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 $\mathbb{R}^N$, where the cost function is expressed as a sum $\sum f_i$. In this paper, we study the private distributed optimization (PDOP) problem 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 the optimal value. Our analysis reveals the dependence of the achieved accuracy and the privacy levels on the the parameters of the algorithm.

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. We study iterative 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 [6, 8, 9]), to our knowledge this paper is the first attempt to achieve this goal while maintaining privacy. The notion of privacy we adopt is derived from ϵ-differential privacy [1, 2] applied to continuous bit streams in [3]. 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 this paper, we study a class of synchronous iterative distributed algorithms for solving PDOP. Iterative algorithms proceed in round. In each round, an 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 the other 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 step size. The step sizes are chosen so that they decay down to zero over rounds.

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 1). 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 [8]), 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 [4], 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 optimality of the algorithm (Theorems 4 and 10).

The algorithm has four parameters: the initial noise, the

noise decay rate, the initial step size and the step size decay rate. By choosing these parameters appropriately, the algorithm guarantees either ϵ-differential privacy for arbitrary $\epsilon > 0$ or d-accurate optimization for arbitrary $d > 0$. Our analysis reveals in detail the influence of these parameters on the accuracy level and the privacy level of the algorithm.

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 $p_c(x) \triangleq \frac{1}{2c} e^{-|x|/c}$. This distribution has mean zero and variance $2c^2$. For any $x, y \in \mathbb{R}$, it can be shown that $\frac{p_c(x)}{p_c(y)} \leq exp(\frac{|y-x|}{c})$.

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, $||v|| = \sqrt{v^T v}$. Recall that 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 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 $z \in \mathbb{R}^n \setminus \mathcal{X}$. 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*. Strongly convex functions on compact domains have unique minima [7].

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) $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ is a set of real-valued, strongly convex and differentiable individual cost functions on domain $\mathcal{X}$, (iii) $f : \mathcal{X} \mapsto \mathbb{R}$ is the global cost function which is a sum of N cost functions in $\mathcal{F}$, that is, $f(x) \triangleq \sum f_i(x)$ with $f_i \in \mathcal{F}$ for each $i \in [N]$, and (iv) $\mathcal{A} = \{A_t\}_{t \in \mathbb{N}}$ 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 as the i^{th} additive term of the global cost f. The individual cost f_i is only known to agent i. Together the agents aim to minimize:

$$f(x) = \min \sum_{i \in [N]} f_i(x), \tag{1}$$

subject to the constraint $x \in \mathcal{X}$. We define $OPT_{\mathcal{P}} \triangleq \min_{x \in \mathcal{X}} f(x)$ 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 $\mathcal{X}_{\mathcal{P}}, \mathcal{F}_{\mathcal{P}}, f_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}}, OPT_{\mathcal{P}}$ 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 $\mathcal{X}, \mathcal{F}$, ..., and $\mathcal{X}'$, $\mathcal{F}'$, etc.

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 $C_1 \triangleq \sup_{x \in \mathcal{X}} ||x||$ denote the bound on $\mathcal{X}$.*

(ii) The gradients of all the individual cost function are uniformly bounded. That is, there exists $C_2 > 0$ such that for any $x \in \mathcal{X}$ and any $g \in \mathcal{F}$, $|| \bigtriangledown g(x)|| \leq C_2$. Since the functions in $\mathcal{F}$ are strongly convex, there also exists $C_3 > 0$ such that for any $x, y \in \mathcal{X}$ and for any $g \in \mathcal{F}$, $\bigtriangledown f(x)^T(y - x) \leq f(y) - f(x) - \frac{C_3}{2}||y - x||^2$.

The first part of Assumption 1 is standard which guarantees the existence of an optimal solution. The second part of the above assumption holds if the magnitude of the gradients of individual cost functions do not grow unbounded. In many optimization problems, it is standard to assume the cost function to be convex, for which gradient based method is effective. Strong convexity provides a stronger bound on the gradient term $\bigtriangledown f(x)^T(y - x)$ which is necessary of analyzing the accuracy of our algorithm.

We assume a synchronous model of distributed computation though the communication network among the agents is time varying. 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, and (iii) $\mathcal{W}_t : \mathcal{E}_t \mapsto (0, 1]$ is the weighted function 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, this assumption ensures each agent's decision has an equal influence on the final decision. This statement will become clearer after we introduce the algorithm. We use the following assumption on the robustness 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 robustness parameter $\eta \in (0, 1]$ such that for each $t \in \mathbb{N}$:*

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

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

This assumption guarantees that there exists a path in the graph linking each pair of the agents. Moreover, the sum of weights along the path is lower bounded.

3.2 Iterative Distributed Algorithms for PDOP

We study an 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, which are (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: Algorithmic 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 R
```

Message exchanging between agents is assumed to be atomic. That is, the **Receive**(y_i) routine of agent i broadcasts y_i to all his neighbors and the **Receive**() routine receives all neighbors' broadcasts immediately. 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 at the end of round t. We denote the aggregate state $x(t)$ 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 an infinite sequence of the form $\alpha = x(0), \langle x(1), y(1), z(1), buffer(1)\rangle, \langle x(2), y(2), z(2), buffer(2)\rangle, \ldots$. The observable part of such an execution are the corresponding infinite sequence of messages $y(1), y(2), \ldots$. We denote the observation mapping $\mathcal{R}(\alpha) \triangleq y(1), y(2), \ldots$.

Note that the set of messages stored in $buffer_i(t)$ 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 these subroutines. 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 [5]).

3.3 Convergence, Accuracy and Differential Privacy

An iterative distributed algorithm solves the PDOP problem if the estimates of all the agents converge to the optimal point of f and the algorithm preserves differential privacy of the f_i's.

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 value of the global cost function evaluated at the average.

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

$$\lim_{t\to\infty} \mathbb{E}[f(\bar{x}(t))] \leq f^*_{\mathcal{P}} + d,$$

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 [3] 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. Formally, for an iterative distributed algorithm, there is an inverse observation mapping $\mathcal{R}^{-1}$ that maps from (i) a PDOP $\mathcal{P}$, (ii) an observation sequence $\rho \in Obs$, and (iii) an initial state vector $x(0)$ to the set of corresponding executions $\{\alpha \in \Xi_{\mathcal{P}} : \mathcal{R}(\alpha) = \rho \wedge \alpha(0) = x(0)\}$.

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

[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.

(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* $\epsilon \geq 0$, *the iterative distributed algorithm is* ϵ-differentially private, *if for any two adjacent PDOPs* $\mathcal{P}$ *and* $\mathcal{P}'$, *any set of observation sequences* $Y \subseteq Obs$ *and any initial state* $\Theta \in \mathcal{X}^N$

$$\mathbb{P}[\mathcal{R}^{-1}(\mathcal{P}, Y, \Theta)] \leq e^{\epsilon}\mathbb{P}[\mathcal{R}^{-1}(\mathcal{P}', Y, \Theta)], \tag{2}$$

where the expectation 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. The probability measure is over the σ-algebra of of cones over the set of executions. 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 subroutine R, shown in Algorithm 2, computes a value to broadcast based on agent i's estimate 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. Lemma 1 shows how to select an M_t. The value to report is the sum of x_i and the noise vector w_i.

Algorithm 2: Subroutine R

Input: x_i, t
$w_i \sim Lap(M_t)$;
$y_i = x_i + w_i$;
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 stores agent j's broadcast into a local variable y_j if it presents in agent i's buffer, otherwise let $y_j \triangleq$ 0s Recall that $a_{i,j}(t)$ is the entry on the i^{th} column and j^{th} row of a 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.

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

Algorithm 3: Subroutine F

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

z_i. The parameter γ_t is the step size at round t. The projection $Proj_{\mathcal{X}}$ guarantees that the estimate for agent i is in $\mathcal{X}$.

Algorithm 4: Subroutine U

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

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

$$y_i(t) = x_i(t-1) + w_i(t) \tag{3}$$

$$z_i(t) = \sum_{j\in[N]} a_{ij}(t)y_j(t). \tag{4}$$

$$x_i(t) = Proj_{\mathcal{X}}[z_i(t) - \gamma_t(\bigtriangledown f_i(z_i(t)))]. \tag{5}$$

There are two undefined parameters: the noise parameter M_t and the step size γ_t based on the round t. We propose to choose these parameter as:

$$\begin{aligned} M_t &= c_1 q_1^{t-1} \\ \gamma_t &= c_2 q_2^{t-1}. \end{aligned} \tag{6}$$

where $c_1 > 0$ is the initial noise parameter, $q_1 \in (0, 1)$ is the noise decay rate, $c_2 \in (0, \frac{1}{C_3})$ is the initial step size and $q_2 \in (0, q_1)$ is the step size decay rate (C_3 is a constant introduced in Assumption 1). As a result, the noise distribution converges to Dirac distribution and the step size converges to 0. Later we show that by choosing these two parameters geometrically decaying on t, the iterative distributed algorithm is convergent and ensure certain level of accuracy and differential privacy.

4.2 Differential Privacy

Recall in Definition 3, two PDOP $\mathcal{P}$ and $\mathcal{P}'$ are said to be 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 PDOP. 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 of the set of the executions $\alpha \in \mathcal{R}^{-1}(\mathcal{P}, \rho, x(0))$.

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

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

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 1. *At each round $t \in \mathbb{N}$, if each agent adds a noise vector $\omega_i(t)$ consisting 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. Fix any pair of adjacent PDOP $\mathcal{P}$ and $\mathcal{P}'$, any set of observation sequence Obs and any initial state $x(0) \in \mathcal{X}$. For simplicity, we denote the sets of executions $\mathcal{R}^{-1}(\mathcal{P}, Obs, \Theta)$ and $\mathcal{R}^{-1}(\mathcal{P}', Obs, \Theta)$ by A and A' respectively. First we introduce a proposition of the uniqueness of the mapping $\mathcal{R}^{-1}$. The proof can be found in appendix.

Proposition 2. *For any PDOP $\mathcal{P}$, any observation sequence $\rho \in Obs$, for any initial state Θ, $\mathcal{R}^{-1}(\mathcal{P}, \rho, \Theta)$ is a singleton set.*

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 Obs, 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'.

$$\frac{\mathbb{P}[\mathcal{R}^{-1}(f, Obs, x(0))]}{\mathbb{P}[\mathcal{R}^{-1}(f', Obs, x(0))]} = \frac{\int_{\alpha \in A} \mathbb{P}[\alpha] d\mu}{\int_{\alpha' \in A'} \mathbb{P}[\alpha'] d\mu'}. \tag{7}$$

Changing the variable using the bijection B we have,

$$\int_{\alpha' \in A'} \mathbb{P}[\alpha'] d\mu' = \int_{B(\alpha) \in A'} \mathbb{P}[B(\alpha)] d\mu = \int_{\alpha \in A} \mathbb{P}[B(\alpha)] d\mu \tag{8}$$

From Algorithm 2-4, recall that we fixed the observation sequence ρ, the probability comes from the noise $w_i(t)$. That is,

$$\int_{\alpha \in A} \mathbb{P}[\alpha] d\mu = \int_{\alpha \in A} \mathbb{P}[\xi|\rho] d\mu.$$

where ξ is the sequence of state vector $x(t)$ corresponding to α; ρ is the corresponding observation sequence. Along the sequence ξ, $x_i(t)$ 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 independent noise to $x(t)$, from the distribution $Lap(M_t)$, it follows that the probability density of an execution is reduced to

$$\mathbb{P}[\xi|\rho] = \prod_{\substack{i \in [N], k \in [n] \\ t \in \mathbb{N}}} p_{M_t}(y_i^{(k)}(t) - x_i^{(k)}(t)), \tag{9}$$

where $p_b(x)$ is the probability density function of $Lap(b)$ at x. Then, we relate the distance at time t between the state of α and $B(\alpha)$ with the sensitivity $\Delta(t)$. By the 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,

$$\begin{aligned}
&\prod_{i \in [N] k \in [n]} \frac{p_{M_t}(y_i^{(k)}(t) - x_i^{(k)}(t))}{p_{M_t}(y_i'^{(k)}(t) - x_i'^{(k)}(t))} \\
\leq &\prod_{i \in [N], k \in [n]} exp\left(\frac{|y_i^{(k)}(t) - x_i^{(k)}(t) - y_i'^{(k)}(t) + x_i'^{(k)}(t)|}{M_t}\right) \\
= &\prod_{i \in [N], k \in [n]} exp\left(\frac{|x_i^{(k)}(t) - x_i'^{(k)}(t)|}{M_t}\right) \\
= &exp\left(\sum_{i \in [N], k \in [n]} \frac{|x(\alpha(t)) - x(B(\alpha)(t))|}{M_t}\right) \leq e^{\frac{\Delta(t)}{M_t}}.
\end{aligned} \tag{10}$$

Combining Equation (7), (8), (9) and (10), we derive

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

If M_t satisfy $\sum_{t=0}^{\infty} \frac{\Delta_D(t)}{M_t} \leq \epsilon$, then $\prod_{t \in \mathbb{N}} e^{\frac{\Delta(t)}{M_t}} \leq e^{\epsilon}$. Thus the lemma follows. □

Lemma 1 states that by adding Laplace noises drawn independently from some Laplace distribution, the iterative distributed algorithm defined by Algorithm 2-4 guarantees ϵ-differential privacy. The parameters of the noise to add depends on the sensitivity of the algorithm. In the next lemma, we state a bound of the sensitivity of our proposed algorithm.

Lemma 3. *If Assumption 1 holds, the sensitivity of the proposed algorithm is*

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

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

By fixing the observation sequence ρ for both executions, 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:

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

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,

$$\Delta(t) = \sup_{\substack{x \in \mathcal{R}_{x(t)}^{-1}(\mathcal{P}, \rho, x(0), t) \\ x' \in \mathcal{R}_{x(t)}^{-1}(\mathcal{P}', \rho, x(0), t)}} \gamma_t || \bigtriangledown f_i(z_i(t))) - \bigtriangledown f_i'(z_i(t)))||_1 \leq 2C_2 \sqrt{n} \gamma_t.$$

□

With Lemma 1 and 3, it directly follows that our algorithm guarantees ϵ-differential privacy.

Theorem 4. *The proposed algorithm guarantees ϵ-differential privacy with $\epsilon = \frac{2C_2\sqrt{n}c_2q_1}{c_1(q_1-q_2)}$.*

Proof. Recall that in Section 4.1, the Laplace noise at time t is drawn from distribution $Lap(M_t)$ with $M_t = c_1q_1^{t-1}$. Besides, $\gamma_t = c_2q_2^{t-1}$ with $q_2 \in (0, q_1)$. Then $\sum_{t=1}^{\infty} \frac{\Delta}{M_t} \leq \sum_{t=1}^{\infty} \frac{2C_2\sqrt{n}c_2q_2^{t-1}}{c_1q_1^{t-1}} = \frac{2C_2\sqrt{n}c_2q_1}{c_1(q_1-q_2)}$ is bounded. From Lemma 1, the algorithm guarantees ϵ-differential privacy with $\epsilon = \frac{2C_2\sqrt{n}c_2q_1}{c_1(q_1-q_2)}$. □

We can observe from the above theorem that $\frac{2C_2\sqrt{n}c_2q_1}{c_1(q_1-q_2)} \to 0$ by letting $c_2 \to 0$. Thus, we conclude that by choosing the parameters (c_1, c_2, q_1, q_2) properly, the iterative distributed algorithm guarantees any level of ϵ-differential privacy.

4.3 Convergence

In this section, we prove that the algorithm guarantees convergence. We define the transfer matrix $\Phi(k, s) = \prod_{t=s+1}^{k} A(t)$, 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 dues to [6] states that $\Phi(k, s)$ convergence to a constant matrix as $k \to \infty$.

Lemma 5. *If Assumption 2 holds, there exist $\theta > 0$ and $\beta \in (0, 1)$ such that for any $i, j \in [N]$,*

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

Recall in Algorithm 3, agent j influences agent i's computation through the entry $a_{i,j}(t)$ of the communication graph A_t. Roughly, Lemma 5 states that any two agents j and k has the same longterm influence on agent i's local state under our assumption of the robustness of connectivity of communication graphs. 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 Algorithm 2-4, we compute the distance between any two local state using the previous lemma.

Lemma 6. *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:*

$$||x_i(t)-x_j(t)|| \leq 2NC_1\theta\beta^t+2NC_2\theta\sum_{s=1}^{t}\gamma_s\beta^{t-s}+2N\theta\sum_{s=1}^{t}\beta^{t-s+1}||w_k(s)||$$

where $\theta > 0$ and $\beta \in (0, 1)$ are defined in Lemma 5.

The proof can be found in appendix. The above lemma bounds the distance between two agents' local states by three terms. The first term $2NC_1\theta\beta^t$ dies down. The later two terms have the form of convolutions. The later lemma determines the limits of a convolution.

Lemma 7. *For a constant $\beta \in (0, 1)$ and a convergent scalar sequence $\{a_k\}$ such that $\lim_{k\to\infty} a_k = 0$, the following holds:*

$$\lim_{n\to\infty}\sum_{k=1}^{n}\beta^{n-k}a_k = 0. \tag{11}$$

Proof of Lemma 7 is shown in appendix. This lemma suggests that the limit of Equation (17) depends on the limit of the noise magnitude as well as the limit of the step size. With Lemma 6 and 7, the convergence of Algorithm described in Section 4.1 follows directly.

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

Proof. From Lemma 6, $||x_i(t)-x_j(t)|| \leq 2NC_1\theta\beta^t+2NC_2\theta\sum_{s=1}^{t}\gamma_s\beta^{t-s}+2N\theta\sum_{s=1}^{t}\beta^{t-s+1}||w_k(s)||$. From Section 4.1, $\lim_{t\to\infty}\gamma_t = 0$ and $\lim_{t\to\infty}\mathbb{E}||w_k(t)|| = 0$. From Lemma 7, we have

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

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 value as a solution of the optimization problem.

4.4 Accuracy

In this section, we discuss whether the local values eventually minimize the global cost function by running the 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,*

$$\sum_{i\in[N]}||z_i(t) - x'||^2 \leq \sum_{i\in[N]}||x_i(t-1) - x' + w_i(t)||^2. \tag{12}$$

We will state our last main theorem which shows that our proposed algorithm guarantees d-accuracy. The accuracy guarantee shown in Theorem 10 has a complected expression. Later we will show that the algorithm can be arbitrarily accurate by properly choosing the algorithm parameters (c_1, c_2, q_1, q_2).

Theorem 10. *The proposed iterative distributed algorithm guarantees d-accuracy with*

$$d = 2C_1C_2e^{-\frac{C_3c_2}{1-q_2}} + \frac{C_2^3c_2^2}{1-q_2^2} + \frac{2C_2c_1^2}{1-q_1^2} \tag{13}$$

Proof of the theorem can be found in appendix. In the above theorem, we derived a bound of the accuracy the algorithm guarantees. This bound has three terms. The third term $\frac{2c_1^2}{1-q_1^2} \to 0$ as $c_1 \to 0$. The first term decreases if $\frac{c_2}{1-q_2}$ increases, while the second term decreases if $\frac{c_2^2}{1-q_2^2}$ decreases.

Lemma 11. *For any $d' > 0$ there exists a selection of $c_1, c_2 > 0$ and $q_1 \in (0, 1), q_2 \in (0, q_1)$ such that $2C_1C_2e^{-\frac{C_3c_2}{1-q_2}} + \frac{C_2^3c_2^2}{1-q_2^2} + \frac{2C_2c_1^2}{1-q_1^2} \leq d'$.*

The proof of this lemma can be found in appendix. Although the first term and the second term seems competing each other, this lemma shows that their sum can be reduced to arbitrary small. That is, our algorithm guarantees arbitrary level of accuracy if needed.

4.5 Discussion

The algorithm has two noise parameters: the initial noise c_1 and the noise decay rate q_1; and it has two step-size parameters: the initial step size c_2 and the step size decay rate q_2. We have established that the algorithm guarantees ϵ-differential privacy and d-accuracy with $\epsilon = \frac{2C_2\sqrt{n}c_2q_1}{c_1(q_1-q_2)}$ and $d = 2C_1C_2e^{-\frac{C_3c_2}{1-q_2}} + \frac{C_2^3c_2^2}{1-q_2^2} + \frac{2C_2c_1^2}{1-q_1^2}$. By choosing these parameters appropriately, the algorithm can either guarantee ϵ-differential privacy for arbitrary $\epsilon > 0$ or it can guarantee d-accuracy for arbitrary $d > 0$. The dependency of the accuracy level and the privacy level of the algorithm on each of the four parameters based on the partial derivative of ϵ and d. The dependency sometimes involves the relative magnitudes of constants C_1, C_2 and C_3. In that case, we assume that $C_1 \gg C_2, C_3$, because in many applications we want to search for an optima over a relatively large domain. We observe that if we fix any other parameters:

(I) if the initial noise c_1 increases, then (i) the privacy level increases, but (ii) the accuracy level decreases,

(II) if the noise decaying decay q_1 increases, then (i) the privacy level increases, and (ii) accuracy level decreases,

(III) if the initial step size c_2 increases, (i) the privacy level decreases, and (ii) the accuracy level increase first and then decreases. The best accuracy can be achieved if $c_2 = c^*$ which solves the equation $C_1C_3e^{\frac{C_3c^*}{q_2-1}} = \frac{C_2c^*}{1+q_2}$,

(IV) if the step decaying rate q_2 increases, (i) the privacy level decreases, and (ii) the accuracy level increase first and then decreases. The best accuracy can be achieved if $q_2 = q^*$ which solves the equation $C_1C_3e^{\frac{C_3c_2}{q^*-1}} = \frac{C_2^2c_2}{(1+q^*)^2}$.

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 $\Sigma_{i=1}^N f_i$ 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, namely the f_i's, 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 a the gradient of its own cost function (f_i for agent i). 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.

Accurately solving distributed coordination problems require information sharing. Participants in such distributed 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 or inaccuracy 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] C. Dwork. Differential privacy. In *AUTOMATA, LANGUAGES AND PROGRAMMING*, volume 4052 of *Lecture Notes in Computer Science*, 2006.

[2] 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.

[3] 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.

[4] 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.

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

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

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

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

[9] 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 Proposition 2

Proof. Fixed a DOP $\mathcal{P}$, the communication graphs A_t are fixed. Fixed an observation sequence ρ, the messages $y(t)$ at each round t are fixed. From Equation (4), for each $i \in [N]$ and $t \in \mathbb{N}$, $z_i(t)$ is uniquely determined. Then by Equation (5), recalling that f_i is specified by DOP $\mathcal{P}$, we can conclude that $x_i(t)$ is uniquely specified for each $i \in [N]$ and $t \in \mathbb{N}$. Besides, $x(0)$ is specified by Θ. Thus, the execution $\alpha = x(0), \langle x(1), y(1), z(1)\rangle, \cdots = \mathcal{R}^{-1}(\mathcal{P}, \rho, \Theta)$ is uniquely determined. □

Proof of Lemma 6

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,

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

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

$$x_i(t) = \sum_{j\in[N]} \Phi(t,0)_{i,j}x_j(0) + \sum_{s=1}^{t}\sum_{j\in[N]} \Phi(t,s)_{i,j}u_j(s) + \sum_{s=0}^{t-1}\sum_{j\in[N]} \Phi(t,s-1)_{i,j}w_j(s)$$

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

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

By applying Lemma 5, the above expression can be reduced to

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

From Assumption 1, we have $||x_k(0)|| \leq C_1$ and $||\bigtriangledown f_k(z_k(s))|| \leq C_2$. From the property of projection, $||u_k(s)|| = ||Proj_{\mathcal{X}}[z_k(s) - \gamma_t \bigtriangledown f_k(z_k(s))] - z_k(s)|| \leq ||\gamma_t \bigtriangledown f_k(z_k(s))|| \leq \gamma_t C_2$. Thus we derive

$$||x_i(t) - x_j(t)|| \leq 2NC_1\theta\beta^t + 2NC_2\theta\sum_{s=1}^{t}\gamma_s\beta^{t-s} + 2N\theta\sum_{s=1}^{t}\beta^{t-s+1}||w_k(s)|| \quad (15)$$

□

Proof of Lemma 7:

Proof. $\{a_k\}_{k=1}^{\infty}$ is a convergent sequence, thus bounded. Let $|a_k| \leq M$ for all $k \in \mathbb{N}$. Fixed any $\epsilon > 0$, there exists an $N_1 \in \mathbb{N}$ such that for all $k \geq N_1$, $|a_k| \leq \frac{\epsilon(1-\beta)}{2}$. There exists an $N_2 \in \mathbb{N}$ such that $\beta^{N_2} \leq \frac{\epsilon(1-\beta)\beta^{N_1}}{2M(\beta-\beta^{N_1})}$. For all $n \geq \max\{N_1, N_2\}$, the absolute value of the summation in Equation (11) is bounded:

$$\begin{aligned}|\sum_{k=1}^{n}\beta^{n-k}a_k| &\leq \sum_{k=1}^{N_1-1}|\beta^{n-k}a_k| + \sum_{k=N_1}^{n}|\beta^{n-k}a_k| \quad (16)\\ &\leq M\sum_{k=1}^{N_1-1}\beta^{n-k} + \sum_{k=N_1}^{n}\beta^{n-k}|a_k| \quad (17)\end{aligned}$$

The first summation of Expression (17) is $\sum_{k=1}^{N_1-1}\beta^{n-k} = \frac{\beta^n(\beta-\beta^{N_1})}{\beta^{N_1}(1-\beta)}$. For $n \geq N_2$, we have $\beta^n \leq \frac{\epsilon(1-\beta)\beta^{N_1}}{2M(\beta-\beta^{N_1})}$. Thus,

$$M\sum_{k=1}^{N_1-1}\beta^{n-k} \leq M\frac{\epsilon(1-\beta)\beta^{N_1}}{2M(\beta-\beta^{N_1})}\frac{(\beta-\beta^{N_1})}{\beta^{N_1}(1-\beta)} \leq \frac{\epsilon}{2}.$$

In the second summation of Expression (17), we have $|a_k| \leq \frac{\epsilon(1-\beta)}{2}$ from the construction of N_1. Thus

$$\sum_{k=N_1}^{n}\beta^{n-k}|a_k| \leq \frac{\epsilon(1-\beta)}{2}\sum_{k=N_1}^{n}\beta^{n-k} \leq \frac{\epsilon(1-\beta)}{2}\sum_{i=0}^{\infty}\beta^i = \frac{\epsilon(1-\beta)}{2}\frac{1}{1-}$$

Substitute the above inequilities into Equation (17), we have $|\sum_{k=1}^{n}\beta^{n-k}a_k| \leq \epsilon$ for $n \geq \max\{N_1, N_2\}$. That is $\lim_{n\to\infty}\sum_{k=1}^{n}\beta^{n-k}$ 0. □

Proof of Lemma 9

Proof. From Equation (3)-(4), we have $z_i(t) = \sum_{j\in[N]} a_{i,j}(t)(x_j(t-1) + w_j(t))$. Thus,

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

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 (18), we have

$$\sum_{i\in[N]} ||z_i(t) - x^*||^2 = \sum_{i\in[N]} || \sum_{j\in[N]} a_{i,j}(t)\left(x_j(t-1) + w_j(t) - x'\right)||^2. \quad (19)$$

By triangular inequality and reordering of summation, we have

$$\begin{aligned}||\sum_{j\in[N]} a_{i,j}(t)\left(x_j(t-1) + w_j(t) - x'\right)||^2 &\leq \sum_{i\in[N]}\sum_{j\in[N]} a_{i,j}(t)| \\ &= \sum_{j\in[N]}\sum_{i\in[N]} a_{i,j}(t)| \quad (20)\end{aligned}$$

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

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

Combining above equation with Equations (19) and (20), we derive

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

Thus the lemma follows. □

Proof of Theorem 10.

Proof. From the property of stronly convex function, we have $\bigtriangledown f_i(x))(y-x) \leq f_i(y) - f_i(x) - \frac{C_3}{2}||y-x||^2$ for any

$x, y \in \mathcal{X}$. We denote $u_i(t) = -\bigtriangledown f_i(z_i(t))$. Let x^* be the minimum of the problem. Thus

$$u_i^T(t)(z_i(t)-x^*) \leq f_i(x^*)-f_i(z_i(t))-\frac{C_3}{2}||z_i(t)-x^*||^2 \leq -\frac{C_3}{2}||z_i(t)-x^*||^2. \quad (21)$$

Take 2-norm on both side of Equation (5), using the property of projection, we have

$$||x_i(t)-x^*||^2 \leq ||z_i(t)+\gamma_t u_i(t)-x^*||^2 = ||z_i(t)-x^*||^2+2\gamma_t u_i^T(t)(z_i(t)-x^*)+\gamma_t^2||u_i(t)||^2$$

Combining this equation with Equation (21) we have

$$||x_i(t)-x^*||^2 \leq (1-C_3\gamma_t)||z_i(t)-x^*||^2+C_2^2\gamma_t^2.$$

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

$$\frac{1}{N}\sum_{i\in[N]}||x_i(t)-x^*||^2 \leq \frac{1-C_3\gamma_t}{N}\sum_{i\in[N]}||z_i(t)-x^*||^2+C_2^2\gamma_t^2. \quad (22)$$

We will replace the terms $||z_i(t)-x^*||^2$ using Lemma 9. From Equation (12), we have:

$$\begin{aligned}\sum_{i\in[N]}||z_i(t)-x'||^2 &\leq \sum_{i\in[N]}||x_i(t-1)-x'+w_i(t)||^2 \\ &= \sum_{i\in[N]}||x_i(t-1)-x'||^2+\sum_{i\in[N]}[(x_i(t-1)-x')^T w_i(t)]+\sum_{i\in[N]}||w_i(t)||^2\end{aligned}$$

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:

$$\sum_{i\in[N]}\mathbb{E}||x_i(t)-x^*||^2 \leq \sum_{i\in[N]}\mathbb{E}||x_i(t-1)-x^*||^2+2NM_t^2. \quad (23)$$

For simplicity we denote $S(t) \triangleq \frac{1}{N}\sum_{i\in[N]}\mathbb{E}||x_i(t)-x^*||^2$. Combining Equation (22) and (23), we have:

$$S(t) \leq (1-C_3\gamma_t)S(t-1)+C_2^2\gamma_t^2+2(1-C_3\gamma_t)M_t^2 \quad (24)$$

Recursively apply Equation (24), we ultimately get:

$$S(t) \leq \prod_{s=1}^{t}(1-C_3\gamma_s)S(0)+C_2^2\sum_{s=1}^{t}\gamma_s^2\prod_{l=s+1}^{t}(1-C_3\gamma_l)+2\sum_{s=1}^{t}M_s^2\prod_{l=s}^{t}(1-C_3\gamma_l). \quad (25)$$

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

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

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

$$S(t) \leq 2C_1\Psi(t,0)+C_2^2\sum_{s=1}^{t}\gamma_s^2+2\sum_{s=1}^{t}M_s^2 \leq 2C_1\Psi(t,0)+C_2^2\sum_{s=1}^{\infty}\gamma_s^2+2\sum_{s=1}^{\infty}M_s^2 \leq 2C_1\Psi(t,0)+\frac{C_2^2c_2^2}{1-q_2^2}+\frac{2c_1^2}{1-q_1^2}. \quad (26)$$

We computes a tighter bound of term $\Psi(t,0)$. We use a standard property of exponential function, that is, $1-a \leq e^{-a}$ for any real number a. Thus

$$\Psi(t,0) = \prod_{s=1}^{t}(1-C_3\gamma_t) \leq e^{-\sum_{s=1}^{t}C_3\gamma_t} \leq e^{-\frac{C_3c_2(1-q_2^t)}{1-q_2}}.$$

Substitute the above inequality into Equation (26), we have:

$$S(t) \leq 2C_1e^{-\frac{C_3c_2(1-q_2^t)}{1-q_2}}+\frac{C_2^2c_2^2}{1-q_2^2}+\frac{2c_1^2}{1-q_1^2}$$

By triangular inequality, we have $\mathbb{E}||\bar{x}(t)-x^*||^2 = \mathbb{E}||\frac{1}{N}\sum_{i\in[N]}x_i(t)-x^*||^2 \leq \frac{1}{N}\sum_{i\in[N]}\mathbb{E}||x_i(t)-x^*||^2 = S(t)$. It follows that

$$\mathbb{E}||\bar{x}(t)-x^*||^2 \leq 2C_1e^{-\frac{C_3c_2(1-q_2^t)}{1-q_2}}+\frac{C_2^2c_2^2}{1-q_2^2}+\frac{2c_1^2}{1-q_1^2}. \quad (27)$$

By the property of a convex function f,

$$f(\bar{x}(t)) \leq OPT-\bigtriangledown f(\bar{x}(t))^T(x^*-\bar{x}(t)) \leq OPT+||\bigtriangledown f(\bar{x}(t))||||x^*-\bar{x}(t)$$

Taking expected value on both sides and combining with Equation (27), we have

$$\mathbb{E}[f(\bar{x}(t))] \leq OPT+2C_1C_2e^{-\frac{C_3c_2(1-q_2^t)}{1-q_2}}+\frac{C_2^2c_2^2}{1-q_2^2}+\frac{2C_2c_1^2}{1-q_1^2}.$$

Letting $t \to \infty$, we get the following which proves the theorem:

$$\lim_{t\to\infty}\mathbb{E}[f(\bar{x}(t))] \leq 2C_1C_2e^{-\frac{C_3c_2}{1-q_2}}+\frac{C_2^3c_2^2}{1-q_2^2}+\frac{2C_2c_1^2}{1-q_1^2}$$

□

Proof of Lemma 11.

Proof. The idea of the proof is to construct the following sequences: $c_1(n) = \frac{1}{n^2}$, $c_2(n) = \frac{1}{n^2}$, $q_1(n) = 1-\frac{0.5}{n^3}$ and $q_2(n) = 1-\frac{0.8}{n^3}$. It is clear that for each $n \in \mathbb{N}$, the following conditions hold: $c_1(n), c_2(n) > 0$, $q_1(n) \in (0,1)$ and $q_2(n) \in (0, q_1(n))$. We define

$$d(n) \triangleq 2C_1C_2e^{-\frac{C_3c_2(n)}{1-q_2(n)}}+\frac{C_2^2c_2(n)^2}{1-q_2(n)^2}+\frac{2c_1(n)^2}{1-q_1(n)^2} = 2C_1C_2e^{-\frac{C_3n^3}{n^2}}+\frac{}{0.}$$

It can be shown that $d(n) \to 0$ as $n \to \infty$. Thus, for any d', there exists a $M \in naturals$ such that $d(M) \leq d'$. Then, by choosing $c_1 = c_1(M), c_2 = c_2(M), q_1 = q_1(M), q_2 = q_2(M)$, we guarantees that the corresponding algorithm guarantees $\mathbb{E}[f(\bar{x}(t))] - OPT \leq d'$.

□