Differentially Private Distributed Optimization *

Zhenqi Huang Sayan Mitra Nitin Vaidya
{zhuang25, mitras, nhv}@illinois.edu
Coordinate Science Laboratory
University of Illinois at Urbana Champaign
Urbana, IL 61801

ABSTRACT

In distributed optimization and iterative consensus literature, a standard problem is for N agents to minimize a function f over a subset of Euclidean space, where the cost function is expressed as a sum $\sum f_i$. In this paper, we study the private distributed optimization problem (PDOP) with the additional requirement that the cost function of the individual agents should remain differentially private. The adversary attempts to infer information about the private cost functions from the messages that the agents exchange. Achieving differential privacy requires that any change of an individual's cost function only results in unsubstantial changes in the statistics of the messages. We propose a class of iterative algorithms for solving PDOP, which achieves differential privacy and convergence to a common value. Our analysis reveals the dependence of the achieved accuracy and the privacy levels on the the parameters of the algorithm. We observe that to achieve ϵ-differential privacy the accuracy of the algorithm has the order of $O(\frac{1}{\epsilon^2})$.

Categories and Subject Descriptors

D.1.3 [**Software**]: Concurrent Programming—*Distributed programming*

General Terms

Algorithm

Keywords

Distributed Optimization, Differential Privacy, Iterative Consensus

1. INTRODUCTION

We introduce the private distributed optimization problem (PDOP) in which N agents are required to minimize a global cost function f that is the sum $\Sigma_{i=1}^{N} f_i$ of N cost functions for the individual agents. An instance of the problem arises when N secretive agents (with their own convex travel costs) wish to agree on a rendezvous point in a country such that (a) the travel cost for the entire group is minimized and (b) an adversary reading all the communication between the agents is unable to deduce the cost functions for the individuals. Similar problems arise in other applications such as balancing load of nodes in a network and fusing data for distributed sensors. We study iterative distributed algorithms for solving this problem in which agents exchange information about their current estimates for the optimal point and then update their estimates based on the information received from their neighbors. In doing so, however, the agents must preserve the privacy of their individual cost functions. The agents communicate over a communication network in which the connectivity may change over time. While iterative solutions for distributed optimization have been explored previously (see [11, 15, 16]), to our knowledge this paper is the first attempt to achieve this goal while maintaining privacy.

An alternative to distributed iterative optimization is a centralized strategy wherein a trusted *leader* is identified, with its task being to collect cost functions from all the other agents, perform the optimization centrally, and the distribute the results to all the other agents. While appealing for its simplicity, this strategy requires election of a leader, and maintenance of routes from all agents to the leader. The centralized scheme is then vulnerable to failure of the leader. Also, there is non-trivial cost of leader election and route maintenance in time-varying topologies, and in some systems learning the network topology itself violates privacy of the agents. Therefore, there has been significant interest in designing completely distributed algorithms for network-wide optimization and consensus. For instance, such algorithms have been designed for the *smart grids* [3] and *sensor networks* [13].

The notion of privacy we adopt is derived from ϵ-differential privacy [4–6] applied to continuous bit streams in [7]. This ϵ-differential privacy ensures that an adversary with access to all the communication in the system—we call this an observation sequence—cannot gain any significant information about the cost function of any agent.

In [1, 9] the authors solve a privacy preserving optimization problem with two methods: output perturbation and objective perturbation. In this problem, the cost functions of the individual are assumed to have a template and the computation is done by an entity that has access to all the

*The authors are supported by NSA SoS grant (W911NSF-13-0086) and NSF CAREER grant (CNS 10-54247)

ICDCN '15 Jan. 4-7, Goa, India

agents' individual data. In contrast, we study a class of problems that have to be solved in distributed ways without relying on any template of the individual cost functions.

In this paper, we propose a class of synchronous iterative distributed algorithms for solving PDOP. Iterative algorithms proceed in rounds. In each round, each agent participating in our algorithm executes three subroutines. First, it adds a vector of random noise, drawn from Laplace distribution, to its estimate for the optimal point and broadcasts this noisy estimate to its neighboring agents. Sharing noisy estimates enables the agent to protect the privacy of its cost functions. For convergence of the estimates to the optimal point, however, the noise added in successive rounds must decay down to 0. Indeed, in our algorithm the parameters of the successive Laplace distributions are chosen such that they converge to the Dirac distribution. Next, the agent computes a weighted average over its neighbors' noisy broadcasts based on the communication graph of that round. Finally, the agent computes a new estimate by moving the average value against the gradient of its own cost function according to a carefully chosen step size.

A key quantity which determines the amount of noise to be added in each round for achieving differential privacy is the sensitivity of the algorithm. Roughly, the sensitivity at round t is the change in the observable behavior of the system at round t, namely the messages exchanged at round t, with change in the cost function of any agent (see Definition 5). For differential privacy, the ratio of the sensitivity and the parameter for the Laplace noise must be small (see Lemma 2). For the estimate of the optimal point to get arbitrarily close to the optimal point, standard iterative algorithms for distributed optimization (for example, the ones discussed in [15]), require the sum of the step sizes to be infinite. This strategy, however, would increase the sensitivity of the system for later rounds. That is, an adversary could begin to infer significant information about the individual cost functions. Thus, unlike the standard algorithms and our previous algorithm for private consensus [8], our algorithm for PDOP uses step sizes that sum to a finite quantity. Assuming that the domain is bounded, we then establish convergence and both the level of differential privacy and the accuracy of the algorithm (Theorems 5 and 10).

The algorithm has four parameters: the privacy level, the initial step size, the step size decay rate and the noise decay rate. Our analysis reveals that the accuracy level d has the oder of the inverse-square of the privacy level ϵ.

2. PRELIMINARIES

The algorithms presented in this paper rely on random real numbers drawn according to the Laplace distribution. For a constant $c > 0$, $Lap(c)$ denotes the Laplace distribution with probability density function $p_c(x) \triangleq \frac{1}{2c} e^{-\frac{|x|}{c}}$. This distribution has mean zero and variance $2c^2$. For any $x, y \in \mathbb{R}$, it can be shown that $\frac{p_c(x)}{p_c(y)} \leq e^{\frac{|y-x|}{c}}$.

For a natural number $N \in \mathbb{N}$, we denote the set $\{1, \ldots, N\}$ by $[N]$. For a vector v of length n, the i^{th} component is denoted by v_i. The transpose of v is denoted by v^T. For a vector v in $\mathbb{R}^n$ and $1 \leq p \leq \infty$, $||v||_p$ stands for the standard L^p-norm for v. Without a subscript, $||\cdot||$ stands for L^2-norm. That is, $||v|| = \sqrt{v^T v}$. For any vector $v \in \mathbb{R}^n$, the inequality $||v||_2 \leq ||v||_1 \leq \sqrt{n}||v||_2$ holds.

An *Euclidean projection* of a point $x \in \mathbb{R}^n$ onto a convex compact set $\mathcal{X} \subseteq \mathbb{R}^n$ is a point in $\mathcal{X}$ that is closest to x measured by Euclidean norm. If there are multiple candidate points, one is chosen arbitrarily, and to reduce notational overhead we treat the Euclidean projection $Proj_{\mathcal{X}}(x)$ as a function of x and $\mathcal{X}$. That is, $y = Proj_{\mathcal{X}}(x)$ if $y \in \mathcal{X}$ and $||y - x|| \leq ||z - x||$ for any $z \in \mathcal{X}$. A well known property of projection is that it does not increase the distance between points. That is, $||Proj_{\mathcal{X}}(x) - Proj_{\mathcal{X}}(y)|| \leq ||x - y||$ for any $x, y \in \mathbb{R}^n$.

A differentiable function $f : \mathcal{X} \mapsto \mathbb{R}$ is convex if for any $x, y \in \mathcal{X}$, $\bigtriangledown f(x)^T (y - x) \leq f(y) - f(x)$. Moreover, if there exists a positive constant $c > 0$ such that $\bigtriangledown f(x)^T (y - x) \leq f(y) - f(x) - \frac{c}{2}||y - x||^2$, the function f is said to be *strongly convex*. If f is second-order differentiable, strongly convexity of f is equivalent to $\bigtriangledown^2 f(x) - cI \succeq 0$. Strongly convex functions on compact domains have a unique minima [12]. For example, for constant $a \in \mathbb{R}^n$, the quadratic function $f(x) = ||x - a||^2$ is a strongly convex function in $\mathbb{R}^n$.

The following basic Lemma from [11] will be used in our analysis.

Proposition 1. *For a constant $\beta \in (0, 1)$ and a convergent scalar sequence $\{a_t\}_{t \in \mathbb{N}}$ such that $\lim_{t \to \infty} a_t = 0$, the following holds:*

$$\lim_{t \to \infty} \sum_{s=1}^{t} \beta^{t-s} a_s = 0. \tag{1}$$

3. THE PRIVATE DISTRIBUTED OPTIMIZATION PROBLEM

A *Private Distributed Optimization* (PDOP) problem $\mathcal{P}$ for N agents is specified by four parameters:

(i) $\mathcal{X} \subseteq \mathbb{R}^n$ is the domain of optimization,

(ii) $\mathcal{F} \subseteq \{\mathbb{R}^n \mapsto \mathbb{R}\}$ is a set of real-valued, strongly convex and differentiable individual cost functions, and $f(x) \triangleq \sum f_i(x)$ with $f_i \in \mathcal{F}$ for each $i \in [N]$ is the global cost function, and

(iii) $\mathcal{A} = \{A_t\}_{t \in \mathbb{N}}$ is a sequence of $N \times N$ matrices which specify the time-varying communication graph.

More details on these parameters and additional assumptions we use for solving PDOP will be stated in Section 3.1. In Section 4.1, we introduce the class of algorithms we study in this paper. In Section 3.3, we formally state the requirements for solving PDOP.

We describe the problem $\mathcal{P}$ as follows. The system consists of N agents. Each agent $i \in [N]$ is associated with an individual cost function $f_i : \mathbb{R}^n \mapsto \mathbb{R}$. The individual cost f_i is only known to agent i. Together the agents aim to minimize:

$$f(x) = \sum_{i \in [N]} f_i(x), \tag{2}$$

subject to the constraint $x \in \mathcal{X}$. We define $f^*_{\mathcal{P}} \triangleq \min_{x \in \mathcal{X}} f(x)$ as the global minimum for f and $x^*_{\mathcal{P}} \triangleq \arg\min_{x \in \mathcal{X}} f(x)$ as the point in $\mathcal{X}$ that minimizes the cost function. For a PDOP $\mathcal{P}$ we denote its components and related quantities by $\mathcal{X}_{\mathcal{P}}, \mathcal{F}_{\mathcal{P}}, f_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}}, f^*_{\mathcal{P}}$ and $x^*_{\mathcal{P}}$. We drop the subscript when it is clear from context. For a pair of PDOPs $\mathcal{P}$ and $\mathcal{P}'$, we

will also denote the corresponding quantities by $\mathcal{X}, \mathcal{F}, \ldots$, and $\mathcal{X}'$, $\mathcal{F}'$, etc. To illustrate the idea, we present a private rendezvous problem in Example 1.

Example 1 N agents live in a compact region $\mathcal{X}$ in a 2-D plane, where the address of each agent i is a point $x_i \in \mathcal{X} \subseteq \mathbb{R}^2$. The agents wants to decide an assembly point $x \in \mathbb{R}^2$ without sharing their actual address. The cost of agent i to go to point x is the squared distance $f_i(x) = ||x_i - x||^2$. Moreover, each agent can only keep in touch with a subset of the other agents. Then, we can cast this problem as a PDOP, where (i) the domain of optimization is $\mathcal{X} \subseteq \mathbb{R}^2$, (ii) the set of objective functions $\mathcal{F} = \{f | a \in \mathcal{X}, f(x) = ||a - x||^2\}$, (iii) the global cost function $f(x) = \sum f_i(x)$, and (iv) $\mathcal{A}$ is a sequence of matrices that specify the possibly time-varying communication topology. □

3.1 Domain, Cost Function and Communication Graph

We make the following assumptions on the domain of optimization and the set of individual cost functions throughout the paper.

Assumption 1 (Convexity and compactness). *(i) The set $\mathcal{X}$ is compact and convex. Let $C_1 \triangleq \sup_{x,y \in \mathcal{X}} ||x - y||$ denote the diameter of $\mathcal{X}$.*

(ii) The gradients of all the individual cost function are bounded on $\mathcal{X}$. We denote $C_2 > 0$ as the bound for $\mathcal{X}$ such that for any $x \in \mathcal{X}$ and any $g \in \mathcal{F}$, $|| \bigtriangledown g(x)|| \leq C_2$.

(iii) The functions in $\mathcal{F}$ are second-order differentiable and strongly convex. That is, there exists $C_3 > 0$ such that for any $x \in \mathbb{R}^n$ and any $g \in \mathcal{F}$, $C_3 I \preceq \bigtriangledown^2 g(x)$.

(iv) The norm of the second-order derivative of functions in $\mathcal{F}$ are bounded. We denote $C_4 > 0$ is the bound such that $|| \bigtriangledown^2 g(x)|| \leq C_4$.

The last part requires that the second-order derivatives of the objective functions exist. The condition $C_3 I \preceq \bigtriangledown^2 g(x)$ implies that the objective functions are strongly convex such that for any $x, y \in \mathbb{R}^n$ and for any $g \in \mathcal{F}$, $\bigtriangledown g(x)^T (y - x) \leq g(y) - g(x) - \frac{C_3}{2} ||y - x||^2$. The condition $|| \bigtriangledown^2 g(x)|| \leq C_4$ is a technical assumption such that the existence of C_4 helps to derive a clean proof of convergence of the algorithm. It can be checked that the PDOP introduced in Example 1 satisfies Assumption 1.

We assume a synchronous model of distributed computation through a time varying communication network among the agents. We model the communication network at round t as a weighted graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathcal{W}_t)$, where (i) $\mathcal{V} = [N]$ is the set of agents, (ii) $\mathcal{E}_t \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges over which information is exchanged at round $t \in \mathbb{N}$, and (iii) $\mathcal{W}_t : \mathcal{E}_t \mapsto (0, 1]$ is the weighted function that labels each edge with a positive weight. The graph $\mathcal{G}_t$ is represented by an $N \times N$ matrix A_t, where the entry $a_{i,j}(t) \triangleq \mathcal{W}_t(i, j)$ if $(i, j) \in \mathcal{E}_t$ otherwise $a_{i,j}(t) := 0$. We assume that the matrix A_t is doubly stochastic. That is, for each $i \in [N]$, $\sum_{j \in [N]} a_{i,j}(t) = 1$ and for each $j \in [N]$, $\sum_{i \in [N]} a_{i,j}(t) = 1$. Roughly, a doubly stochastic A_t ensures that each agent's decision has an equal influence on the final decision. This statement will become clearer after we introduce the algorithm. There are existing distributed algorithms to derive a doubly stochastic matrix among a network (see e.g. [2]). We use the following technical assumptions of the time-varying communication network $\mathcal{A}$ throughout the paper.

Assumption 2 (Robust connectivity). *We assume that for each $t \in \mathbb{N}$, the graph A_t is strongly connected. In addition, there exists a minimal connection strength $\eta \in (0, 1]$ such that for each $t \in \mathbb{N}$:*

(i) $a_{i,i}(t) \geq \eta$ for each $i \in [N]$. And

(ii) $a_{i,j}(t) > 0$ implies that $a_{i,j}(t) \geq \eta$.

This assumption guarantees that there exists a path in the graph linking each pair of the agents and the product of weights along the path is lower bounded. Notice that the links could be undirected.

3.2 Iterative Distributed Algorithms for PDOP

We study a class of iterative distributed algorithms for solving PDOP. As shown in Algorithm 1, R, U and F are functions or subroutines, which when instantiated will give candidate algorithms. The constant T is the total number of rounds over which the algorithm is executed and it determines the accuracy of the final answer. The agents have internal states. An agent's state is defined by the valuations of individual variables. Each agent has four internal variables: (i) $x_i \in \mathcal{X}$ is agent i's current estimate of the optimal point; it is initialized to an arbitrary point x_{i0} in $\mathcal{X}$, (ii) $y_i \in \mathbb{R}^n$ is the value agent i broadcasts to other agents, (iii) $z_i \in \mathbb{R}^n$ is the value agent i computes based on the values it receives from its neighbors, (iv) $t \in \mathbb{N}$ is the current round number, and (v) *buffer* is an ordered set which stores the messages received by agent i in a given round from its neighbors.

Algorithm 1: Template for iterative solution of PDOP.

```
1: Input: f_i, X, A
2: x_i ← x_i0;
3: for t = 1 : T do
4:     y_i ← R(x_i, t);
5:     Broadcast(y_i);
6:     buffer_i ← Receive();
7:     z_i ← F(A_t, buffer_i) ;
8:     x_i ← U(z_i, t, f_i, X);
9: end for
10: return x_i
```

Message exchange between agents is assumed to be atomic. That is, the **Broadcast**(y_i) routine of agent i broadcasts y_i to all his neighbors and the **Receive**() routine receives all neighbors' broadcasts from that round. This can be implemented by underlying message exchanging protocols. In each round $t \in \mathbb{N}$, the algorithm has four phases: (i) each agent executes a subroutine R to compute the value to report (y_i) based on his individual value (x_i) (line 4), (ii) each agent broadcasts its value (y_i) and receives all neighbors' reports (line 5-6), (iii) each agent executes a subroutine F to compute a aggregate value (z_i) based on its neighbors' messages (line 7), and (iv) each agent executes a subroutine U to compute a new individual value (x_i) that reduces the individual cost function f_i (line 8).

We denote $x_i(t)$ as the valuation of $x_i \in \mathcal{X}$ at the end of round t. We denote the aggregate state $x(t) \in \mathcal{X}^N$ as a vector of the N individual valuations: $x(t) \triangleq [x_1(t), \ldots, x_N(t)]$. $y_i(t)$, $z_i(t)$, $y(t)$ and $z(t)$ are similarly defined as valuations of individual variables and vectors at the end of round t. Each round of a iterative distributed algorithm transforms the state vector of the entire system to a new state vector. An *execution* of such an algorithm, for a given PDOP, is a possibly infinite sequence of the form $\alpha = x(0), \langle x(1), y(1), z(1), \mathit{buffer}(1)\rangle, \langle x(2), y(2), z(2), \mathit{buffer}(2)\rangle, \ldots$. The observable part of such an execution are the corresponding infinite sequence of messages $y(1), y(2), \ldots$. We denote the observation mapping $\mathcal{R}(\alpha) \triangleq y(1), y(2), \ldots$ which gives the sequence of messages exchanged for the execution α..

Note that the set of messages stored in $\mathit{buffer}_i(t)$ is uniquely specified by the vector $y(t)$ and the communication graph for the round A_t. Thus, for deterministic subroutines R, F, and U, and particular choices of the initial valuations of the variables, and a given PDOP $\mathcal{P}$ an iterative distributed algorithm has a unique execution. For fixed (possibly randomized) subroutines U, R, F, and a fixed initial state $x(0)$, let *Obs* denote the set of all sequences of messages that the resulting algorithm can produce for any PDOP problem[1].

In this paper, we will study randomized versions of Algorithm 1. For a fixed choice of these randomized subroutines (to be stated in Section 4.1), $\Xi_{\mathcal{P}}$ denotes the set of all executions of the resulting algorithm for a given PDOP $\mathcal{P}$ and a given set of initial conditions[2]. The probability measure over the space of executions $\mathbb{P}_{\mathcal{P}}$ is defined in the standard way by first defining a σ-algebra of cones over the space of executions, and then by defining the probability of the cones by integrating over μ (see for example [10, 14]).

3.3 Convergence, Accuracy and Differential Privacy

An iterative distributed algorithm solves the PDOP problem if the estimates of all the agents converge to a common value as the time bound T goes to infinity and the algorithm preserves differential privacy of the f_i's. Furthermore, we want this convergence point close to the optima x^* of f.

Definition 1 (Convergence)**.** *An iterative distributed algorithm converges if for any PDOP $\mathcal{P}$ and any initial configuration, for any agents $i, j \in [N]$,*

$$\lim_{t\to\infty} \mathbb{E}||(x_i(t) - x_j(t)|| = 0,$$

where the expectation is taken over the coin-flips of the algorithm, that is, the randomization in the R, F and U subroutines of the individual agents.

We define $\bar{x}(t) \triangleq \frac{1}{N}\sum_{i\in[N]} x_i(t)$ as the average of the individual agent estimates at the end of round t. We define the accuracy of the algorithm by the expected squared distance of the average to the optima x^*.

Definition 2 (Accuracy)**.** *For a $d \geq 0$, an iterative distributed algorithm is said to be d-accurate if,*

$$\lim_{t\to\infty} \mathbb{E}||\bar{x}(t) - x^*||^2 \leq d, \tag{3}$$

where the expectation is taken over the coin-flips of the algorithm.

The smaller the d-accuracy, the more accurate the algorithm. If the algorithm converges to the exact global optimal point x^*, then it is 0-accurate.

Our definition of privacy is a modification of the notion of *differential privacy* introduced in [7] in the context of streaming algorithms. We consider an adversary with full access to all the communication channels. That is, he can peek inside all the messages ($y(t)$) going back and forth between the agents. For a give PDOP $\mathcal{P}$, observation sequence of messages $\rho \in \mathit{Obs}$, and an initial state $x(0)$, then $\mathcal{R}^{-1}(\mathcal{P}, \rho, x(0))$ is the set of executions $\{\alpha \in \Xi_{\mathcal{P}} : \mathcal{R}(\alpha) = \rho \wedge \alpha(0) = x(0)\}$ that can generate the observation ρ.

Definition 3 (Adjacency)**.** *Two PDOPs $\mathcal{P}$ and $\mathcal{P}'$ are* adjacent*, if the following holds:*

(i) $\mathcal{X} = \mathcal{X}'$, $\mathcal{F} = \mathcal{F}'$ and $\mathcal{A} = \mathcal{A}'$, that is, the domain of optimization, the set of individual cost functions and the communication graphs are identical, and

(ii) there exists an $i \in [N]$, such that $f_i \neq f_i'$ and for all $j \neq i$, $f_j = f_j'$.

That is, two PDOP are adjacent if only one agent changed its individual cost function while all other parameters are identical.

Definition 4. *For an $\epsilon > 0$, the iterative distributed algorithm is* ϵ-differentially private*, if for any two adjacent PDOPs $\mathcal{P}$ and $\mathcal{P}'$, any set of observation sequences $Y \subseteq \mathit{Obs}$ and any initial state $x(0) \in \mathcal{X}^N$*

$$\mathbb{P}[\mathcal{R}^{-1}(\mathcal{P}, Y, x(0))] \leq e^{\epsilon}\mathbb{P}[\mathcal{R}^{-1}(\mathcal{P}', Y, x(0))], \tag{4}$$

where the probability is taken over the coin-flips of the algorithm.

Roughly, the notion of ϵ-differential privacy ensures that an adversary with access to all the observation sequence cannot gain information about the individual cost function of any agent with any significant probability. A smaller ϵ suggests a higher privacy level. In the rest of the paper, we discuss algorithms to solve PDOP which guarantee ϵ-differential privacy and d-accuracy.

4. AN ALGORITHM FOR PDOP INSTANTIATING THE SUBROUTINES

4.1 Algorithm Description

In Section 4.1, we introduced a class of iterative distributed algorithms in terms of the subroutines R, F and U. In this section, we instantiate the subroutines and analyze the convergence, accuracy and differential privacy of the resulting algorithm. The algorithm has 4 parameters: (i) the privacy parameter $\epsilon > 0$, (ii) the initial step size parameter $c \in (0, \frac{1}{C_3})$, (iii) the step size decay rate $q \in (0, 1)$ and (iv) the noise decay rate $p \in (q, 1)$.

The subroutine R, shown in Algorithm 2, computes a value to broadcast based on agent i's local value x_i at round t. The agent first generates a vector of n (which is the length of x_i) noise values drawn independently from the Laplace distribution $Lap(M_t)$ with parameter M_t, which we will define later. The value to report is the sum of x_i and the noise vector w_i.

[1]Here we are suppressing the dependence of *Obs* on U, R, F and $x(0)$ for notational convenience.

[2]Here we are suppressing the dependence of $\Xi_{\mathcal{P}}$ and $\mathbb{P}_{\mathcal{P}}$ on U, R, F and $x(0)$ for notational convenience.

Algorithm 2: Subroutine R

Input: $x_i \in \mathcal{X}, t \in [T]$
$w_i \sim Lap(M_t)$;
$y_i \leftarrow x_i + w_i$;
return y_i

The subroutine F, shown in Algorithm 3, computes a value z_i based on all the neighbors' messages. In this subroutine, the agent i first reads its neighbors' broadcasts from its own buffer. Recall that $a_{i,j}(t)$ is the entry on the i^{th} column and j^{th} row of a doubly stochastic matrix A_t. Thus, the value $z_i = \sum_{j\in[N]} a_{ij}(t)y_j$ is indeed the weighted average of neighbors' broadcast based on the communication graph A_t.

Algorithm 3: Subroutine F

Input: $A_t, buffer_i$
For all $j \in [N]$, $y_j \leftarrow read(buffer_i, j)$;
$z_i \leftarrow \sum_{j\in[N]} a_{ij}(t)y_j$;
return z_i

The subroutine U is shown in Algorithm 4. In this subroutine, the agent computes a new local value x_i by moving from z_i against the gradient of f_i. Roughly, this computation reduces the individual cost function f_i from the point z_i. The parameter γ_t is the step size at round t, which we will define later in Equation (8). The projection $Proj_{\mathcal{X}}$ guarantees that the estimate for agent i is in $\mathcal{X}$.

Algorithm 4: Subroutine U

Input: $z_i \in \mathcal{X}, t \in [T], f_i, \mathcal{X}$
$x_i \leftarrow Proj_{\mathcal{X}}[z_i - \gamma_t(\bigtriangledown f_i(z_i))]$;
return x_i

From Algorithms 2-4, we can write down the computation of each agent i at round t as following three equations:

$$y_i(t) = x_i(t-1) + w_i(t) \quad (5)$$

$$z_i(t) = \sum_{j\in[N]} a_{ij}(t)y_j(t). \quad (6)$$

$$x_i(t) = Proj_{\mathcal{X}}[z_i(t) - \gamma_t(\bigtriangledown f_i(z_i(t)))]. \quad (7)$$

In Equations (5)-(7), there are two undefined parameters: the noise parameter M_t and the step size γ_t at round t. We propose to choose M_t and γ_t as geometrically decaying sequences depending on 4 parameters (c, q, p, ϵ):

$$\begin{aligned} \gamma_t &= cq^{t-1} \\ M_t &= 2C_2\sqrt{n}\tfrac{cp}{\epsilon(p-q)}p^{t-1}. \end{aligned} \quad (8)$$

where $c \in (0, \frac{1}{C_3})$ is the initial step size parameter, $q \in (0, 1)$ is the step size decay parameter, $p \in (q, 1)$ is the noise decay parameter and $\epsilon > 0$ is the privacy parameter. The initial noise $(2C_2\sqrt{n}\frac{c}{\epsilon(p-q)})$ depends on the four parameters c, q, p, ϵ, the dimension of domain n (See (i) in Definition of PDOP) and constant C_2 from Assumption 1. Note that the noise distribution converges to Dirac distribution and the step size converges to 0.

Thus, the algorithm we introduced to solve the PDOP (Algorithms 2-4) has three tunable parameters: the initial noise parameter c, its decaying rate q and step size decay rate p. Later we show that by any choice of $c \in (0, \frac{1}{C_3})$, $q \in (0, 1)$ and $p \in (q, 1)$, the iterative distributed algorithm is ϵ-differentially private, convergent, and ensures certain level of accuracy. In Section 4.4 we will discuss a tradeoff between convergence rate and accuracy.

4.2 Differential Privacy

Recall in Definition 3, two PDOP $\mathcal{P}$ and $\mathcal{P}'$ are adjacent if only one term of f' is different from that of f. The notion of sensitivity of a mechanism captures the maximum change in the states $(x(t))$ for two adjacent PDOPs. Recall that in Section 3.3, we introduced a inverse observation mapping $\mathcal{R}^{-1}(\mathcal{P}, \rho, x(0))$ that maps a PDOP $\mathcal{P}$, an observation sequence ρ and an initial configuration $x(0)$ to a set of executions, each of which is in the form $\alpha = x(0), \langle x(1), y(1), z(1), buffer(1), \rangle, \langle x(2), y(2), z(2), buffer(2)\rangle, \ldots$.

Let $\mathcal{R}^{-1}_{x(t)}(\mathcal{P}, \rho, x(0))$ be the set of $x(t)$ component from each of the executions α in the set of the executions $\alpha \in \mathcal{R}^{-1}(\mathcal{P}, \rho, x(0))$. The definition of $\mathcal{R}^{-1}$ is extended to a set of observation sequences in natural ways.

Definition 5. *At each round $t \in \mathbb{N}$, any initial state $x(0) \in \mathcal{X}^N$ and any adjacent PDOPs $\mathcal{P}, \mathcal{P}'$, We define the sensitivity of Algorithm 1 as*

$$\Delta(t) \triangleq \sup_{\rho\in Obs} \sup_{\substack{x\in\mathcal{R}^{-1}_{x(t)}(\mathcal{P},\rho,x(0)) \\ x'\in\mathcal{R}^{-1}_{x(t)}(\mathcal{P}',\rho,x(0))}} ||x - x'||_1,$$

where the norm used is L^1-norm.

We will show that $\Delta(t)$ is bounded for any $t \in \mathbb{N}$ for the algorithm. We state the following lemma which is a sufficient condition on the amount of noise to guarantees ϵ-differential privacy.

Lemma 2. *At each round $t \in \mathbb{N}$, if each agent adds a noise vector $w_i(t)$ consisting of n Laplace noise independently drawn from $Lap(M_t)$ such that $\sum_{t=1}^{\infty}\frac{\Delta(t)}{M_t} \leq \epsilon$, then the iterative distributed algorithm is ϵ-differentially private.*

Proof. We first fix an arbitrary time bound T and discuss executions bounded by T. Then by letting T goes to infinity the result holds. Fix any pair of adjacent PDOP $\mathcal{P}$ and $\mathcal{P}'$, any set of observation sequence $Y \subseteq Obs$ of length T and any initial state $x(0) \in \mathcal{X}$. For simplicity, we denote the sets of executions $\mathcal{R}^{-1}(\mathcal{P}, Y, x(0))$ and $\mathcal{R}^{-1}(\mathcal{P}', Y, x(0))$ by A and A' respectively. First we show that $\mathcal{R}^{-1}$ gives a singleton set.

Proposition 3. *For any PDOP $\mathcal{P}$, any observation sequence $\rho \in Y$ and for any initial state $x(0) \in \mathcal{X}^N$, the set of executions $\mathcal{R}^{-1}(\mathcal{P}, \rho, x(0))$ is a singleton set.*

Proof. Fixed $\mathcal{P}$, the communication graphs A_t are fixed. Fixed an observation sequence ρ, the messages $y(t)$ at each round t are fixed. From Equation (6), for each $i \in [N]$ and $t \in [T]$, $z_i(t)$ is uniquely determined. Then by Equation (7), recalling that f_i is specified by $\mathcal{P}$, we can conclude that $x_i(t)$ is uniquely specified for each $i \in [N]$ and $t \in [T]$. Thus, the execution $\alpha = x(0), \langle x(1), y(1), z(1)\rangle, \cdots = \mathcal{R}^{-1}(\mathcal{P}, \rho, x(0))$ is uniquely determined. □

We define a correspondence B between the sets A and A'. For $\alpha \in A$ and $\alpha' \in A'$, $B(\alpha) = \alpha'$ if and only if they have the same observation sequence. That is $\mathcal{R}(\alpha) = \mathcal{R}(\alpha')$. Fix any observation sequence ρ in Y, there is an unique execution $\alpha \in A$ that can produce the observation. Similarly, α' is also unique in A'. So B is indeed a bijection. we relate the probability measures of the sets of executions A and A'.

$$\frac{\mathbb{P}[\mathcal{R}^{-1}(\mathcal{P}, Y, x(0))]}{\mathbb{P}[\mathcal{R}^{-1}(\mathcal{P}', Y, x(0))]} = \frac{\int_{\alpha\in A}\mathbb{P}[\alpha]d\mu}{\int_{\alpha'\in A'}\mathbb{P}[\alpha']d\mu'}. \tag{9}$$

Changing the variable using the bijection B we have,

$$\int_{\alpha'\in A'}\mathbb{P}[\alpha']d\mu' = \int_{B(\alpha)\in A'}\mathbb{P}[B(\alpha)]d\mu = \int_{\alpha\in A}\mathbb{P}[B(\alpha)]d\mu \tag{10}$$

From Algorithms 2-4, recall that we fixed the observation sequence ρ, the probability comes from the noise $w_i(t)$. Along the execution, $x_i(t)$ is a vector of length n. We denote the k state component of $x_i(t)$ by $x_i^{(k)}(t)$. From Algorithm 2, $y_i(t)$ is obtained by adding n independently generated noise terms to $x(t)$, from the distribution $Lap(M_t)$, it follows that the probability density of an execution α is reduced to

$$\mathbb{P}[\alpha] = \prod_{\substack{i\in[N],k\in[n]\\ t\in[T]}} p_{M_t}(y_i^{(k)}(t) - x_i^{(k)}(t)), \tag{11}$$

where $p_b(x)$ is the probability density function of $Lap(b)$ at x. Then, for any $t \in [T]$, we relate the distance at time t between the state of α and $B(\alpha)$ with the sensitivity $\Delta(t)$. By Definition 5, we have

$$||x(t) - x'(t)||_1 \leq \Delta(t).$$

The norm in above equation is L^1-norm. The global state $x(t)$ consists of N local state $x_i(t)$, each of which has n component. So $(x(t) - x'(t))$ lives in space $\mathbb{R}^{nN}$. By definition of L^1-norm:

$$\sum_{i=1}^{N}\sum_{k=1}^{n}|x_i^{(k)}(t) - x_i'^{(k)}(t)| = ||x_i(t) - x_i'(t)||_1 \leq \Delta(t).$$

Recall that by definition of B, the observations of α and $B(\alpha)$ match, that is $y(t) = y'(t)$. From the property of Laplace distribution introduced in Section 2,

$$\begin{aligned}
&\prod_{i\in[N]k\in[n]}\frac{p_{M_t}(y_i^{(k)}(t) - x_i^{(k)}(t))}{p_{M_t}(y_i'^{(k)}(t) - x_i'^{(k)}(t))}\\
\leq &\prod_{i\in[N],k\in[n]} exp\left(\frac{|y_i^{(k)}(t) - x_i^{(k)}(t) - y_i'^{(k)}(t) + x_i'^{(k)}(t)|}{M_t}\right)\\
= &\prod_{i\in[N],k\in[n]} exp\left(\frac{|x_i^{(k)}(t) - x_i'^{(k)}(t)|}{M_t}\right)\\
= &exp\left(\sum_{i\in[N],k\in[n]}\frac{|x_i^{(k)}(t) - x_i'^{(k)}(t)|}{M_t}\right) \leq e^{\frac{\Delta(t)}{M_t}}.
\end{aligned} \tag{12}$$

If M_t satisfy $\sum_{t=0}^{\infty}\frac{\Delta(t)}{M_t} \leq \epsilon$, by combining it with Equation (11),(12),(9) and (10), we have

$$\begin{aligned}
\frac{\mathbb{P}[\mathcal{R}^{-1}(f, Y, x(0))]}{\mathbb{P}[\mathcal{R}^{-1}(f', Y, x(0))]} &= \frac{\int_{\alpha\in A}\mathbb{P}[\alpha]d\mu}{\int_{\alpha\in A}\mathbb{P}[B(\alpha)]d\mu}\\
&\leq \frac{\int_{\alpha\in A} e^{\sum_{t\in\mathbb{N}}\frac{\Delta(t)}{M_t}}\mathbb{P}[B(\alpha)]d\mu}{\int_{\alpha\in A}\mathbb{P}[B(\alpha)]d\mu} \leq e^{\epsilon}.
\end{aligned}$$

Letting the time bound T go to infinity, the same bound holds. Thus the lemma follows. □

Lemma 2 states that by adding noise drawn independently from a carefully designed Laplace distribution, the iterative distributed algorithm defined by Algorithms 2-4 guarantees ϵ-differential privacy. The parameters of the noise depend on the sensitivity of the algorithm. In the next lemma, we state a bound of the sensitivity of our proposed algorithm under Assumption 1,.

Lemma 4. *Under Assumption 1, the sensitivity of the proposed algorithm is*

$$\Delta(t) \leq 2C_2\sqrt{n}\gamma_t.$$

Proof. Fix any observation sequence ρ, any initial state $x(0) \in \mathcal{X}^N$ and any adjacent $\mathcal{P}, \mathcal{P}'$. Let $\mathcal{R}^{-1}(\mathcal{P}, \rho, x(0)) = x(0), \langle x(1), y(1), z(1), buffer(1)\rangle, \ldots$ and $\mathcal{R}^{-1}(\mathcal{P}', \rho, x(0)) = x'(0), \langle x'(1), y'(1), z'(1), buffer'(1)\rangle, \ldots$ be the executions for PDOP $\mathcal{P}$ and $\mathcal{P}'$ respectively. We will establish a bound on

$$||\mathcal{R}_{x(t)}^{-1}(\mathcal{P}, \rho, x(0)) - \mathcal{R}_{x(t)}^{-1}(\mathcal{P}', \rho, x(0))||_1.$$

Since the observation sequence ρ for both executions are identical, we have $y(t) = y'(t)$ for all t. From Algorithm 3, $z_i(t) = \sum_{j\in[N]} a_{ij}(t)y_j(t) = \sum_{j\in[N]} a_{ij}(t)y_j'(t) = z_i'(t)$ for each $i \in [N]$ and each round t. From Definition 3, f and f' are identical except for the i^{th} components. Thus, by applying Algorithm 4, we have:

$$\begin{aligned}
&||\mathcal{R}_{x(t)}^{-1}(\mathcal{P}, \rho, x(0)) - \mathcal{R}_{x(t)}^{-1}(\mathcal{P}', \rho, x(0))||_1\\
= &||z_i(t) - \gamma_t(\bigtriangledown f_i(z_i(t))) - z_i'(t) + \gamma_t(\bigtriangledown f_i'(z_i'(t)))||_1\\
= &\gamma_t||\bigtriangledown f_i(z_i(t))) - \bigtriangledown f_i'(z_i'(t)))||_1
\end{aligned}$$

From Assumption 1, the L^2 norm $||\bigtriangledown f_i(z_i(t))) - \bigtriangledown f_i'(z_i(t)))|| \leq 2C_2$. By the norm inequality introduced in Section 2, we have,

$$||\mathcal{R}_{x(t)}^{-1}(\mathcal{P}, \rho, x(0)) - \mathcal{R}_{x(t)}^{-1}(\mathcal{P}', \rho, x(0))||_1 \leq 2C_2\sqrt{n}\gamma_t.$$

Since the observation sequence ρ and the pair of adjacent PDOPs $\mathcal{P}, \mathcal{P}'$ can be chosen arbitrarily and the bound is oblivious of ρ, we have $\Delta(t) \leq 2C_2\sqrt{n}\gamma_t$. □

With Lemma 2 and 4, it directly follows that our algorithm guarantees ϵ-differential privacy.

Theorem 5. *The proposed algorithm (Algorithms 1-4) guarantees ϵ-differential privacy with any choice of $c > 0$, $q \in (0, 1)$ and $p \in (q, 1)$.*

Proof. Recall that in Equation (8), the step size at round t is $\gamma_t = cq^{t-1}$. Besides the Laplace noise at round t is drawn from distribution $Lap(M_t)$ with $M_t = 2C_2\sqrt{n}\frac{cp}{\epsilon(p-q)}p^{t-1}$. Then, from Lemma 4, we have

$$\Delta(t) \leq 2C_2\sqrt{n}\gamma_t = 2C_2\sqrt{n}cq^{t-1}.$$

Then, from $p \in (q, 1)$, we have:

$$\sum_{t=1}^{\infty} \frac{\Delta(t)}{M_t} \leq \frac{\epsilon(p-q)}{p} \sum_{t=1}^{\infty} \left(\frac{q}{p}\right)^{t-1} = \frac{\epsilon(p-q)}{p} \frac{p}{p-q} = \epsilon.$$

From Lemma 2, the algorithm guarantees ϵ-differential privacy □

4.3 Convergence

In this section, we prove that the algorithm converges following a similar idea presented in [11]. We define the transfer matrix $\Phi(k, s) = \prod_{t=s+1}^{k} A_t$ and $\Phi(s, s) = I$, which captures the evolution of states under a sequence of communication graph $\{A_t\}_{s+1}^{k}$. We denote $\Phi(k, s)_{i,j}$ as the entry of $\Phi(k, s)$ on the i^{th} row and j^{th} column. The following lemma (Lemma 3.2 of [11]) states that $\Phi(k, s)$ converges to a constant matrix as $k \to \infty$. Moreover, the convergence rate depends on: (i) the number of agents N, and (ii) the robust connectivity parameter η given in Assumption 2.

Lemma 6. *Under Assumption 2, there exist constants $\theta > 0$ and $\beta \in (0, 1)$ such that for any $i, j \in [N]$ for any naturals $t > s$,*

$$|\Phi(t, s)_{i,j} - \frac{1}{N}| \leq \theta \beta^{t-s},$$

where $\theta = \left(1 - \frac{\eta}{4N^2}\right)^{-2}$ and $\beta = 1 - \frac{\eta}{4N^2}$.

Lemma 6 states a fundamental restriction on the rate of convergence given a communication topology. We can observe from the above lemma that: as the number of agents (N) grows or the robustness of communication (η) decreases, β becomes closer to 1, that is, the transition matrix (Φ) converges slower.

Recall in Algorithm 3, agent j influences agent i's computation through the entry $a_{i,j}(t)$ of the communication graph A_t. Lemma 6 states that any two agents j and k has the same longterm influence on agent i's local state. As a direct result from this lemma, any two entries of $\Phi(t, s)$ converge to each other geometrically. That is, for any $i, j, k, l \in [N]$, $|\Phi(t, s)_{i,j} - \Phi(t, s)_{k,l}| \leq 2\theta\beta^{t-s}$. For the algorithm defined by Algorithms 2-4, we compute the distance between any two local state using the previous lemma.

Lemma 7. *Under Assumptions 1 and 2, for the proposed iterative distributed algorithm, for any agents $i, j \in [N]$ and any time $t \in \mathbb{N}$, the following holds:*

$$\begin{aligned} ||x_i(t) - x_j(t)|| \leq \quad & M_1\beta^t + M_2 \textstyle\sum_{s=1}^{t} \gamma_s \beta^{t-s} + \\ & M_3 \textstyle\sum_{s=1}^{t} \beta^{t-s} \sup_{k\in[N]} ||w_k(s)|| + \\ & M_4 \textstyle\sum_{s=1}^{t} \beta^{t-s} \gamma_s \sup_{k\in[N]} ||w_k(s)|| \end{aligned} \tag{13}$$

where $\beta \in (0, 1)$ is defined in Lemma 6 and $M_1, M_2, M_3, M_4 > 0$ are bounded constants depending on the constants C_1, C_2, C_3, C_4 introduced in Assumption 1.

Lemma 7 is proved by interactively unrolling Equation (5)-(7) and applying Lemma 6. The proof is presented in Appendix. With Lemma 7, we can bound the distance between two agents' local states by three terms. The first term $M_1\beta^t$ decays to 0 as t goes to infinity. The limits of the later terms can be derived using Proposition 1. This lemma suggests that the limit of Equation (1) depends on the limit of the noise magnitude as well as the limit of the step size. With Lemma 7 and Proposition 1, the convergence of Algorithm described in Section 4.1 follows directly.

Theorem 8. *The algorithm described in Section 4.1 converges.*

Proof. From Equation (8), we have that

$$\lim_{t\to\infty} \gamma_t = 0, \text{ and } \lim_{t\to\infty} \mathbb{E}||w_k(t)|| = 0.$$

Applying Proposition 1, we have $\lim_{t\to\infty} \sum_{s=1}^{t} \beta^{t-s}\gamma_s = 0, \lim_{t\to\infty} \sum_{s=1}^{t} \beta^{t-s} \sup_{k\in[N]} \mathbb{E}||w_k(s)|| = 0$, and $\lim_{t\to\infty} \sum_{s=1}^{t} \beta^{t-s}\gamma_s \sup_{k\in[N]} \mathbb{E}||w_k(s)|| = 0$. Then, by taking the limit of the expectation of Equation (13), we derive

$$\lim_{t\to\infty} \mathbb{E}||x_i(t) - x_j(t)|| = 0.$$

Thus the iterative distributed algorithm converges. □

Theorem 8 shows that our proposed algorithm converges, which requires the expected distance between local values of different agents to converge to 0. That is, the agents will eventually agree on a common value.

4.4 Accuracy

In this section, we establish bounds on the accuracy of the proposed iterative distributed algorithm. We first state a lemma which compares the sum of distance from $z_i(t)$ to any fixed point x' to that of distance from $x_i(t)$ to x'.

Lemma 9. *Fixed any point $x' \in \mathcal{X}$, for our proposed iterative distributed algorithm, for all $i \in [N]$, the following holds,*

$$\sum_{i\in[N]} ||z_i(t) - x'||^2 \leq \sum_{i\in[N]} ||x_i(t-1) - x' + w_i(t)||^2. \tag{14}$$

The proof is given in Appendix. We will derive a bound on the accuracy of the proposed algorithm (Theorem 10). This bound is derived using Lemma 9 and strong convexity.

Theorem 10. *The algorithm guarantees d-accuracy with*

$$\begin{aligned} d = \quad & 2C_1 e^{-\frac{C_3 c}{1-q}} + \frac{2C_2^2 c^2}{1-q^2} + \frac{8C_2^2 nc^2 p^2}{\epsilon^2 (p-q)^2 (1-p^2)} \\ & + \frac{16C_2^2 C_4^2 nc^4 p^2}{\epsilon^2 (p-q)^2 (1-p^2 q^2)}. \end{aligned} \tag{15}$$

Proof. We denote $u_i(t) = -\bigtriangledown f_i(z_i(t))$. Let x^* be the minimum of the problem. Taking 2-norm on both side of Equation (7) and applying the property of projection, we have

$$\begin{aligned} & ||x_i(t) - x^*||^2 \leq ||z_i(t) + \gamma_t u_i(t) - x^*||^2 \\ = \quad & ||z_i(t) - x^*||^2 + 2\gamma_t u_i^T(t)(z_i(t) - x^*) + \gamma_t^2 ||u_i(t)||^2. \end{aligned} \tag{16}$$

Since f_i is strongly convex, we have

$$\begin{aligned} u_i^T(t)(z_i(t) - x^*) \leq \quad & f_i(x^*) - f_i(z_i(t)) - \tfrac{C_3}{2}||z_i(t) - x^*||^2 \\ \leq \quad & -\tfrac{C_3}{2}||z_i(t) - x^*||^2. \end{aligned} \tag{17}$$

Combining Equation (16) and (17) we have

$$||x_i(t) - x^*||^2 \leq (1 - C_3\gamma_t)||z_i(t) - x^*||^2 + \gamma_t^2 ||u_t||^2.$$

Sum up above equations over $i \in [N]$ and divided by N, we have

$$\frac{1}{N} \sum_{i\in[N]} ||x_i(t) - x^*||^2 \leq \frac{1 - C_3\gamma_t}{N} \sum_{i\in[N]} ||z_i(t) - x^*||^2 + \gamma_t^2 ||u_t||^2. \tag{18}$$

We will replace the terms $||z_i(t) - x^*||^2$ using Lemma 9. From Equation (14), we have:

$$\begin{aligned}\sum_{i\in[N]} ||z_i(t) - x^*||^2 &\leq \sum_{i\in[N]} ||x_i(t-1) - x^* + w_i(t)||^2 \\ = \sum_{i\in[N]} ||x_i(t-1) - x^*||^2 &+ 2\sum_{i\in[N]} [(x_i(t-1) - x^*)^T w_i(t)] \\ &+ \sum_{i\in[N]} ||w_i(t)||^2\end{aligned}$$

Under the condition $w_i(t) \sim Lap(M_t)$, we have $\mathbb{E}[w_i(t)] = 0$ and $\mathbb{E}||w_i(t)||^2 = 2M_t^2$. Noticing that $w_i(t)$ and $x_i(t-1)$ are independent, we have:

$$\sum_{i\in[N]} \mathbb{E}||z_i(t) - x^*||^2 \leq \sum_{i\in[N]} \mathbb{E}||x_i(t-1) - x^*||^2 + 2NM_t^2. \tag{19}$$

For simplicity we denote $S(t) \triangleq \frac{1}{N}\sum_{i\in[N]} \mathbb{E}||x_i(t) - x^*||^2$. Combining Equation (18) and (19), we have:

$$S(t) \leq (1 - C_3\gamma_t)S(t-1) + \gamma_t^2\mathbb{E}||u_t||^2 + 2(1 - C_3\gamma_t)M_t^2 \tag{20}$$

We will establish a bound on $\mathbb{E}||u_t||^2$. By taking squared of Equation (28), we have

$$\begin{aligned}||u_s||^2 \leq \; & 2||\bigtriangledown f_k(\textstyle\sum_{j\in[N]} a_{kj}(t)x_j(s-1))||^2 + \\ & 2||\nabla^2 f_k(v)\textstyle\sum_{j\in[N]} a_{kj}(s)w_j(s)||^2\end{aligned}$$

Taking expectation on both sides yields,

$$\mathbb{E}||u_s||^2 \leq 2C_2^2 + 4C_4^2M_s^2 \tag{21}$$

Recursively applying Equation (20) and (21), we ultimately get:

$$\begin{aligned}S(t) \leq \; & \left[\textstyle\prod_{s=1}^t(1 - C_3\gamma_s)\right] S(0) \\ & + 2C_2^2\textstyle\sum_{s=1}^t \gamma_s^2 \prod_{l=s+1}^t(1 - C_3\gamma_l) \\ & + 2\textstyle\sum_{s=1}^t M_s^2 \prod_{l=s}^t(1 - C_3\gamma_l) \\ & + 4C_4^2\textstyle\sum_{s=1}^t M_s^2\gamma_s^2 \prod_{l=s+1}^t(1 - C_3\gamma_l).\end{aligned} \tag{22}$$

We define $\Psi(k,s) \triangleq \prod_{t=s+1}^k(1 - C_3\gamma_t)$. From Assumption 1,we have that $S(0) \leq 2C_1$. Thus, we have

$$\begin{aligned}S(t) \quad \leq \quad & 2C_1\Psi(t,0) + 2C_2^2\sum_{s=1}^t \gamma_s^2\Psi(t,s) \\ & + 2\sum_{s=1}^t M_s^2\Psi(t,s-1) + 4C_4^2\sum_{s=1}^t \gamma_s^2M_s^2\Psi(t,s-1).\end{aligned}$$

The above equation has three terms, each of which involves $\Psi(k,s)$.We will give a bound to the term $\Psi(k,s)$. Since $\Psi(k,s)$ is the product of factors no larger than 1, $\Psi(k,s) \leq 1$ by definition. Thus, the above inequality reduces to

$$S(t) \leq 2C_1\Psi(t,0) + 2C_2^2\sum_{s=1}^\infty \gamma_s^2 + 2\sum_{s=1}^\infty M_s^2 + 4C_4^2\sum_{s=1}^\infty \gamma_s^2M_s^2.$$

Substituting Equation (8) into the right-hand side, we have,

$$\begin{aligned}S(t) \leq \; & 2C_1\Psi(t,0) + \tfrac{2C_2^2c^2}{1-q^2} + \tfrac{8C_2^2nc^2p^2}{\epsilon^2(p-q)^2(1-p^2)} \\ & + \tfrac{16C_2^2C_4^2nc^4p^2}{\epsilon^2(p-q)^2(1-p^2q^2)}.\end{aligned} \tag{23}$$

To compute a upper bound on term $\Psi(t,0)$, we use a standard property of exponential function, that is, $1 - a \leq e^{-a}$ for any $a \in \mathbb{R}$. Thus

$$\Psi(t,0) = \prod_{s=1}^t(1 - C_3\gamma_t) \leq e^{-\sum_{s=1}^t C_3\gamma_t} \leq e^{-\frac{C_3c(1-q^t)}{1-q}}.$$

Substitute the above inequality into Equation (23), we have:

$$\begin{aligned}S(t) \leq \; & 2C_1e^{-\frac{C_3c(1-q^t)}{1-q}} + \tfrac{2C_2^2c^2}{1-q^2} + \tfrac{8C_2^2nc^2p^2}{\epsilon^2(p-q)^2(1-p^2)} \\ & + \tfrac{16C_2^2C_4^2nc^4p^2}{\epsilon^2(p-q)^2(1-p^2q^2)}.\end{aligned}$$

By triangular inequality, we have

$$\begin{aligned}\mathbb{E}||\bar{x}(t) - x^*||^2 &= \mathbb{E}||\frac{1}{N}\sum_{i\in[N]} x_i(t) - x^*||^2 \\ \leq \quad \frac{1}{N}\sum_{i\in[N]} \mathbb{E}||x_i(t) - x^*||^2 &= S(t)\end{aligned}$$

Letting $t \to \infty$, we have

$$\begin{aligned}\limsup_{t\to\infty} \mathbb{E}||\bar{x}(t) - x^*||^2 \leq \; & 2C_1e^{-\frac{C_3c}{1-q}} + \tfrac{2C_2^2c^2}{1-q^2} + \tfrac{8C_2^2nc^2p^2}{\epsilon^2(p-q)^2(1-p^2)} \\ & + \tfrac{16C_2^2C_4^2nc^4p^2}{\epsilon^2(p-q)^2(1-p^2q^2)}.\end{aligned}$$

Thus the theorem follows. □

In the above theorem, we derived a bound of the accuracy the algorithm guarantees. The first term in the bound depends on the size of domain $\mathcal{X}$, which is exponentially decaying. This bound depends on the four parameters ϵ, c, p, q. Fixing other three parameter, the accuracy has the order of $d \sim O(\frac{1}{\epsilon^2})$ for small ϵ. As ϵ converges to 0, that is, for complete privacy for individuals, the accuracy becomes arbitrarily bad.

4.5 Experiment and Discussion

The algorithm has four parameters: the privacy level ϵ, the initial step size c, the step size decay rate q and the noise decay rate p. We have established that the algorithm guarantees ϵ-differential privacy for any choice of parameters. If we fix the privacy level ϵ, the dependency of the accuracy level of the algorithm on each of the other three parameters based on the partial derivative of d. Since the accuracy level d is not convex on c, q, p, the global optimal choice of the parameters does not have a clean close form expression. However, we observe that if we fix any other two parameters, the other parameter has a local optima: In practice, we can tune the parameters with the following heuristic: (i) pick c, q, p randomly initially, (ii) fix two parameters and tune the remaining parameter to the local optima, and (iii) repeat step (ii) several times with different choice of parameters to be tuned. We use the proposed algorithm to solve Example 1 where the parameters (c, q, p) are tuned with the above heuristic.

Example 2 We solve a version of Example 1 with seven different privacy levels: $\epsilon = 0.1, 0.2, 0.5, 1, 2, 5$ and 10. We assign the domain of optimization $\mathcal{X}$ as the unit square $\mathcal{X} = \{(x, y) \in \mathbb{R}^2 | -1 \leq x, y \leq 1\}$. For each privacy level ϵ, we first decide the parameters (c, q, p) using the proposed heuristic, and then solve the DPOP repeatedly for 5000 times. Each time, we record the squared distance from the convergent point to the optima. Then, the accuracy level d of a privacy level is approximated by the average of

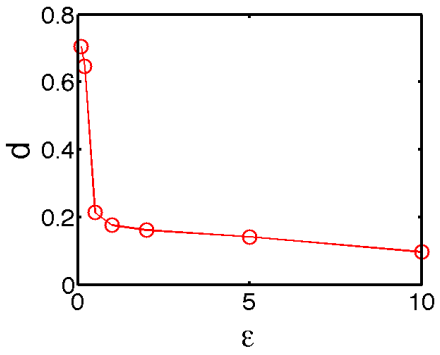

Figure 1: Accuracy level d as a function of privacy level ϵ for Example 2.

the squared distances over the 5000 runs. The experimental results are illustrated in Fig 1. □

5. CONCLUSION

We formulated the private distributed optimization (PDOP) problem in which N agents are required to minimize a global cost function f that is the sum $\sum_{i=1}^{N} f_i$ of N cost functions for the individual agents. The agents may exchange information about their estimates for the optimal solution, but are required to keep their cost functions differentially private from an adversary with access to all the communication. We studied structurally simple iterative distributed algorithms for solving PDOP. Like other iterative algorithms for consensus and optimization, our algorithm proceeds in rounds. In each round, however, an agent first adds a vector of carefully chosen random noise to its current estimate for the optimal point and broadcasts this noisy estimate to its neighbors. The noise is chosen from a Laplace distribution that converges to the Dirac distribution with increasing number of rounds. In the second phase, the agent updates its estimate by (a) taking a weighted average of the noisy estimates it received from its neighbors and (b) moving the estimate, by a carefully chosen step-size, in opposite direction of the gradient of its own cost function. The communication topology and hence the neighbors of an agent may change from one round to another, yet, this structurally simple algorithm solves PDOP. We establish its differential privacy as well as its approximate convergence to the optimal point. The analysis also reveals the dependence of the accuracy and the privacy levels of the algorithm on the the noise and the step-size parameters. We observe that, by fixing other parameters, the accuracy level has the order of $O(\frac{1}{\epsilon^2})$.

Accurately solving distributed coordination problems requires information sharing. Participants in the coordination might be willing to sacrifice on the quality of the solution provided this loss is commensurate with the gain in the level of privacy of their individual preferences. Thus, a natural question is to quantify the cost incurred in solving the problem as a function of the privacy level. In this paper, we have addressed this question in the context of PDOP and the class of iterative algorithms. Even for the class of iterative algorithms, establishing a lower-bound on the maximum level of differential privacy that can be achieved for a certain level of accuracy remains an open problem.

6. REFERENCES

[1] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *J. Mach. Learn. Res.*, 12:1069–1109, July 2011.

[2] A. Dominguez-Garcia and C. Hadjicostis. Distributed matrix scaling and application to average consensus in directed graphs. *Automatic Control, IEEE Transactions on*, 58(3):667–681, 2013.

[3] A. D. Domınguez-Garcıa, S. T. Cady, and C. N. Hadjicostis. Decentralized optimal dispatch of distributed energy resources. In *IEEE Conf. on Decision and Control,(Maui, HI)*, 2012.

[4] C. Dwork. Differential privacy. In *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, 2006.

[5] C. Dwork. Differential privacy: a survey of results. In *Proceedings of the 5th international conference on Theory and applications of models of computation*, TAMC'08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.

[6] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006*, pages 486–503. Springer, 2006.

[7] C. Dwork, M. Naor, G. Rothblum, and T. Pitassi. Differential privacy under continual observation. In *Proceedings of the 42nd ACM symposium on Theory of computing*, 2010.

[8] Z. Huang, S. Mitra, and G. Dullerud. Differentially private iterative synchronous consensus. In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, WPES '12, pages 81–90, New York, NY, USA, 2012. ACM.

[9] D. Kifer, A. Smith, and A. Thakurta. Private convex empirical risk minimization and high-dimensional regression. *Journal of Machine Learning Research*, 1:41, 2012.

[10] S. Mitra. *A Verification Framework for Hybrid Systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA 02139, September 2007.

[11] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48–61, 2009.

[12] Y. NESTEROV. Gradient methods for minimizing composite objective function. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.

[13] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[14] R. Segala. Modeling and verification of randomized distributed real-time systems. 1996.

[15] S. Sundhar Ram, A. Nedic, and V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optimization Theory and Applications*, 147(3):516–545, 2010.

[16] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *Automatic Control, IEEE Transactions on*, 31(9):803–812, 1986.

APPENDIX

Proof of Lemma 7:

Proof. For brevity, we denote

$$u_i(t) = Proj_{\mathcal{X}}[z_i(t) - \gamma_t \bigtriangledown f_i(z_i(t))] - z_i(t).$$

Then, we can rewrite Equation (7) as

$$x_i(t) = \sum_{j\in[N]} a_{ij}(t)x_j(t-1) + \sum_{j\in[N]} a_{ij}(t)w_j(t) + u_i(t).$$

By the property of projection, we have Recursively apply the above equation, we have:

$$\begin{aligned} x_i(t) = &\textstyle\sum_{j\in[N]} \Phi(t,0)_{i,j}x_j(0) + \sum_{s=1}^{t}\sum_{j\in[N]} \Phi(t,s-1)_{i,j}u_j(s) \\ &+ \textstyle\sum_{s=1}^{t}\sum_{j\in[N]} \Phi(t-1,s-1)_{i,j}w_j(s). \end{aligned} \tag{24}$$

Thus, the distance between two local states $x_i(t)$ and $x_j(t)$ is:

$$\begin{aligned} &||x_i(t) - x_j(t)|| \\ = &\sum_{k\in[N]} |\Phi(t,0)_{i,k} - \Phi(t,0)_{j,k}|||x_k(0)|| + \\ &\sum_{s=1}^{t}\sum_{k\in[N]} |\Phi(t,s-1)_{i,k} - \Phi(t,s-1)_{j,k}|||u_k(s)|| + \\ &\sum_{s=1}^{t}\sum_{k\in[N]} |\Phi(t-1,s-1)_{i,k} - \Phi(t-1,s-1)_{j,k}|||w_k(s)||. \end{aligned}$$

By applying Lemma 6, the above expression becomes

$$\begin{aligned} ||x_i(t) - x_j(t)|| \leq \; & 2N\theta\beta^t \sup_{k\in[N]} ||x_k(0)|| + \\ & 2N\theta \textstyle\sum_{s=1}^{t} \beta^{t-s+1} \sup_{k\in[N]} ||u_k(s)|| + \\ & 2N\theta \textstyle\sum_{s=1}^{t} \beta^{t-s} \sup_{k\in[N]} ||w_k(s)||. \end{aligned} \tag{25}$$

From Assumption 1, we have $||x_k(0)||$ is bounded. Since the distance between $z_k(s)$ and the set $\mathcal{X}$ is bounded by $\sum_{j\in[N]} a_{kj}(t)||w_j(t)||$, from the property of projection,

$$\begin{aligned} ||u_k(s)|| = \; & ||Proj_{\mathcal{X}}[z_i(t) - \gamma_t \bigtriangledown f_i(z_i(t))] - z_i(t)|| \\ \leq \; & ||Proj_{\mathcal{X}}[z_k(s) - \gamma_t \bigtriangledown f_k(z_k(s))] - Proj_{\mathcal{X}}[z_k(s)]|| \\ & + ||Proj_{\mathcal{X}}[z_k(s)] - z_k(s)|| \\ \leq \; & \gamma_t|| \bigtriangledown f_k(z_k(s))|| + \textstyle\sum_{j\in[N]} a_{kj}(t)||w_j(t)|| \end{aligned} \tag{26}$$

Since $z_k(s) = \sum_{j\in[N]} a_{kj}(t)(x_j(s-1) + w_j(s))$, by mean-value theorem, there exists a point v between $z_k(s)$ and $\Sigma_{j\in[N]} a_{kj}(t)x_j(s-1)$, such that

$$\begin{aligned} \bigtriangledown f_k(z_k(s)) = \; & \bigtriangledown f_k(\textstyle\sum_{j\in[N]} a_{kj}(t)x_j(s-1)) + \\ & \nabla^2 f_k(v) \textstyle\sum_{j\in[N]} a_{kj}(s)w_j(s)) \end{aligned} \tag{27}$$

Since $\sum_{j\in[N]} a_{kj}(s)x_j(s-1) \in \mathcal{X}$, from with Assumption 1 we have

$$|| \bigtriangledown f_k(z_k(s))|| \leq C_2 + C_4 \textstyle\sum_{j\in[N]} a_{kj}(s)||w_j(s))||. \tag{28}$$

Combining Equation (26) and (28), we have

$$||u_k(s)|| \leq C_2\gamma_s + (C_4\gamma_s + 1) \sum_{j\in[N]} a_{kj}(s)||w_j(s))||.$$

Since A_s is doubly stochastic, we have $\sum_{j\in[N]} a_{kj}(s) = 1$. Thus the above equation is reduced to

$$||u_k(s)|| \leq C_2\gamma_s + (C_4\gamma_s + 1) \sup_{k\in[N]} ||w_k(s))|| \tag{29}$$

Combining Equation (25) and (29) we have,

$$\begin{aligned} ||x_i(t) - x_j(t)|| \leq \; & M_1\beta^t + M_2 \textstyle\sum_{s=1}^{t} \gamma_s\beta^{t-s} + \\ & M_3 \textstyle\sum_{s=1}^{t} \beta^{t-s} \sup_{k\in[N]} ||w_k(s)|| + \\ & M_4 \textstyle\sum_{s=1}^{t} \beta^{t-s}\gamma_s \sup_{k\in[N]} ||w_k(s)||, \end{aligned} \tag{30}$$

where $M_1 = 2N\theta \sup_{x\in\mathcal{X}} ||x||$, $M_2 = 2NC_2\theta\beta$, $M_3 = 2N\theta(1+\beta)$ and $M_4 = 2NC_4\theta\beta$.

□

Proof of Lemma 9:

Proof. From Equation (5)-(6), we have $z_i(t) = \sum_{j\in[N]} a_{i,j}(t)(x_j(t-1) + w_j(t))$. It follows that,

$$\begin{aligned} \textstyle\sum_{i\in[N]} ||z_i(t) - x'||^2 = \; & \textstyle\sum_{i\in[N]} ||\sum_{j\in[N]} a_{i,j}(t)(x_j(t-1) \\ & + w_j(t)) - x'||^2 \end{aligned} \tag{31}$$

From the assumption that the matrix A_t is doubly stochastic, we have$\sum_{j\in[N]} a_{i,j}(t) = 1$. So we have $x' = \sum_{j\in[N]} a_{i,j}(t)x'$. Applying this trick to Equation (31), we have

$$\sum_{i\in[N]} ||z_i(t) - x'||^2 = \sum_{i\in[N]} || \sum_{j\in[N]} a_{i,j}(t) \left(x_j(t-1) + w_j(t) - x'\right) ||^2. \tag{32}$$

By triangle inequality and reordering of summation, we have

$$\begin{aligned} & || \textstyle\sum_{j\in[N]} a_{i,j}(t) \left(x_j(t-1) + w_j(t) - x'\right) ||^2 \\ \leq \; & \textstyle\sum_{i\in[N]} \sum_{j\in[N]} a_{i,j}(t)||x_j(t-1) + w_j(t) - x'||^2. \end{aligned} \tag{33}$$

Again from the double stochasticity of A_t, $\sum_{i\in[N]} a_{i,j}(t) = 1$. Then the above expression can be reduced to

$$\begin{aligned} & \sum_{j\in[N]} \sum_{i\in[N]} a_{i,j}(t)||x_j(t-1) + w_j(t) - x'||^2 \\ = \; & \sum_{j\in[N]} ||x_j(t-1) + w_j(t) - x'||^2. \end{aligned}$$

Combining above equation with Equations (32) and (33), we derive

$$\sum_{i\in[N]} ||z_i(t) - x'||^2 \leq \sum_{j\in[N]} ||x_j(t-1) - x' + w_j(t)||^2.$$

By changing the variable of the right-hand side, the lemma follows. □