global solution sensitivity of x^* in a, defined as

$$\|x^*(a',b) - x^*(a,b)\| \, . \tag{20}$$

When strict complementary slackness holds, the following lemma gives properties of the local solution sensitivity of problem (16) with respect to a.

Lemma 21 (Local Solution Sensitivity in a): When strict complementary slackness holds, the local solution sensitivity $\partial_a x^*$ of problem (16) satisfies

$$\sum_{i=1}^T \|\partial_{a_i} x^*\|_1 \leq 2.$$

Proof: See Appendix II-C. ■

Similar to computing the sensitivity in b, we can obtain the global solution sensitivity in a by integration of the local sensitivity. For convenience, we choose the integration path L from any a to a' such that only one component of a is varied at a time. Namely, the path L is given by

$$\begin{aligned}L : (a_1, a_2, \ldots, a_T) &\to (a_1', a_2, \ldots, a_T) \to \cdots \\ &\to (a_1', a_2', \ldots, a_T') \, . \end{aligned} \tag{21}$$

For convenience, we also define the subpaths $L_1, L_2, \ldots, L_T$ such that

$$L_i : (a_1', \ldots, a_{i-1}', a_i, \ldots, a_T) \to (a_1', \ldots, a_{i-1}', a_i', \ldots, a_T). \tag{22}$$

It is also possible to establish the fact that $\partial_a x^*$ exists excepts for a finite number of locations along the integration path L. Note that we do not need to check whether strict complementary slackness holds for the constraint $x_i \geq 0$, since in this case $\partial_{a_i} x^*$ always exists (in fact, $\partial_{a_i} x^* = 0$); instead, we only need to check strict complementary slackness

$$\mu_i^* > 0 \quad \text{when} \quad x_i^* = a_i \quad \text{for all } i \tag{23}$$

associated with the constraint $x \preceq a$.

Lemma 22: When constrained on the integration path L given by (21), the set of possible values of a in problem (16) for which the strict complementary slackness condition (23) fails to hold is finite.

Proof: See Appendix II-D. ■

Lemma 22 guarantees that $\partial_a x^*$ is Riemann integrable along L, so that we can obtain the global solution sensitivity (20) through integration.

Proposition 23 (Global Solution Sensitivity in a): For any given a, a', and b that satisfy $\mathbf{1}^T a \geq b$ and $\mathbf{1}^T a' \geq b$, we have

$$\|x^*(a',b) - x^*(a,b)\|_1 \leq 2\|a' - a\|_1.$$

Proof: Similar to the proof of Proposition 20, we can show that $\partial_a x_i^*$ is Riemann integrable using both Lemma 22 and the fact that x^* is continuous in a. Then, we can define

$$I_{ij} := \int_{L_j} \partial_a x_i^*(a,b) \cdot d\ell$$

which is the line integral of the vector field $\partial_{a_j} x_i^*(\cdot, b)$ along the path L_j. Define $x_{ij}^a := \partial_{a_j} x_i^*(\cdot, b)$. Using the definition of L_j as given by (22), we can write I_{ij} as

$$I_{ij} = \int_{a_j}^{a_j'} x_{ij}^a \left(a_1', a_2', \ldots, a_{j-1}', a_j, \ldots, a_T\right)\, da_j.$$

Then, we have

$$x_i^*(a', b) - x_i^*(a, b) = \int_L \partial_a x_i^*(a,b) \cdot d\ell = \sum_{j=1}^{T} I_{ij}$$

and consequently

$$|x_i^*(a', b) - x_i^*(a, b)| \le \sum_{j=1}^{T} |I_{ij}|.$$

Substituting the expression of $|x_i^*(a', b) - x_i^*(a, b)|$ into the the global sensitivity expression (20), we obtain

$$\begin{aligned} \|x^*(a', b) - x^*(a, b)\|_1 &= \sum_{i=1}^{T} |x_i^*(a', b) - x_i^*(a, b)| \\ &\le \sum_{i=1}^{T}\sum_{j=1}^{T} |I_{ij}| = \sum_{j=1}^{T}\sum_{i=1}^{T} |I_{ij}|. \end{aligned} \tag{24}$$

Note that we have

$$|I_{ij}| \le \int_{\underline{a}_j}^{\bar{a}_j} \left| x_{ij}^a \left(a_1', \ldots, a_{j-1}', a_j, \ldots, a_T\right)\right| \, da_j$$

where $\underline{a}_j := \min(a_j, a_j')$ and $\bar{a}_j := \max(a_j, a_j')$, so that

$$\begin{aligned} \sum_{i=1}^{T} |I_{ij}| &\le \sum_{i=1}^{T} \int_{\underline{a}_j}^{\bar{a}_j} \left| x_{ij}^a \left(a_1', \ldots, a_{j-1}', a_j, \ldots, a_T\right)\right| \, da_j \\ &= \int_{\underline{a}_j}^{\bar{a}_j} \sum_{i=1}^{T} \left| x_{ij}^a \left(a_1', \ldots, a_{j-1}', a_j, \ldots, a_T\right)\right| \, da_j \\ &= \int_{\underline{a}_j}^{\bar{a}_j} \left\| \partial_{a_j} x^* \left(\left(a_1', \ldots, a_{j-1}', a_j, \ldots, a_T\right), b\right)\right\|_1 \, da_j \\ &:= \bar{I}_j. \end{aligned}$$

Using Lemma 21, we can show that $\bar{I}_j$ satisfies

$$\bar{I}_j \le \int_{\underline{a}_j}^{\bar{a}_j} 2\, da_j = 2\left|a_j' - a_j\right|.$$

Substitute the above into (24) to obtain

$$\|x^*(a', b) - x^*(a, b)\|_1 \le 2 \sum_{j=1}^{T} \left|a_j' - a_j\right| = 2\|a' - a\|_1$$

which completes the proof. ■

We are now ready to prove Theorem 8 using results from Propositions 20 and 23.

Proof (of Theorem 8): Consider problem (16). By combining Propositions 20 and 23, we can obtain the global solution sensitivity with respect to both a and b as defined by (17). Consider any given (a, b) and (a', b') that satisfy $\mathbf{1}^T a \ge b$ and $\mathbf{1}^T a' \ge b'$. Without loss of generality, we assume that $\mathbf{1}^T a' \ge \mathbf{1}^T a$, so that $\mathbf{1}^T a' \ge b$; this implies that the corresponding optimization problem (16) is feasible, and the optimal solution $x^*(a', b)$ is well-defined. Then, we have

$$\begin{aligned} &\|x^*(a', b') - x^*(a, b)\|_1 \\ &\quad = \|x^*(a', b') - x^*(a', b) + x^*(a', b) - x^*(a, b)\|_1 \\ &\quad \le \|x^*(a', b') - x^*(a', b)\|_1 + \|x^*(a', b) - x^*(a, b)\|_1 \\ &\quad \le \|b' - b\|_1 + 2\|a' - a\|_1. \end{aligned} \tag{25}$$

By letting $x_0 = r_0$, $a = \bar{r}_i$, and $b = E_i$, we can map problem (16) back to problem (15). Recall that x^* is defined as the optimal solution of problem (16) for a given x_0. Then, the inequality (25) implies that for any given r, the ℓ_2-sensitivity

$$\Delta_r \le 2\|\bar{r}_i' - \bar{r}_i\|_1 + \|E_i' - E_i\|_1 = 2\delta r + \delta E.$$

However, since the right-hand side of the above inequality does not depend on r, we have

$$\Delta = \max_{r \in \mathbb{R}^T} \Delta_r \le 2\delta r + \delta E$$

which completes the proof. ■

Remark 24: Alternatively, one may use the fact $\Pi_{\mathcal{C}_i}(\cdot) \in \mathcal{C}_i$ to obtain a bound on Δ. Recall that for any $r_i \in \mathcal{C}_i$ we have

$$\|r_i\| \le \|\bar{r}_i\| \quad \text{and} \quad \|r_i\| \le \|\hat{r}\|_1 = E_i.$$

Then, we have

$$\Delta \le \max_{i \in [n]} \left(\|\bar{r}_i\| + \|\bar{r}_i'\|\right) \le 2 \max_{i \in [n]} \|\bar{r}_i\| + \delta r$$

$$\Delta \le \max_{i \in [n]} \left(E_i + E_i'\right) \le 2 \max_{i \in [n]} E_i + \delta E.$$

However, this bound can be quite loose in practice. Since the magnitude of w_k in Algorithm 2 is proportional to Δ, a loose bound on Δ implies introducing more noise to the gradient $p^{(k)}$ than what is necessary for preserving ϵ-differential privacy. As we have already seen in Section VI-D, the noise magnitude is closely related to the performance loss of Algorithm 2 caused by preserving privacy, and less noise is always desired for minimizing such a loss.

D. Revisited: Suboptimality Analysis

The bound (14) given by Corollary 16 does not clearly indicate the dependence of suboptimality on the number of participating users n, because ρ, G, and L also depend on n. In order to reveal the dependence on n, we will further refine the suboptimality bound (14) for the specific objective function U given in (7). The resulting suboptimality bound is shown in Corollary 9.

Proof (of Corollary 9): Define $r_{\max} = \max_{i\in[n]} \|r_i\|$. Then, we have $\rho \le \sqrt{n} r_{\max}$. For U given by (7), its gradient can be computed as

$$\nabla U\left(\sum_{i=1}^{n} r_i\right) = \frac{1}{m}\left(d + \sum_{i=1}^{n} \frac{r_i}{m}\right)$$

so that

$$\left\|\nabla U\left(\sum_{i=1}^{n} r_i\right)\right\| \le \frac{1}{m}\left(\|d\| + \sum_{i=1}^{n} \|r_i\|/m\right).$$

Then we obtain

$$\begin{aligned} G &:= \max\left\{\left\|\nabla U\left(\sum_{i=1}^{n} r_i\right)\right\| : r_i \in \mathcal{C}_i,\ i \in [n]\right\}. \\ &\le \frac{1}{m}\left(\|d\| + n r_{\max}/m\right) \\ &= \frac{\gamma}{n}\left(\|d\| + \gamma r_{\max}\right). \end{aligned}$$

In order to compute the Lipschitz constant L for ∇U, note that for any $x, y \in \mathbb{R}^T$, we have

$$\|\nabla U(x) - \nabla U(y)\| = \frac{1}{m}\left\|\frac{x}{m} - \frac{y}{m}\right\| = \frac{1}{m^2}\|x - y\|$$

so that we obtain $L = 1/m^2 = \gamma^2/n^2$. Substitute ρ, G, and L into (14) to obtain (11). Note that we have dropped the dependence on d, $r_{\max}$, and γ for brevity, since we are most interested in the relationship between suboptimality and (n, ϵ). ■

The suboptimality bound (11) indicates how performance (cost) is affected by incorporating privacy. As ϵ decreases, the level of privacy is elevated but at the expense of sacrificing performance as a result of increased suboptimality. This increase in suboptimality can be mitigated by introducing more participating users (i.e., by increasing n) as predicted by the bound (11); this coincides with the common intuition that it is easier to achieve privacy as the number of users n increases when only aggregate user information is available. Indeed, in the distributed EV charging algorithm, the gradients $p^{(k)}$ is a function of the aggregate load profile $\sum_{i=1}^{n} r_i^{(k)}$.

VII. Numerical Simulations

We consider the cost function as given by (7). The base load d is chosen according to the data provided in Gan *et al.* [13]. The scheduling horizon is divided into 52 time slots of 15 min. We consider a large pool of EVs $(n = 100\,000)$ in a large residential area $(m = 500\,000)$. For computational efficiency, instead of assigning a different charging specification $(\bar{r}_i, E_i)$ to every user $i \in [n]$, we divide the users into N $(N \ll n)$ groups and assign the same charging specification for every user in the same group. If we choose the same initial conditions $r_i^{(1)}$ for all users in the same group, the projected gradient descent update (step 3, Algorithm 2) becomes identical for all users in the group, so that the projection $\Pi_{\mathcal{C}_i}$ only needs to be computed once for a given group. We choose $N = 100$ and draw $(\bar{r}_j, E_j)$ for all $j \in [N]$ as follows. The entries of $\bar{r}_j$ are drawn independently from a Bernoulli distribution, where $\bar{r}_j(t) = 3.3$ kW with probability 0.5 and $\bar{r}_j(t) = 0$ kW with probability 0.5. The amount of energy E_j is drawn from the uniform distribution on the interval [28, 40] (kW). Note that E_j has been normalized against $\Delta T = 0.25$ h to match the unit of $\bar{r}_j$; in terms of energy required, this implies that each vehicle needs an amount between $[28, 40]\text{kW} \times 0.25\ \text{h} = [7, 10]$ kWh by the end of the scheduling horizon.

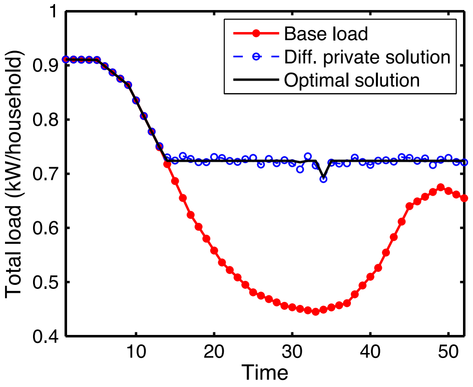


Fig. 1. Typical output of the differentially private distributed EV charging algorithm (Algorithm 2) with $\epsilon = 0.1$ compared to the optimal solution of problem (2). The other parameters used in the simulation are $K = 6$ and $c = 10$.

The constants δr and δE in the adjacency relation (8) are determined as follows. We choose $\delta r = 3.3 \times 4 = 13.2$ kW, so that the privacy of any events spanning less than 4 time slots (i.e., 1 h) can be preserved; we choose $\delta E = 40 - 28 = 12$ kW, which corresponds to the maximum difference in E_j $(j \in [N])$ as E_j varies between the interval [28,40] (kW). The other parameters in Algorithm 2 are chosen as follows: $L = 1/m^2$ (as computed in Section VI-D), $\eta = 1$, whereas ϵ, K, and c will vary among different numerical experiments.

Fig. 1 plots a typical output from Algorithm 2 with $\epsilon = 0.1$, alongside the optimal solution of problem (2). The dip at $t = 34$ that appears both in the differentially private solution and the optimal solution is due to the constraint imposed by $\bar{r}_i(t)$. Because of the noise introduced in the gradients, the differentially private solution given by Algorithm 2 exhibits some additional fluctuations compared to the optimal solution. The constant c that determines the step sizes is found to be insensitive for $c \in [10, 20]$ as shown in Fig. 2, so that we choose $c = 10$ in all subsequent simulations. In Fig. 2, the relative suboptimality is defined as $[U(\sum_{i=1}^{n} \hat{r}_i^{(K+1)}) - U^*]/U^*$, which is obtained by normalizing the suboptimality against U^*.

Fig. 3 shows the relative suboptimality as a function of K. It can be seen from Fig. 3 that an optimal choice of K exists, which coincides with the result of Theorem 7.

Fig. 4 shows the dependence of the relative suboptimality on ϵ. A separate experiment for investigating the dependence on n is not performed, since changing n is expected to have a similar effect as changing ϵ according to Corollary 9. As the privacy requirement becomes less stringent (i.e., as ϵ grows), the suboptimality of Algorithm 2 improves, which coincides qualitatively with the bound given in Corollary 9. One can quantify the relationship between the suboptimality and ϵ from the slope of the curve in Fig. 4; if the slope is s, then the relationship between the suboptimality and ϵ is given by $\mathcal{O}(\epsilon^s)$. By performing linear regression on the curve in Fig. 4, we can obtain the slope as $s \approx -0.698$, which is in contrast to

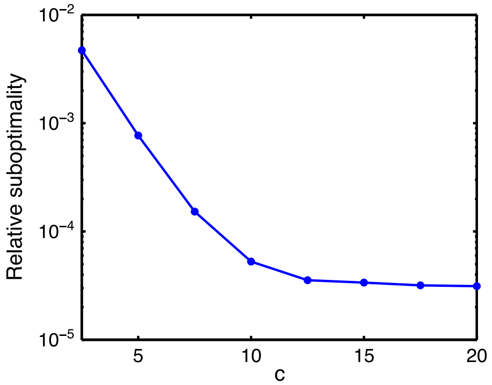


Fig. 2. Relative suboptimality of the differentially private distributed EV charging algorithm (Algorithm 2) as a function of the step size constant c. The other parameters used in the simulations are $\epsilon = 0.1$ and $K = 6$.

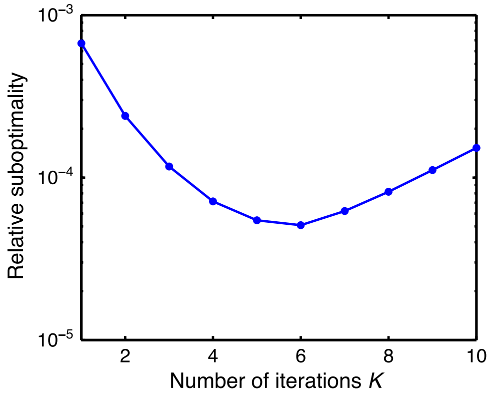


Fig. 3. Relative suboptimality of the differentially private distributed EV charging algorithm (Algorithm 2) as a function of the number of iterations K. The other parameters used in the simulations are $\epsilon = 0.1$ and $c = 10$.

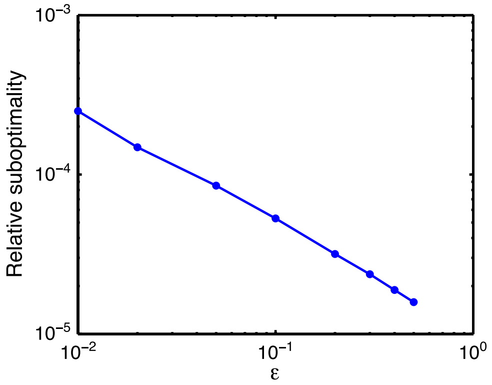


Fig. 4. Relative suboptimality of Algorithm 2 as a function of ϵ. Larger ϵ implies that less privacy is preserved. The slope is approximately -0.698 (compared to the theoretical bound -0.25 as given by Corollary 9). All simulations use $c = 10$. The number of iterations K is optimized for every choice of ϵ.

-0.25 given by Corollary 9. This implies that the suboptimality of Algorithm 2 decreases faster than the rate given by Corollary 9 as ϵ increases. In other words, the bound given by Corollary 9 is likely to be loose; this is possibly due to the fact that the result on the suboptimality of stochastic gradient descent (Proposition 14) does not consider additional properties of the objective function such as strong convexity.

VIII. Conclusion

This paper develops an ϵ-differentially private algorithm for distributed constrained optimization. The algorithm preserves privacy in the specifications of user constraints by adding noise to the public coordination signal (i.e., gradients). By using the sequential adaptive composition theorem, we show that the noise magnitude is determined by the sensitivity of the projection operation $\Pi_{\mathcal{C}}$, where $\mathcal{C}$ is the parameterized set describing the user constraints. By viewing the projection operation as a least-squares problem, we are able to compute the sensitivity of $\Pi_{\mathcal{C}}$ through a solution sensitivity analysis of optimization problems. We demonstrate how this sensitivity can be computed in the case of EV charging.

We also analyze the trade-off between privacy and performance of the algorithm through results on suboptimality analysis of the stochastic gradient descent method. Specifically, in the case of EV charging, the expected suboptimality of the ϵ-differentially private optimization algorithm with n participating users is upper bounded by $\mathcal{O}((n\epsilon)^{-1/4})$. For achieving the best suboptimality, both the suboptimality analysis and numerical simulations show that there exists an optimal choice for the number iterations: too few iterations affects the convergence behavior, whereas too many iterations leads to too much noise in the gradients. Simulations have indicated that the bound $\mathcal{O}((n\epsilon)^{-1/4})$ is likely not tight. One future direction is to derive a tighter bound for similar distributed optimization problems using information-theoretic approaches (e.g., [8], [23]). We have also found that it is not straightforward to extend the same techniques to other optimization algorithms beyond the gradient descent method, which makes designing other differentially private optimization algorithms a valuable and interesting future research direction.

Appendix I
Proof of Lemma 13

Consider any adjacent D and D' such that $\mathcal{C}_j = \mathcal{C}'_j$ for all $j \neq i$. We will first show that when the outputs of $\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(k-1)}$ are given, we have

$$\left\| r_i^{(k)}(D') - r_i^{(k)}(D) \right\| \leq (k-1)\Delta$$

$$\left\| r_j^{(k)}(D') - r_j^{(k)}(D) \right\| = 0, \quad \forall j \neq i.$$

We will prove the above result by induction. For $k = 1$, we have $\|r_i^{(1)}(D') - r_i^{(1)}(D)\| = 0$ for all $i \in [n]$.

Consider the case when $k > 1$. For notational convenience, we define for $i \in [n]$

$$v_i^{(k-1)}(D) := r_i^{(k-1)}(D) - \alpha_{k-1}\hat{p}^{(k-1)}. \qquad (26)$$

In (26), we have used the fact that the output of $\hat{p}^{(k-1)}$ is given so that the dependence of $\hat{p}^{(k-1)}$ on D is dropped according to the adaptive sequential composition theorem (Proposition 12). Then, for all $j \neq i$, we have

$$\begin{aligned}
&\left\|r_j^{(k)}(D') - r_j^{(k)}(D)\right\| \\
&\quad = \left\|\Pi_{\mathcal{C}_j'}\left(v_j^{(k-1)}(D')\right) - \Pi_{\mathcal{C}_j}\left(v_j^{(k-1)}(D)\right)\right\| \\
&\quad = \left\|\Pi_{\mathcal{C}_j}\left(v_j^{(k-1)}(D')\right) - \Pi_{\mathcal{C}_j}\left(v_j^{(k-1)}(D)\right)\right\| \\
&\quad \leq \left\|v_j^{(k-1)}(D') - v_j^{(k-1)}(D)\right\| \\
&\quad = \left\|r_j^{(k-1)}(D') - r_j^{(k-1)}(D)\right\| = 0 \\
&\left\|r_i^{(k)}(D') - r_i^{(k)}(D)\right\| \\
&\quad = \left\|\Pi_{\mathcal{C}_i'}\left(v_i^{(k-1)}(D')\right) - \Pi_{\mathcal{C}_i}\left(v_i^{(k-1)}(D)\right)\right\| \\
&\quad \leq \left\|\Pi_{\mathcal{C}_i'}\left(v_i^{(k-1)}(D')\right) - \Pi_{\mathcal{C}_i}\left(v_i^{(k-1)}(D')\right)\right\| \\
&\qquad\quad + \left\|\Pi_{\mathcal{C}_i}\left(v_i^{(k-1)}(D')\right) - \Pi_{\mathcal{C}_i}\left(v_i^{(k-1)}(D)'\right)\right\| \\
&\quad \leq \Delta + \left\|v_i^{(k-1)}(D') - v_i^{(k-1)}(D)\right\| \\
&\quad = \Delta + \left\|r_i^{(k-1)}(D') - r_i^{(k-1)}(D)\right\| \leq (k-1)\Delta
\end{aligned}$$

where we have used the induction hypothesis

$$\begin{aligned}
\left\|r_i^{(k-1)}(D') - r_i^{(k-1)}(D)\right\| &\leq (k-2)\Delta \\
\left\|r_j^{(k-1)}(D') - r_j^{(k-1)}(D)\right\| &= 0, \quad \forall j \neq i.
\end{aligned}$$

Then, the ℓ_2-sensitivity of $p^{(k)}$ can be computed as follows:

$$\begin{aligned}
&\left\|p^{(k)}(D') - p^{(k)}(D)\right\| \\
&\qquad \leq L\left\|\sum_{i=1}^{n}\left[r_i^{(k)}(D') - r_i^{(k)}(D)\right]\right\| \leq L(k-1)\Delta.
\end{aligned}$$

Since the above results hold for all i such that D and D' satisfy $\mathcal{C}_j = \mathcal{C}_j'$ for all $j \neq i$, we have

$$\begin{aligned}
\Delta^{(k)} &:= \max_{D,D':\mathrm{Adj}(D,D')} \left\|p^{(k)}(D') - p^{(k)}(D)\right\| \\
&= \max_{i\in[n]} \max\Big\{\left\|p^{(k)}(D') - p^{(k)}(D)\right\| : D, D' \text{ satisfy} \\
&\qquad\qquad \mathcal{C}_j = \mathcal{C}_j' \text{ for all } j \neq i\Big\} \\
&= L(k-1)\Delta.
\end{aligned}$$

Appendix II
Proofs on the Local Solution Sensitivities

A. Proof of Lemma 18

The Lagrangian of problem (16) can be written as

$$L(x,\lambda,\mu,\nu) = \frac{1}{2}\|x-x_0\|^2 - \lambda^T x + \mu^T(x-a) + \nu(b - \mathbf{1}^T x). \tag{27}$$

Denote by λ^*, μ^*, and ν^* the corresponding optimal Lagrange multipliers. It can be verified that all conditions in Proposition 17 hold. Apply Proposition 17 to obtain

$$\begin{aligned}
\partial_b x^* - \partial_b \lambda^* + \partial_b \mu^* - \partial_b \nu^* \cdot \mathbf{1} &= 0 && (28)\\
\mathbf{1}^T \partial_b x^* &= 1 && (29)\\
\lambda_i^* \cdot \partial_b x_i^* + x_i^* \cdot \partial_b \lambda_i^* &= 0, \quad i \in [T] && (30)\\
\mu_i^* \cdot \partial_b x_i^* + (x_i^* - a_i) \cdot \partial_b \mu_i^* &= 0, \quad i \in [T]. && (31)
\end{aligned}$$

Strict complementary slackness implies that either: 1) $x_i^* = 0$ and $\lambda_i^* > 0$, so that $\partial_b x_i^* = 0$ according to (30) or 2) $x_i^* \neq 0$ and $\lambda_i^* = 0$, so that $\partial_b \lambda_i^* = 0$ also according to (30). In other words, under strict complementary slackness, condition (30) is equivalent to

$$\partial_b \lambda_i^* \cdot \partial_b x_i^* = 0. \tag{32}$$

Similarly, we can rewrite condition (31) as

$$\partial_b \mu_i^* \cdot \partial_b x_i^* = 0. \tag{33}$$

Conditions (32) and (33) imply that one and only one of the following is true for any $i \in [T]$: 1) $\partial_b x_i^* = 0$ and 2) $\partial_b \lambda_i^* = 0$ and $\partial_b \mu_i^* = 0$. Define $\mathcal{I} := \{i : \partial_b x_i^* \neq 0\}$, and we have

$$\sum_{i\in\mathcal{I}} \partial_b x_i^* = 1 \tag{34}$$

from (29). Note that (28) implies that for all $i, j \in [T]$

$$\partial_b x_i^* - \partial_b \lambda_i^* + \partial_b \mu_i^* = \partial_b x_j^* - \partial_b \lambda_j^* + \partial_b \mu_j^*.$$

Since $\partial_b \lambda_i^* = 0$ and $\partial_b \mu_i^* = 0$ for all $i \in \mathcal{I}$, we have $\partial_b x_i^* = \partial_b x_j^*$ for all $i, j \in \mathcal{I}$ and hence $\partial_b x_i^* = 1/|\mathcal{I}|$ for all $i \in \mathcal{I}$ according to (34). On the other hand, from the definition of $\mathcal{I}$, we have $\partial_b x_i^* = 0$ for all $i \notin \mathcal{I}$. In summary, we have $\partial_b x^* \succeq 0$, which completes the proof.

B. Proof of Lemma 19

The optimality conditions for problem (16) imply that

$$\begin{aligned}
x^* - \lambda^* + \mu^* - \nu^* \mathbf{1} &= x_0 && (35)\\
\mathbf{1}^T x^* &= b && (36)\\
\lambda_i^* x_i^* &= 0, \quad i \in [T] && (37)\\
\mu_i^* (x_i^* - a_i) &= 0, \quad i \in [T]. && (38)
\end{aligned}$$

Suppose strict complementary slackness fails for a certain value of b. Denote the set of indices of the constraints that violate strict complementary slackness by $\mathcal{I}_\lambda := \{i : \lambda_i^* = 0,\ x_i^* = 0\}$ and $\mathcal{I}_\mu := \{i : \mu_i^* = 0,\ x_i^* = a_i\}$. If both $\mathcal{I}_\lambda$ and $\mathcal{I}_\mu$ are empty, then strict complementary slackness holds for all constraints.

Proof: When $\mathcal{I}_\lambda$ is nonempty, we know from (38) that $\mu_i^* = 0$ for all $i \in \mathcal{I}_\lambda$. For any $i \in \mathcal{I}_\lambda$, substitute $x_i^* = 0$, $\lambda_i^* = 0$, and $\mu_i^* = 0$ into (35) to obtain $\nu^* = x_{0,i}$. For any other $j \notin \mathcal{I}_\lambda$, one of the following three cases must hold: 1) $x_j^* = 0$; 2) $x_j^* = a_j$; or 3) $0 < x_j^* < a_j$. Consider a partition $(\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$ of the set $[n] \setminus \mathcal{I}_\lambda$ as follows:

$$\mathcal{I}_1 := \{j : x_j^* = 0\}, \qquad \mathcal{I}_2 := \{j : x_j^* = a_j\}$$
$$\mathcal{I}_3 := \{j : 0 < x_j^* < a_j\}.$$

For any $j \in \mathcal{I}_3$, we have $\lambda_j^* = \mu_j^* = 0$ from (30) and (31), so that we have $x_j^* = \nu^* + x_{0,j}$ according to (28). Then, we can write using (36)

$$\begin{aligned} b &= \mathbf{1}^T x^* \\ &= \sum_{i \in \mathcal{I}_\lambda} x_i^* + \sum_{j \in \mathcal{I}_1} x_j^* + \sum_{j \in \mathcal{I}_2} x_j^* + \sum_{j \in \mathcal{I}_3} x_j^* \\ &= |\mathcal{I}_\lambda| \nu^* + 0 + \sum_{j \in \mathcal{I}_2} a_j + \sum_{j \in \mathcal{I}_3} (\nu^* + x_{0,j}). \end{aligned} \tag{39}$$

Since both a and x_0 are fixed, we know that the choice of $\nu^* = x_{0,i}$ (for any $i \in \mathcal{I}_\lambda$) is finite. By enumerating all finitely many partitions $(\mathcal{I}_\lambda, \mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$ of $[T]$, we know that b can only take finitely many values according to (39). The proof is similar for the case when $\mathcal{I}_\mu$ is nonempty by making use of (37). When both $\mathcal{I}_\lambda$ and $\mathcal{I}_\mu$ are nonempty, the possible values of b are given by the intersection of those when only one of $\mathcal{I}_\lambda$ and $\mathcal{I}_\mu$ are empty; hence the number of possible values is also finite. ■

C. Proof of Lemma 21

Similar to the proof of Lemma 18, we apply Proposition 17 using the Lagrangian as given by (27). We can show that the following holds for all $i \in [T]$:

$$\partial_{a_i} x_i^* - \partial_{a_i} \lambda_i^* + \partial_{a_i} \mu_i^* - \partial_{a_i} \nu^* = 0 \tag{40}$$

$$\partial_{a_j} x_i^* - \partial_{a_j} \nu^* = 0, \quad \forall j \neq i \tag{41}$$

$$\lambda_i^* \partial_{a_j} x_i^* + x_i^* \partial_{a_j} \lambda_i^* = 0, \quad \forall j \in [T] \tag{42}$$

$$\mu_i^* \partial_{a_j} x_i^* + (x_i^* - a_i)\, \partial_{a_j} \mu_i^* - \mu_i^* = 0 \tag{43}$$

$$\mu_i^* \partial_{a_j} x_i^* + (x_i^* - a_i)\, \partial_{a_j} \mu_i^* = 0, \quad \forall j \neq i \tag{44}$$

$$\sum_{j=1}^{n} \partial_{a_i} x_j^* = 0. \tag{45}$$

From (40) and (41), we know that the following holds for all i:

$$\partial_{a_i} \nu^* = \partial_{a_i} x_i^* - \partial_{a_i} \lambda_i^* + \partial_{a_i} \mu_i^* = \partial_{a_i} x_j^*, \quad \forall j \neq i. \tag{46}$$

The first equality in (46) implies that there exists a constant C_i such that $\partial_{a_i} x_j^* = C_i$ for all $j \neq i$. Then, we can rewrite (45) as

$$\partial_{a_i} x_i^* + (T-1) C_i = 0, \quad \forall i \in [T] \tag{47}$$

which implies that

$$\begin{aligned} \|\partial_{a_i} x^*\|_1 &= \sum_{j=1}^{T} \left|\partial_{a_i} x_j^*\right| = \left|\partial_{a_i} x_i^*\right| + \sum_{j \neq i} \left|\partial_{a_i} x_j^*\right| \\ &= 2(T-1)|C_i|. \end{aligned} \tag{48}$$

Suppose strict complementary slackness holds. Then, for any $i \in [T]$, only one of the three following cases holds:

1) $x_i^* = 0$, $\lambda_i^* > 0$, $\mu_i^* = 0$;
2) $x_i^* = a_i$, $\lambda_i^* = 0$, $\mu_i^* > 0$;
3) $0 < x_i^* < a_i$, $\lambda_i^* = 0$, $\mu_i^* = 0$.

In the following, we will derive the expression of $\|\partial_{a_i} x^*\|_1$ for the three cases separately:

1) $x_i^* = 0$, $\lambda_i^* > 0$, $\mu_i^* = 0$:
Using (42), we obtain $\partial_{a_j} x_i^* = 0$ for all $j \in [T]$; in particular, this implies $\partial_{a_i} x_i^* = 0$. Substituting $\partial_{a_i} x_i^* = 0$ into (47), we obtain $C_i = 0$, so that $\|\partial_{a_i} x^*\|_1 = 0$ according to (48).
2) $x_i^* = a_i$, $\lambda_i^* = 0$, $\mu_i^* > 0$:
Using (43), we obtain $\partial_{a_i} x_i^* = 1$; substitute this into (47) to obtain $C_i = -(1/(T-1))$, so that we have $\|\partial_{a_i} x^*\|_1 = 2$ according to (48).
3) $0 < x_i^* < a_i$, $\lambda_i^* = 0$, $\mu_i^* = 0$:
Using (42)–(44), we obtain $\partial_{a_j} \lambda_i^* = \partial_{a_j} \mu_i^* = 0$ for all $j \in [T]$; in particular, this implies $\partial_{a_i} \lambda_i^* = \partial_{a_i} \mu_i^* = 0$. Then, using (46), we have $\partial_{a_i} x_i^* = C_i$; substitute this into (47) to obtain $C_i = 0$, so that $\|\partial_{a_i} x^*\|_1 = 0$ according to (48).

For case 2 in the above, the fact that $\partial_{a_i} x_j^* = C_i = -(1/(T-1)) \neq 0$ for all $j \neq i$ also implies that $x_j^* \neq a_j$ for all $j \neq i$. Otherwise, the fact $x_j^* = a_j$ would imply $\mu_j^* > 0$ as a result of strict complementary slackness. According to (44), we have

$$\mu_j^* \partial_{a_i} x_j^* + \left(x_j^* - a_j\right) \partial_{a_i} \mu_j^* = 0, \quad \forall i \neq j$$

so that $\mu_j^* > 0$ would imply $\partial_{a_i} x_j^* = 0$, which causes a contradiction. To summarize, there exists at most one $i \in [T]$ such that $\|\partial_{a_i} x^*\|_1 = 2$ (i.e., case 2 holds), whereas for other $j \neq i$ we have $\|\partial_{a_j} x^*\|_1 = 0$ (i.e., either case 1 or 3 holds). This implies that $\sum_{j=1}^{T} \|\partial_{a_j} x^*\|_1 \leq 2$, which completes the proof.

D. Proof of Lemma 22

Define $\mathcal{I}_\mu = \{i : \mu_i^* = 0,\ x_i^* = a_i\}$. If $\mathcal{I}_\mu$ is empty, then strict complementary slackness for the constraint $x \preceq a$ holds. For any $i \in \mathcal{I}_\mu$, we have $\nu^* = a_i - x_{0,i}$ and $\lambda_i^* = 0$ according to (35) and (37). For any other $j \notin \mathcal{I}_\mu$, one of the following three cases must hold: 1) $x_j^* = 0$; 2) $x_j^* = a_j$; or 3) $0 < x_j^* < a_j$. The last case implies that $\lambda_j^* = \mu_j^* = 0$, so that we have $x_j^* = a_i - x_{0,i} + x_{0,j}$, where $i \in \mathcal{I}_\lambda$. Since both b and x_0 are fixed, and only one a_i among all $i \in [T]$ is allowed to change due to the constraint imposed by the integration path L, we can use a similar argument as the one in the proof of Lemma 19 to conclude that there are finitely many values of a along L such that the constraint $b = \mathbf{1}^T x^*$ is satisfied.

References

[1] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. ACM Symp. Theory Comput.*, 1988, pp. 1–10.

[2] C. Berge, *Topological Spaces: Including a Treatment of Multi-Valued Functions, Vector Spaces and Convexity*. Mineola, NY, USA: Courier Dover, 1963.

[3] D. Block, J. Harrison, and P. Brooker, "Electric vehicle sales for 2014 and future projections," Florida Solar Energy Center, Cocoa, FL, USA, Tech. Rep., 2015.

[4] E. S. Canepa and C. G. Claudel, "A framework for privacy and security analysis of probe-based traffic information systems," in *Proc. ACM Int. Conf. High Confidence Netw. Syst.*, 2013, pp. 25–32.

[5] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 371–380, Feb. 2010.

[6] G. B. Dantzig, J. Folkman, and N. Shapiro, "On the continuity of the minimum set of a continuous function," *J. Math. Anal. Appl.*, vol. 17, no. 3, pp. 519–548, 1967.

[7] S. Deilami, A. S. Masoum, P. S. Moses, and M. A. S. Masoum, "Real-time coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile," *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 456–467, 2011.

[8] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE Annu. Symp. Found. Comput. Sci.*, 2013, pp. 429–438.

[9] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*. New York, NY, USA: Springer-Verlag, 2008, pp. 1–19.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*. New York, NY, USA: Springer-Verlag, 2006, pp. 265–284.

[11] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Theoret. Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2013.

[12] A. V. Fiacco, "Sensitivity analysis for nonlinear programming using penalty methods," *Math. Programm.*, vol. 10, no. 1, pp. 287–311, 1976.

[13] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 940–951, 2013.

[14] S. Han, U. Topcu, and G. J. Pappas, "Differentially private convex optimization with piecewise affine objectives," in *Proc. IEEE Conf. Decision Control*, 2014, pp. 2160–2166.

[15] J. Hsu, Z. Huang, A. Roth, T. Roughgarden, and Z. S. Wu, "Private matchings and allocations," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, 2014, pp. 21–30.

[16] J. Hsu, A. Roth, T. Roughgarden, and J. Ullman, "Privately solving linear programs," *Automata, Lang., Programm.*, vol. 8572, pp. 612–624, 2014.

[17] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. Int. Conf. Distrib. Comput. Netw.*, 2015, pp. 4:1–4:10.

[18] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power Energy Mag.*, vol. 7, no. 2, pp. 52–62, 2009.

[19] M. Kearns, M. Pai, A. Roth, and J. Ullman, "Mechanism design in large games: Incentives and privacy," in *Proc. Conf. Innov. Theor. Comput. Sci.*, 2014, pp. 403–410.

[20] J. L. Ny and G. J. Pappas, "Differentially private filtering," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 341–354, Feb. 2014.

[21] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 67–78, Jan. 2013.

[22] A. Narayanan and V. Shmatikov, "How to break anonymity of the Netflix prize dataset," Univ. Texas at Austin, Austin, TX, USA, 2007. [Online]. Available: http://arxiv.org/abs/cs/0610105

[23] M. Raginsky and A. Rakhlin, "Information-based complexity, feedback and dynamics in convex programming," *IEEE Trans. Inform. Theory*, vol. 57, no. 10, pp. 7036–7056, Oct. 2011.

[24] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, no. 3, pp. 400–407, 1951.

[25] R. M. Rogers and A. Roth, "Asymptotically truthful equilibrium selection in large congestion games," in *Proc. ACM Conf. Econ. Comput.*, 2014, pp. 771–782.

[26] L. Sankar, S. R. Rajagopalan, S. Mohajer, and H. V. Poor, "Smart meter privacy: A theoretical framework," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 837–846, Jun. 2013.

[27] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 71–79.

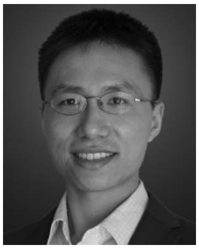
[28] E. Sortomme, M. M. Hindi, S. D. J. MacPherson, and S. S. Venkata, "Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 198–205, Mar. 2011.

[29] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[30] J. Taylor, A. Maitra, M. Alexander, D. Brooks, and M. Duvall, "Evaluations of plug-in electric vehicle distribution system impacts," in *Proc. IEEE Power Energy Society Gen. Meet.*, 2010, pp. 1–6.

[31] J. A. Tropp, I. S. Dhillon, R. W. Heath, and T. Strohmer, "Designing structured tight frames via an alternating projection method," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 188–209, Jan. 2005.

[32] P. Venkitasubramaniam, "Privacy in stochastic control: A markov decision process perspective," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2013, pp. 381–388.



Shuo Han (S'08–M'14) received the B.E. and M.E. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2003 and 2006, and the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2013.

He is currently a Postdoctoral Researcher in the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. His research interests include control theory, convex optimization, applied probability, and their applications in large-scale interconnected systems.

Ufuk Topcu received the bachelor's degree from Bogazici University, Istanbul, Turkey, in 2003, the master's degree from the University of California, Irvine, CA, USA, in 2005, and the Ph.D. degree from the University of California, Berkeley, CA, USA, in 2008.

He is currently an Assistant Professor at The University of Texas at Austin, Austin, TX, USA. Previously, he was a Postdoctoral Scholar at the California Institute of Technology and a Research Assistant Professor at the University of Pennsylvania. His research focuses on the design and verification of autonomous networked systems.

George J. Pappas (S'90–M'91–SM'04–F'09) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998.

He is currently the Joseph Moore Professor and Chair of the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. He also holds a secondary appointment with the Department of Computer and Information Sciences and the Department of Mechanical Engineering and Applied Mechanics. He is a Member of the GRASP Lab and the PRECISE Center. He had previously served as the Deputy Dean for Research with the School of Engineering and Applied Science. His research interests include control theory and, in particular, hybrid systems, embedded systems, cyberphysical systems, and hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and biomolecular networks.

Dr. Pappas has received various awards, such as the Antonio Ruberti Young Researcher Prize, the George S. Axelby Award, the Hugo Schuck Best Paper Award, the George H. Heilmeier Award, the National Science Foundation PECASE award, and numerous best student papers awards at ACC, CDC, and ICCPS.