Distributed Non-Convex First-Order Optimization and Information Processing: Lower Complexity Bounds and Rate Optimal Algorithms

Haoran Sun and Mingyi Hong

***Abstract*—We consider a class of popular distributed non-convex optimization problems, in which agents connected by a network $\mathcal{G}$ collectively optimize a sum of smooth (possibly non-convex) local objective functions. We address the following question: if the agents can only access the gradients of local functions, what are the *fastest* rates that any distributed algorithms can achieve, and how to achieve those rates.**

First, we show that there exist difficult problem instances, such that it takes a class of distributed first-order methods at least $\mathcal{O}(1/\sqrt{\xi(\mathcal{G})} \times \bar{L}/\epsilon)$ communication rounds to achieve certain ϵ-solution [where $\xi(\mathcal{G})$ denotes the spectral gap of the graph Laplacian matrix, and $\bar{L}$ is some Lipschitz constant]. Second, we propose (near) optimal methods whose rates match the developed lower rate bound (up to a ploylog factor). The key in the algorithm design is to properly embed the classical polynomial filtering techniques into modern first-order algorithms. To the best of our knowledge, this is the first time that lower rate bounds and optimal methods have been developed for distributed non-convex optimization problems.

I. Introduction

A. Problem and motivation

We consider the following distributed optimization problem

$$\min_{y\in\mathbb{R}^S} \ \bar{f}(y) := \frac{1}{M}\sum_{i=1}^{M} f_i(y), \tag{1}$$

where $f_i(y) : \mathbb{R}^S \to \mathbb{R}$ is a smooth and possibly non-convex function accessible to agent i. There is no central controller, and the M agents are connected by a network defined by an *undirected* and *unweighted* graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, with $|\mathcal{V}| = M$ vertices and $|\mathcal{E}| = E$ edges. Each agent i can only communicate with its immediate neighbors, and it can access its local component function f_i.

A common way to reformulate problem (1) in the distributed setting is given below. Introduce M local variables $x_1, \cdots, x_M \in \mathbb{R}^S$ and a concatenation of M variables $x := [x_1; \cdots; x_M] \in \mathbb{R}^{SM\times 1}$, then the following formulation is equivalent to (1) whenever $\mathcal{G}$ is connected

$$\min_{x\in\mathbb{R}^{SM}} \ f(x) := \frac{1}{M}\sum_{i=1}^{M} f_i(x_i), \text{ s.t. } x_i = x_j, \forall\ (i,j)\in\mathcal{E}. \tag{2}$$

where $f(x) : \mathbb{R}^{SM} \to \mathbb{R}$. After the reformulation, the objective function now becomes separable, and the linear constraint encodes the network connectivity pattern.

H. Sun and M. Hong are with the Department of Electrical and Computer Engineering (ECE), University of Minnesota, Minneapolis, MN 55414, USA. Email: {sun00111,mhong}@umn.edu. They are supported by NSF grants CMMI- 1727757, CCF-1526078, and by an AFOSR grant 15RT0767.

The conference version of this paper has been accepted by Asilomar Conference on Signal, Systems and Computer, 2019 [1].

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes additional proofs of results in Sec. III.

B. Distributed non-convex optimization

Distributed non-convex optimization has gained considerable attention recently. For example, it finds applications in training neural networks [2], clustering [3], and dictionary learning [4], just to name a few.

The problem (1) and (2) have been studied extensively in the literature when f_i's are all convex; see for example [5]–[7]. Primal based methods such as distributed subgradient (DSG) method [5], the EXTRA method [7], as well as primal-dual based methods such as distributed augmented Lagrangian method [8], Alternating Direction Method of Multipliers (ADMM) [9], [10] have been proposed.

On the contrary, only recently there have been works addressing the more challenging problems without assuming convexity of f_i; see [2], [4], [11]–[24]. The convergence behavior of the distributed consensus problem (1) has been studied in [4], [11], [12]. Reference [13] develops a non-convex ADMM based methods for solving the distributed consensus problem (1). However the network considered therein is a star network in which the local nodes are all connected to a central controller. References [15], [16] propose a primal-dual based method for unconstrained problem over a connected network, and derives a global convergence rate for this setting. In [14], [18], [19], the authors utilize certain gradient tracking idea to solve a constrained nonsmooth distributed problem over possibly time-varying networks. The work [20] summarizes a number of recent progress in extending the DSG-based methods for non-convex problems. References [2], [17], [21] develop methods for distributed stochastic zeroth and/or first-order non-convex optimization. It is worth noting that the distributed algorithms proposed in all these works converge to first-order stationary solutions, which contain local maximum, local minimum and saddle points.

Recently, the authors of [23], [25]–[27] have developed first-order distributed algorithms that are capable of computing second-order stationary solutions (which under suitable conditions become local optimal solutions). Other second-order distributed algorithms such as [28], [29] are design for convex problems, and they utilize high-order Hessian information about local problems.

C. Lower and upper rate bounds analysis

Despite the strong interests and many recent contributions in this field, one major question remains open:

(Q) What is the *best* convergence rate achievable by *any* distributed algorithms for the non-convex problem (1)?

Question (**Q**) seeks to find a *best convergence rate*, which is a characterization of the *smallest number* of iterations required

to achieve certain *high-quality solutions*, among all distributed algorithms. Clearly, understanding $(\mathbf{Q})$ provides fundamental insights to distributed optimization and information processing. The answer to $(\mathbf{Q})$ offers meaningful estimates on the total amount of communication and computation efforts that are required to achieve a given level of accuracy. Further, the identified optimal strategies capable of attaining the best convergence rates will also help guide the practical design of distributed algorithms, convex and non-convex alike.

Convergence rate analysis (aka iteration complexity analysis) for convex problems dates back to early works by Nesterov, Nemirovsky and Yudin [30], [31], in which lower bounds and optimal *first-order* algorithms have been developed; also see [32]. In recent years, many accelerated *first-order* algorithms achieving those lower bounds for different kinds of convex problems have been derived, both for centralized [33], [34] and distributed settings [35]. In those works, the problem is to optimize $\min_x f(x)$ with convex f, and the optimality measure used is $f(x) - f(x^*)$. The lower bound can be expressed as [32, Theorem 2.2.2]

$$f(x^t) - f(x^*) \leq \frac{\|x^0 - x^*\| L}{(t+2)^2}, \tag{3}$$

where L is the Lipschitz constant for ∇f; x^* (resp. x^0) is the global optimal solution (resp. the initial solution); t is the iteration index. Therefore to achieve ϵ-optimal solution in which $f(x^t) - f(x^*) \leq \epsilon$, one needs $\sqrt{\frac{\|x^* - x^0\| L}{\epsilon}}$ iterations. Recently the above approach has been extended to distributed strongly convex optimization in [36], where problem (1) is considered, with each f_i being strongly convex. The authors provide lower and upper rate bounds for a class of algorithms in which the local agents can utilize both $\nabla f_i(x)$ and its Fenchel conjugate $\nabla^* f_i(x)$. We note that this result is not directly related to the class of "first-order" method, since computing the Fenchel conjugate $\nabla^* f_i(x)$ requires performing certain exact minimization, which involves solving a strongly convex optimization problem. Other related works in this direction also include [37] and [38], both are for convex cases. In particular, the work [38] is a non-smooth extension of [36], where the lower complexity bound under the Lipschitz continuity of the global and local objective function are discussed and the optimal algorithm is proposed. The work [37] studies the optimal convergence rates for distributed convex optimization problems, including both strongly convex and convex, smooth and non-smooth cases.

When the problem becomes non-convex, the size of the gradient function can be used as a measure of solution quality. In particular, let $h_T^* := \min_{0 \leq t \leq T} \|\nabla f(x^t)\|^2$, then it has been shown that the classical (centralized) gradient descent (GD) method achieves the following rate [32, page 28]

$$h_T^* \leq \frac{c_0 L (f(x^0) - f(x^*))}{T+1}, \text{ where } c_0 > 0 \text{ is some constant.}$$

It has been shown in [39], [40] that the above rate is optimal for *any* first-order methods that only utilize the gradient information, when applied to problems with Lipschitz gradient. However, no lower bound analysis has been developed for distributed non-convex problem (1); there are even not many algorithms that provide achievable upper rate bounds (except for the recent works [13], [16], [41]), not to mention any analysis on the tightness/sharpness of these upper bounds.

D. Contribution of this work

In this work, provide answers to (some specific versions of) question $(\mathbf{Q})$. Our main contributions are given below:
1) We develop the first lower complexity bound for a class of *distributed first-order methods* to solve problem (1). We show that, to achieve certain ϵ-optimality, it is necessary for any such algorithm to perform $\mathcal{O}(1/\sqrt{\xi(\mathcal{G})} \times \bar{L}/\epsilon)$ rounds of communication among all the nodes, where $\xi(\mathcal{G})$ is certain spectral gap of the graph Laplacian matrix, and $\bar{L}$ is the averaged Lipschitz constant of the gradients of local functions. On the other hand, it is necessary for any such algorithm to perform $\mathcal{O}(\bar{L}/\epsilon)$ rounds of computation among all the nodes.
2) We design an optimal algorithm that is based on a novel *approximate filtering -then- predict and tracking* (xFILTER) strategy, which achieves our derived lower complexity bounds (up to a ploylog factor).

In Table I, we specialize some key results developed in the paper to a few popular graphs, and compare them with the achievable rates of centralized GD.

Notations. For a symmetric matrix B, $\lambda_{\max}(B)$, $\lambda_{\min}(B)$ and $\underline{\lambda}_{\min}(B)$ denote the maximum, the minimum and the minimum *nonzero* eigenvalues; I_P denotes an identity matrix with size P, $\mathbb{1}_M$ denotes an all one vector of size M, and $\otimes$ denotes the Kronecker product. $[M]$ denotes the set $\{1, \cdots, M\}$. For a vector x, $x[i]$ denotes its ith element; We use $\widetilde{\mathcal{O}}$ to denote $\mathcal{O}(\log(M))$ where M is the problem dimension; use $i \sim j$ to denote two connected nodes i and j, i.e., for a graph $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$, $i \sim j$ if $i \neq j$, and $(i,j) \in \mathcal{E}$; use $\text{col}(X)$ to denote the column space of a matrix X.

II. Preliminaries

To properly address $(\mathbf{Q})$, we need to specify the *concrete classes* of problems, networks, algorithms, and the solution measures under consideration.

A. The class $\mathcal{P}$, $\mathcal{N}$, $\mathcal{A}$

Problem Class. A problem is in class $\mathcal{P}_L^M$ if:
A1. The objective is an average of M functions; see (1).
A2. Each component function $f_i(x)$'s has Lipschitz gradient:

$$\|\nabla f_i(x_i) - \nabla f_i(z_i)\| \leq L_i \|x_i - z_i\|, \ \forall \ x_i, z_i \in \mathbb{R}^S, \ \forall \ i, \tag{4}$$

where $L_i \geq 0$ is the *smallest* positive number such that the above inequality holds true.
Define $\bar{L} := \frac{1}{M} \sum_{i=1}^M L_i$, $L_{\max} := \max_i L_i$, and $L_{\min}$ similarly. Define the matrix of Lipschitz constants as:

$$L := \text{diag}([L_1, \cdots, L_M]) \otimes I_S \in \mathbb{R}^{MS \times MS}. \tag{5}$$

A3. The function $f(x)$ is lower bounded over $x \in \mathbb{R}^{MS}$, i.e.,

$$\underline{f} := \inf_x f(x) > -\infty. \tag{6}$$

These assumptions are rather mild. For example an f_i satisfies [A2-A3] is not required to be second-order differentiable.
Network Class. Let $\mathcal{N}$ denote a class of networks represented by an *undirected* and *unweighted* graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, with

Network Instances	Problem Classes		Rate Achieving Algorithm
	Uniform Lipschitz U	Non-uniform Lipschitz $\{L_i\}$	
Complete/Star	$\mathcal{O}(U/\epsilon)$	$\mathcal{O}(1/\epsilon \times \sum_i L_i/M)$	xFILTER/Prox-GPDA
Random Geometric	$\widetilde{\mathcal{O}}(U\sqrt{M}/(\sqrt{\log M}\epsilon))$	$\widetilde{\mathcal{O}}(\sqrt{M}/(\sqrt{\log(M)}\epsilon) \times \sum_i L_i/M)$	xFILTER
Path/Circle	$\widetilde{\mathcal{O}}(UM/\epsilon)$	$\widetilde{\mathcal{O}}(M/\epsilon \times \sum_i L_i/M)$	xFILTER
Grid	$\widetilde{\mathcal{O}}(U\sqrt{M}/\epsilon)$	$\widetilde{\mathcal{O}}(\sqrt{M}/\epsilon \times \sum_i L_i/M)$	xFILTER
Centralized	$\mathcal{O}(U/\epsilon)$	$\mathcal{O}(1/\epsilon \times \sum_i L_i/M)$	Gradient Descent (GD)

TABLE I: The main results of the paper when specializing to a few popular graphs. The entries show the best rate bounds achieved by the proposed xFILTER algorithm for a number of specific graphs and problem class; L_i is the Lipschitz constant for ∇f_i [see (4)]; for the uniform case $U = L_1, \cdots, L_M$. For the uniform Lipschitz the lower rate bounds derived for the particular graph matches the upper rate bounds (we only show the latter in the table). The last row shows the rate achieved by the centralized GD algorithm. The notation $\tilde{\mathcal{O}}$ denotes big $\mathcal{O}$ with some polynomial in logarithms, i.e, use $\tilde{\mathcal{O}}$ to denote $\mathcal{O}(\log(M))$ where M is the problem dimension.

$|\mathcal{V}| = M$ vertices and $|\mathcal{E}| = E$ edges. In this paper the term 'network' and 'graph' will be used interchangeably. Also, we use $\mathcal{N}_D^M$ to denote a class of network similarly as above, but with M nodes and a diameter of D, defined below [where dist$(\cdot)$ indicates the distance between two nodes]: $D := \max_{u,v\in\mathcal{V}} \text{dist}(u, v)$.

Following the convention in [42], we define a number of graph related quantities below. First, define the *degree* of node i as d_i, and define the averaged degree as:

$$\bar{d} := \frac{1}{M}\sum_{i=1}^{M} d_i \tag{7}$$

Define the incidence matrix (IM) $A \in \mathbb{R}^{E\times M}$ as follows: if $e \in \mathcal{E}$ and it connects vertex i and j with $i > j$, then $A_{ev} = 1/\sqrt{d_v}$ if $v = i$, $A_{ev} = -1/\sqrt{d_v}$ if $v = j$ and $A_{ev} = 0$ otherwise [42, Theorem 8.3]. The *graph Laplacian matrix* and the *degree matrix* are defined as follows (see [42, Section 1.2]):

$$\mathcal{L} := A^\top A \in \mathbb{R}^{M\times M},\ P := \text{diag}[d_1, \cdots, d_M] \in \mathbb{R}^{M\times M}. \tag{8}$$

In particular, the elements of the Laplacian are given as:

$$[\mathcal{L}]_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } i \sim j, i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

We note that the graph Laplacian defined here is sometimes known as the *normalized* graph Laplacian in the literature, but throughout this paper we follow the convention used in the classical work [42] and simply refer it as the *graph Laplacian*. For convenience, we also define a scaled version of the IM:

$$F := AP^{1/2} \in \mathbb{R}^{E\times M}. \tag{9}$$

It is known that the scaled IM satisfies the following:

$$F\mathbb{1}_M = AP^{1/2}\mathbb{1}_M = 0. \tag{10}$$

Define the second smallest eigenvalue of $\mathcal{L}$, as $\underline{\lambda}_{\min}(\mathcal{L})$:

$$\underline{\lambda}_{\min}(\mathcal{L}) = \inf_{x:\sum_{i=1}^M x_i d_i = 0} \frac{x^\top \mathcal{L} x}{\sum_{i=1}^M x_i^2 d_i}. \tag{11}$$

Then the *spectral gap* of the graph $\mathcal{G}$ can be defined below:

$$\xi(\mathcal{G}) = \frac{\underline{\lambda}_{\min}(\mathcal{L})}{\lambda_{\max}(\mathcal{L})} \leq 1. \tag{12}$$

Algorithm Class. Define the *neighbor set* for node $i \in \mathcal{E}$ as

$$\mathcal{N}_i := \{i \mid i \sim j, j \neq i\}. \tag{13}$$

We say that a *distributed, first-order* algorithm is in class $\mathcal{A}$ if it satisfies the following conditions.

[B1.] At iteration 0, each node can obtain some network related constants, such as M, D, eigenvalues of the graph Laplacian $\mathcal{L}$, etc.

[B2.] At iteration $t+1$, *each node* $i \in [M]$ first conducts a communication step by broadcasting the local x_i^t to all its neighbors, through a function $Q_i^t(\cdot) : \mathbb{R}^S \to \mathbb{R}^S$. Then each node will generate the new iterate, by combining the received message with its past gradients using a function $W_i^t(\cdot)$:

$$v_i^t = \underbrace{Q_i^t(x_i^t)}_{\text{communication step}},\ x_i^{t+1} \in \underbrace{W_i^t\left(\{\{v_j^k\}_{j\in\mathcal{N}_i}, \nabla f_i(x_i^k), x_i^k\}_{k=1}^t\right)}_{\text{computation step}}. \tag{14}$$

In this work, we will focus on the case where the $Q_i^t(\cdot)$'s and $W_i^t(\cdot)$'s are linear operators.

Clearly $\mathcal{A}$ belongs to the class of *first-order* methods because only local gradient information is used. It is also a class of *distributed* algorithms because at each iteration the nodes only communicate with their immediate neighbors.

Additionally, in practical distributed algorithms such as DSG, ADMM or EXTRA, nodes are dictated to use a *fixed strategy* to linearly combine all its neighbors' information. To model such a requirement, below we consider a slightly restricted algorithm class $\mathcal{A}'$, where we require each node to use the same coefficients to combine its neighbors (note that allowing the nodes to use a fixed but *arbitrary* linear combination is also possible, but the resulting analysis will be more involved). In particular, we say that a *distributed, first-order* algorithm is in $\mathcal{A}'$ if it satisfies [B1] and the following:

[B2'.] At iteration $t+1$, *each node* $i \in [M]$ performs:

$$v_i^t = Q_i^t(x_i^t),\ x_i^{t+1} \in W_i^t\left(\{\sum_{j\in\mathcal{N}_i} v_j^t, \nabla f_i(x_i^k), x_i^k\}_{k=1}^t\right). \tag{15}$$

We remark that, in both algorithm classes, *one round* of communication occurs at each iteration, where *each* node broadcasts its local variable x_i^t once. Therefore, the total iteration number is the same as the total communication rounds. However, the total times that the entire gradient $\{\nabla f_i(x_i)\}_{i=1}^M$ is evaluated could be *smaller* than the total iteration number/communication rounds. This is because when we compute x_i^{t+1}, the operation $W_i^t(\cdot)$ can set the coefficient in front of $\nabla f_i(x_i^r)$ to be zero, effectively *skipping* the local gradient computation.

B. Solution Quality Measure

Next we provide definitions for the quality of the solution. Note that since we consider using first-order methods to solve

non-convex problems, it is expected that in the end some first-order stationary solution with small $\|\nabla f\|$ will be computed.

The measure we provided below is directly related to *local variables* $\{x_i \in \mathbb{R}^S\}_{i=1}^M$. At a given iteration T, we say that $\{x_i^T\}$ is a *local* ϵ-solution if the following holds:

$$h_T^* := \min_{t\in[T]} \left\| \sum_{i=1}^M \frac{\nabla f_i(x_i^t)}{M} \right\|^2 + \frac{1}{M\underline{\lambda}_{\min}(P^{1/2}\mathcal{L}P^{1/2})} \sum_{(i,j):i\sim j} \sqrt{L_iL_j}\|x_i^t - x_j^t\|^2 \leq \epsilon. \tag{16}$$

Clearly this definition takes into consideration both the consensus error and the size of the local gradients. It is easy to check that when h_T^* goes to zero, a first-order stationary solution for problem (1) is obtained. Note that the constant $\frac{1}{M\underline{\lambda}_{\min}(P^{1/2}\mathcal{L}P^{1/2})}$ is needed to balance the two different terms. The "$\min_{t\in[T]}$" operation is needed to track the best solution obtained until iteration T, because the quantity inside this operation may not be monotonically decreasing.

In our work we will focus on providing answers to the following specific version of question $(\mathbf{Q})$:

> For *any* $\epsilon > 0$, what is the minimum iteration T needed for any algorithm in class $\mathcal{A}$ (or class $\mathcal{A}$') to solve instances in classes $(\mathcal{P}, \mathcal{N})$, so to achieve $h_T^* \leq \epsilon$?

C. Some Useful Facts and Definitions

Below we provide a few facts about the above classes.

On Lipschitz constants. Assume that each f_i has Lipschitz continuous gradient with constant L_i in (4). Then we have:

$$\|\nabla \bar{f}(y_1) - \nabla \bar{f}(y_2)\| \leq \bar{L}\|y_1 - y_2\|, \ \forall\ y_1,\ y_2. \tag{17}$$

We also have the following

$$\|\nabla f(x) - \nabla f(z)\|^2 = \frac{1}{M^2}\sum_{i=1}^M \|\nabla f_i(x_i) - \nabla f_i(z_i)\|^2, \ \forall\ x_i,\ z_i$$

which implies

$$\|\nabla f(x) - \nabla f(z)\| \leq \frac{1}{M}\|L(x-z)\|, \quad \forall\ x, z \tag{18}$$

where the matrix L is defined in (5).

On Quantities for Graph $\mathcal{G}$. Let us present some usfeul properties for graph $\mathcal{G}$. Define the following matrices:

$$\Sigma := \text{diag}[\sigma_1, \cdots, \sigma_E] \succ 0,\ \Upsilon := \text{diag}([\beta_1, \cdots, \beta_M]) \succ 0.$$

For two diagonal matrices Υ^2 and Σ^2 of appropriate sizes, the *generalized Laplacian* (GL) matrix is defined as:

$$\mathcal{L}_G = \Upsilon^{-1}F^\top\Sigma^2 F\Upsilon^{-1}, \tag{19}$$

and its elements are given by:

$$[\mathcal{L}_G]_{ij} = \begin{cases} \frac{\sum_{q:i\sim q}\sigma_{iq}^2}{\beta_i^2} & \text{if } i = j \\ -\frac{\sigma_{ij}^2}{\beta_i\times\beta_j} & \text{if } (ij)\in\mathcal{E}, i\neq j \\ 0 & \text{otherwise} \end{cases}.$$

Define a diagonal matrix $K \in \mathbb{R}^{E\times E}$ as below:

$$[K]_{e,q} = \begin{cases} \sqrt{L_iL_j} & \text{if } e = q, \text{ and } e = (i,j) \\ 0 & \text{otherwise} \end{cases}. \tag{20}$$

Then when $\Upsilon = P^{1/2}L^{1/2}$ and $\Sigma^2 = K$, GL becomes:

$$\widetilde{\mathcal{L}} := L^{-1/2}P^{-1/2}F^\top KFP^{-1/2}L^{-1/2}. \tag{21}$$

Note that if any diagonal element in the matrix L is zero, then L^{-1} denotes the Moore - Penrose pseudoinverse. Similarly, if $\Upsilon = L^{1/2}$ and $\Sigma^2 = K$, then the GL matrix becomes:

$$\widehat{\mathcal{L}} := L^{-1/2}F^\top KFL^{-1/2}. \tag{22}$$

These matrices will be used later in our derivations.

Below we list some useful results about the Laplacian [42]–[44]. First, all eigenvalues of $\mathcal{L}$ lie in the interval $[0,\ 2]$. Also because $\underline{\lambda}_{\min}(\mathcal{L}) = \underline{\lambda}_{\min}(P^{-1/2}F^\top FP^{-1/2})$, we have

$$\underline{\lambda}_{\min}(\mathcal{L}) \leq \underline{\lambda}_{\min}(F^\top F). \tag{23}$$

Also we have that [42, Lemma 1.9]

$$\underline{\lambda}_{\min}(\mathcal{L}) \geq \frac{1}{D\sum_i d_i}. \tag{24}$$

The spectral of $\mathcal{L}$ for some special graphs are given below:
1) Complete Graph: The eigenvalues are 0 and $M/(M-1)$ (with multiplicity $M-1$), so $\xi(\mathcal{G}) = 1$;
2) Star Graph: The eigenvalues are 0 and 1 (with multiplicity $M-2$), and 2, so $\xi(\mathcal{G}) = 1/2$;
3) Path Graph: The eigenvalues are $1-\cos(\pi m/(M-1))$ for $m = 0, 1, \cdots, M-1$, and $\xi(\mathcal{G}) \geq 1/M^2$.
4) Cycle Graph: The eigenvalues are $1-\cos(2\pi m/M)$ for $m = 0, 1, \cdots, M-1$, and $\xi(\mathcal{G}) \geq 1/M^2$.
5) Grid Graph: The grid graph is obtained by placing the nodes on a $\sqrt{M}\times\sqrt{M}$ grid, and connecting nodes to their nearest neighbors. We have $\xi(\mathcal{G}) \geq 1/M$.
6) Random Geometric Graph: Place the nodes uniformly in $[0,1]^2$ and connect any two nodes separated by a distance less than $Ra \in (0,1)$. Then if Ra satisfies [44]

$$Ra = \Omega\left(\sqrt{\log^{1+\epsilon}(M)/M}\right), \quad \text{for any } \epsilon > 0, \tag{25}$$

then with high probability $\xi(\mathcal{G}) = \mathcal{O}\left(\frac{\log(M)}{M}\right)$.

III. Complexity Lower Bounds

We begin to develop the complexity lower bounds for algorithms in $\mathcal{A}$ to solve problems $\mathcal{P}_L^M$ over network $\mathcal{N}$. We will mainly focus on the case where f_i's have uniform Lipschitz constants $L_i = U \in (0,1), \quad \forall\ i \in [M]$. At the end of this section, generalization to the non-uniform case will be discussed. Our proof combines ideas from the classical work of Nesterov [45], as well as two recent constructions [40] (for centralized non-convex problems) and [36] (for strongly convex distributed problems). Differently from [45] [36], in our construction we can only use first-order differentiable, gradient Lipschitz continuous, but not second-order differentiable functions. Comparing with [40], we need to carefully construct network structures so that it is challenging for algorithm in $\mathcal{A}$ to achieve local-ϵ solutions.

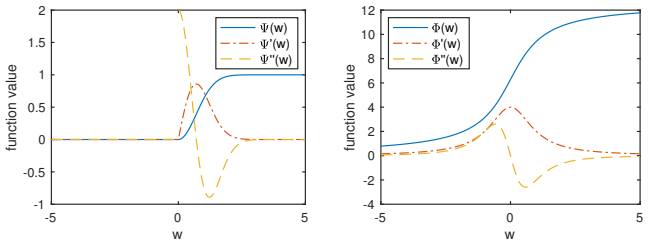


Fig. 1: The functional value, and derivatives of Ψ and Ψ.

To begin with, we construct the following two functions:

$$h(x) := \frac{1}{M}\sum_{i=1}^{M} h_i(x_i), \quad f(x) := \frac{1}{M}\sum_{i=1}^{M} f_i(x_i), \tag{26}$$

as well as the corresponding versions that evaluate on a "centralized" variable y

$$\bar{h}(y) := \frac{1}{M}\sum_{i=1}^{M} h_i(y), \quad \bar{f}(y) := \frac{1}{M}\sum_{i=1}^{M} f_i(y). \tag{27}$$

Here we have $x_i \in \mathbb{R}^T$, for all i, $y \in \mathbb{R}^T$, and $x := (x_1, \cdots x_M) \in \mathbb{R}^{TM\times 1}$. In our subsequent constructions, we will make h and $\bar{h}$ easy to analyze, while make f and $\bar{f}$ fall in the desired class $\mathcal{P}_U^M$.

A. Path Graph ($D = M - 1$)

First we consider the extreme case in which the nodes form a path graph with M nodes and each node i has its own local function h_i. For notational simplicity assume that M is a multiple of 3, that is, $M = 3C$ for some integer $C > 0$. Also assume that T is an odd number without loss of generality.

Define the component functions h_i's in (26) as follows.

$$h_i(x_i) = \begin{cases} \Theta(x_i,1) + 3\sum_{j=1}^{\lfloor T/2\rfloor} \Theta(x_i,2j), & i \in \left[1, \frac{M}{3}\right] \\ \Theta(x_i,1), & i \in \left[\frac{M}{3}+1, \frac{2M}{3}\right] \\ \Theta(x_i,1) + 3\sum_{j=1}^{\lfloor T/2\rfloor} \Theta(x_i,2j+1), & i \in \left[\frac{2M}{3}+1, M\right] \end{cases} \tag{28}$$

where we have defined the following functions

$$\Theta(x_i,j) := \Psi(-x_i[j-1])\Phi(-x_i[j]) - \Psi(x_i[j-1])\Phi(x_i[j]),\ \forall\ j \geq 2$$
$$\Theta(x_i,1) := -\Psi(1)\Phi(x_i[1]). \tag{29}$$

The component functions $\Psi, \Phi : \mathbb{R} \to \mathbb{R}$ are given as below

$$\Psi(w) := \begin{cases} 0 & w \leq 0 \\ 1 - e^{-w^2} & w > 0, \end{cases} \quad \text{and} \quad \Phi(w) := 4\arctan w + 2\pi.$$

Suppose $x_1 = x_2 = \cdots = x_M = y$, then the average function becomes:

$$\begin{aligned} \bar{h}(y) &:= \frac{1}{M}\sum_{j=1}^{M} h_i(y) = \Theta(y,1) + \sum_{i=2}^{T}\Theta(y,i) \\ &= -\Psi(1)\Phi\left(y[1]\right) \\ &+ \sum_{i=2}^{T}\left[\Psi\left(-y[i-1]\right)\Phi\left(-y[i]\right) - \Psi\left(y[i-1]\right)\Phi\left(y[i]\right)\right]. \end{aligned}$$

Further for a given error constant $\epsilon > 0$ and a given averaged Lipschitz constant $U \in (0,1)$, let us define

$$f_i(x_i) := \frac{150\pi\epsilon}{U} h_i\left(\frac{x_i U}{75\pi\sqrt{2\epsilon}}\right). \tag{30}$$

Therefore we also have, if $x_1 = x_2 = \cdots = x_M = y$, then

$$\bar{f}(y) := \frac{1}{M}\sum_{i=1}^{M} f_i(y) = \frac{150\pi\epsilon}{U}\bar{h}\left(\frac{yU}{75\pi\sqrt{2\epsilon}}\right). \tag{31}$$

First we present properties of the component functions h_i.

Lemma 3.1: The functions Ψ and Φ satisfy the following.

1) For all $w \leq 0$, $\Psi(w) = 0$, $\Psi'(w) = 0$.
2) The following bounds hold for the functions and their first and second-order derivatives:

$$0 \leq \Psi(w) < 1, \quad 0 \leq \Psi'(w) \leq \sqrt{\frac{2}{e}},$$
$$-\frac{4}{e^{\frac{3}{2}}} \leq \Psi''(w) \leq 2, \quad \forall w > 0$$

$$0 < \Phi(w) < 4\pi, \quad 0 < \Phi'(w) \leq 4,$$
$$-\frac{3\sqrt{3}}{2} \leq \Phi''(w) \leq \frac{3\sqrt{3}}{2}, \quad \forall w \in \mathbb{R}$$

3) The following key property holds:

$$\Psi(w)\Phi'(v) > 1, \quad \forall\ w \geq 1,\ |v| < 1. \tag{32}$$

4) The function h is lower bounded as follows:

$$h_i(0) - \inf_{x_i} h_i(x_i) \leq 10\pi T,\ h(0) - \inf_x h(x) \leq 10\pi T.$$

5) The first-order derivative of $\bar{h}$ (resp. h_j) is Lipschitz continuous with constant $\ell = 75\pi$ (resp. $\ell_j = 75\pi$, $\forall\ i$).

The next lemma is a simple extension of the previous result.

Lemma 3.2: We have the following properties for the functions f and $\bar{f}$ defined in (31) and (30).

1) We have $\forall\ x \in \mathbb{R}^{TM\times 1}$

$$f(0) - \inf_x f(x) + \frac{1}{MU}\|d_0\|^2 \leq \frac{1650\pi^2\epsilon}{U}T.$$

where we have defined

$$d_0 := [\nabla f_1(0), \cdots, \nabla f_M(0)]. \tag{33}$$

2) We have

$$\left\|\nabla\bar{f}(y)\right\| = \sqrt{2\epsilon}\left\|\nabla\bar{h}\left(\frac{yU}{75\pi\sqrt{2\epsilon}}\right)\right\|,\ \forall\ y \in \mathbb{R}^{T\times 1}. \tag{34}$$

3) The first-order derivatives of $\bar{f}$ and that for each $f_j, j \in [M]$ are Lipschitz continuous, with the same constant $U > 0$.

The next result analyzes the size of $\nabla\bar{h}$.

Lemma 3.3: If there exists $k \in [T]$ such that $|y[k]| < 1$, then the following holds

$$\left\|\nabla\bar{h}(y)\right\| = \left\|\frac{1}{M}\sum_{i=1}^{M}\nabla h_i(y)\right\| \geq \left|\frac{1}{M}\sum_{i=1}^{M}\frac{\partial}{\partial y[k]}h_i(y)\right| > 1.$$

Lemma 3.4: Define $\bar{x} := \frac{1}{M}\sum_{i=1}^{M} x_i$, and assume that $U \in$

$(0,1)$. *Then we have*

$$\left\|\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_i)\right\|^2+\frac{U}{M\lambda_{\min}(P^{1/2}\mathcal{L}P^{1/2})}\sum_{(i,j):i\sim j}\|x_i-x_j\|^2 \geq\frac{1}{2}\left\|\nabla\bar{f}(\bar{x})\right\|^2.$$

Lemma 3.5: Consider using an algorithm in class $\mathcal{A}$ or in class $\mathcal{A}'$ to solve the following problem:

$$\min_{x\in\mathbb{R}^{TM\times 1}}\ h(x)=\frac{1}{M}\sum_{i=1}^{M}h_i(x_i), \tag{35}$$

over a path graph. Assume the initial solution: $x_i=0,\ \forall\ i\in[M]$. Let $\bar{x}=\frac{1}{M}\sum_{i=1}^{M}x_i$ denote the average of the local variables. Then the algorithm needs at least $(\frac{M}{3}+1)T$ iterations to have $x_i[T]\neq 0,\ \forall\ i$ and $\bar{x}[T]\neq 0$.

Now we are ready to show our first main result.

Theorem 3.1: Let $U\in(0,1)$ and ϵ be positive. Then for any distributed first-order algorithm in class $\mathcal{A}$ or $\mathcal{A}'$, there exists a problem in class $\mathcal{P}_U^M$ and a network in class $\mathcal{N}$, such that it requires at least the following number of iterations and communication rounds

$$t\geq\frac{1}{3\sqrt{\xi(\mathcal{G})}}\left\lfloor\frac{\left(f(0)-\inf_x f(x)+\frac{\|d_0\|^2}{MU}\right)U}{1650\pi^2}\epsilon^{-1}\right\rfloor \tag{36}$$

to achieve the following error $h_t^*\leq\epsilon$.

To prove this result, the main idea is to construct a path graph and a particular special problem in $\mathcal{P}_U^M$ such that to reduce h_t^*, it is necessary to traverse the entire graph once.

Next, for the problem class with non-uniform Lipschitz constants, we can extend the previous result to any network in class $\mathcal{N}$ (by properly assigning different values of L_i's to different nodes). In this case the lower bound will be dependent on the spectral property of $\widehat{\mathcal{L}}$ as defined in (22).

Corollary 3.1: Let ϵ be positive. For any given network in $\mathcal{N}$, and for any algorithm in $\mathcal{A}$, there exists a problem in $\mathcal{P}_L^M$ such that to achieve the accuracy $h_t^*<\epsilon$, it requires at least the following number of iterations and communication rounds

$$t\geq\frac{1}{3\sqrt{\xi(\widehat{\mathcal{L}})}}\left\lfloor\frac{\left(f(0)-\inf_x f(x)+\|d_0\|^2_{L^{-1}}/M\right)\bar{L}}{1650\pi^2}\epsilon^{-1}\right\rfloor. \tag{37}$$

IV. The Proposed Algorithms

In this section, we introduce our proposed algorithm for solving problem (2). The algorithm is *near-optimal*, and can achieve the lower bounds derived in Section III except for a multiplicative polylog factor in M. To simplify the notation, we rewrite problem (2) in the following compact form:

$$\min_{x\in\mathbb{R}^{SM}}\ f(x):=\frac{1}{M}\sum_{i=1}^{M}f_i(x_i),\quad\text{s.t. }(F\otimes I_S)x=0. \tag{38}$$

It can be verified that, by using the definition of F, the constraint in this problem is equivalent to the ones given in (2). For notational simplicity, in the following we will assume that $S=1$ (scalar variables). All the results presented in subsequent sections extend easily to case with $S>1$.

A. The xFILTER Algorithm

To motivate our algorithm design, observe that the communication lower bound $\mathcal{O}(1/\sqrt{\xi(\mathcal{G})}\times\bar{L}/\epsilon)$ in Section III can be decomposed into the product two parts, $\mathcal{O}(1/\sqrt{\xi(\mathcal{G})})$ and $\mathcal{O}(\bar{L}/\epsilon)$, corresponding roughly to the communication efficiency and the computational complexity, respectively. Such a product form motivates us to *separate* the computation and communication tasks, and design a *double loop* algorithm to achieve the desired lower bound.

Our proposed algorithm is based on a novel *approximate filtering -then- predict and tracking* (xFILTER) strategy, which properly combines the modern first-order optimization methods and the classical polynomial filtering techniques. It is a "double-loop" algorithm, where in the outer loop local gradients are computed to extract information from local functions, while in the inner loop some filtering techniques are used to facilitate efficient information propagation. Please see **Algorithm 1** for the detailed description, from the system perspective. It is important to note that the algorithm contains an outer loop **(S3)–(S4)** and an inner loop **(S2)**, indexed by r and q, respectively. Further, the local gradient evaluation only appears in the outer loop step **(S3)**.

To understand the algorithm, we note that one important task of each agent is to update its local variable so that it is close to the average $\frac{1}{M}\sum_{i=1}^{M}x_i$. Let us use d_i to denote a local variable that *approximates* the above average. At the beginning of the algorithm, d_i is just a rough estimate of the average, so we have $d_i=\frac{1}{M}\sum_j x_j+e_i$, where e_i is the *deviation* from the true average, and it can be viewed as some kind of "estimation noise". To gradually remove such a noise, in step **S1)** we resort to the so-called graph based joint bilateral filtering used for image denoising [46], [47], which can be formulated as the following regularized least squares problem:

$$x_*^{r+1}:=\arg\min_{x\in\mathbb{R}^M}\frac{1}{2}\|x-d^r\|^2_{\Upsilon^2}+\frac{1}{2}x^\top F^\top\Sigma^2Fx, \tag{39}$$

where d^r is the noisy signal, F is a penalty high pass filter related to the graph structure (in our case, F is the adjacency matrix), and Σ^2 is a regularization parameter. Its solution, denoted as x_*^{r+1} as given below, will be close to the "unfiltered" signal d^r, while having reduced high frequency components, or high fluctuations across the components:

$$Rx_*^{r+1}=d^r,\quad\text{with}\quad R:=\Upsilon^{-2}F^\top\Sigma^2F+I_M. \tag{40}$$

It is important to note that if x_*^{r+1} indeed achieves consensus, then by (10) we have $F^\top\Sigma^2Fx_*^{r+1}=0$, implying $x_*^{r+1}=d^r$, which says d^r should "track" x_*^{r+1}.

Unfortunately, the system (40) cannot be precisely solved in a distributed manner, because inverting R destroys its pattern about the network structure embedded in the product $F^\top\Sigma^2F$. More specifically, $F^\top\Sigma^2F$ is the weighted graph Laplacian matrix whose (i,j)th entry is nonzero if and only if node i, j are connected, but $(\Upsilon^{-2}F^\top\Sigma^2F+I_M)^{-1}$ is a dense matrix without such a property. Therefore in **S2)**, we use a degree-Q Chebyshev polynomial to approximate x_*^{r+1}. The output, denoted as x^{r+1}, stays in a Krylov space $span\{d^r,Rd^r,\cdots,R^Qd^r\}$. Specifically, at each iteration, the

only step that requires communication is the operation Ru, which is given by

$$(Ru_{q-1})[i] = (\Upsilon^{-2}F^\top\Sigma^2 F u_{q-1})[i] + d_{q-1}[i] \quad (41)$$
$$= \frac{1}{\beta_i^2}\sum_{j:j\sim i}\sigma_{ij}^2(u_{q-1}[j] - d^r[i]) + u_{q-1}[i],\ \forall\ i,$$

so this step can be done distributedly, via one round of local message exchange.

After completing $Q > 0$ such Chebyshev iterations (46) (C-iteration for short), the obtained solution x^{r+1} will be an approximate solution to the system 40, with a residual error vector ϵ^{r+1} as given below

$$Rx^{r+1} = d^r + R\epsilon^{r+1},\ \text{with}\quad \epsilon^{r+1} := x^{r+1} - x_*^{r+1}. \quad (42)$$

Up to this point, the filtering technique we have discussed aims at removing the "non-consensus" parts from a vector $d = [d_1, \cdots, d_N]^\top$. However, recall that the goal of distributed optimization is not only to achieve consensus, but also to optimize the objective function $\sum_i f_i(x_i)$. Therefore, a *prediction* step **(S3)** is performed to incorporate the most up-to-date local gradient $\nabla f_i(x_i)$, followed by a *tracking* step **(S4)** to update d. Ideally, one would like the new d_i^{r+1} to have the following three properties: 1) It is close to the previous d_i^r; 2) it takes into consideration the new local gradient information offered by the "predicted" $\tilde{x}_i^{r+1}$; 3) it is a "low frequency" signal, meaning d_i^{r+1} and d_j^{r+1} are relatively close, for all $i \neq j$. Taking a closer look at the "tracking" step, we can see that all three components are included: It adds to the previous d^r the differences of the last two predictions, and it removes some non-consensus components among the local variables.

To end this subsection, we emphasize that, the filtering step **(S2)** is critical to ensure that the proposed algorithm achieve performance lower bounds predicted in Section III. Intuitively, it helps to accelerate information propagation across the network. Indeed, as will be shown shortly, the number Q in **(S2)** is directly related to properties of the underlying graph.

B. Discussion

Below we provide remarks about the proposed algorithm.

Remark 4.1: **(xFILTER as a primal-dual strategy)** First, we provide an interesting interpretation of the xFILTER strategy. Let us introduce an auxiliary (dual) variable $\lambda^r \in \mathbb{R}^E$, which is updated as follows:

$$\lambda^{r+1} = \lambda^r + \Sigma^2 F x^{r+1},\ \text{with},\ \lambda^{-1} = 0. \quad (43)$$

Then according to (47), (48) and the initialization $x^{-1} = 0$, $d^{-1} = -\Upsilon^{-2}\nabla f(0)$, we have the following relationship

$$\begin{aligned} d^0 &:= -\Upsilon^{-2}\nabla f(x^{-1}) + (x^0 - \Upsilon^{-2}\nabla f(x^0) \\ &\quad - (x^{-1} - \Upsilon^{-2}\nabla f(x^{-1}))) - \Upsilon^{-2}F^\top\lambda^0 \\ &= x^0 - \Upsilon^{-2}\nabla f(x^0) - \Upsilon^{-2}F^\top\lambda^0. \end{aligned}$$

By using the induction argument, we can show that for all $r \geq 0$, the following holds

$$d^r := x^r - \Upsilon^{-2}\nabla f(x^r) - \Upsilon^{-2}F^\top\lambda^r. \quad (44)$$

Combining (40) and (44), we obtain the following useful alternative expressions of (40) and (42):

$$\Upsilon^{-2}\Big(\nabla f(x^r) + F^\top(\lambda^r + \Sigma^2 F x_*^{r+1})\Big) + (x_*^{r+1} - x^r) = 0 \quad (45a)$$
$$\Upsilon^{-2}\Big(\nabla f(x^r) + F^\top(\lambda^r + \Sigma^2 F x^{r+1})\Big) + (x^{r+1} - x^r) = R\epsilon^{r+1}. \quad (45b)$$

Algorithm 1. The xFILTER Algorithm (Global View)

(S1) [Initialization]. Assign each node $i \in \mathcal{N}$ with $\beta_i > 0$; Assign each edge $(ij) \in \mathcal{E}$ with $\sigma_{ij} > 0$; Initialize $x^{-1} = 0$, $d^{-1} = -\Upsilon^{-2}\nabla f(x^{-1})$ and $\tilde{x}^{-1} = x^{-1} - \Upsilon^{-2}\nabla f(x^{-1})$. Compute R by (40);

(S2) [Filtering]. At iteration $r+1$, $r \geq -1$: For a fixed constant $Q > 0$, run the following C-iterations (with parameters $\{\alpha_q, \tau\}$)

$$u_0 = x^r,\ u_1 = (I - \tau R)u_0 + \tau d^r, \quad (46)$$
$$u_q = \alpha_q(I - \tau R)u_{q-1} + (1 - \alpha_q)u_{q-2} + \tau\alpha_q d^r,\ q = 2, \cdots, Q;$$

Set $x^{r+1} = u_Q$;

(S3) [Prediction]. Compute $\tilde{x}^{r+1}$ by:

$$\tilde{x}^{r+1} = x^{r+1} - \Upsilon^{-2}\nabla f(x^{r+1}); \quad (47)$$

(S4) [Tracking]. Compute d^{r+1} by:

$$d^{r+1} = d^r + (\tilde{x}^{r+1} - \tilde{x}^r) - \Upsilon^{-2}F^\top\Sigma^2 F x^{r+1}. \quad (48)$$

Set $r = r + 1$, go to **(S2)**.

Using (45a), it is clear that x_*^{r+1} can be equivalently written as the optimal solution of the following problem:

$$\begin{aligned} x_*^{r+1} = \operatorname*{argmin}_x\ &\langle \nabla f(x^r) + F^\top\lambda^r, x - x^r\rangle \\ &+ \frac{1}{2}\|\Sigma F x\|^2 + \frac{1}{2}\|\Upsilon(x - x^r)\|^2. \end{aligned} \quad (49)$$

It follows that, the iterates $\{x^{r+1}\}$ can be viewed as trying to approximately optimize the *primal variable* of the following augmented Lagrangian (AL),

$$\mathsf{AL}(x, \lambda) = f(x) + \langle\lambda, Fx\rangle + \frac{1}{2}\|\Sigma F x\|^2 \quad (50)$$

while the update (43) updates the *dual variable*. For simplicity, we will use AL^r to denote $\mathsf{AL}(x^r, \lambda^r)$.

Remark 4.2: **(Implementation and Algorithm Classes)** First, to compute d^r's, note that $d^{-1} = -\Upsilon^{-2}\nabla f(0)$. Then if d^{r-1} is given, by combining (48) and (47), it is easy to show that each d_i^r can be updated as

$$\begin{aligned} d_i^r &= d_i^{r-1} + (x_i^r - x_i^{r-1}) - \frac{1}{M\beta_i^2}(\nabla f_i(x_i^r) - \nabla f_i(x_i^{r-1})) \\ &\quad + \sum_{j:j\sim i}\frac{\sigma_{ij}^2}{\beta_i^2}(x_i^r - x_j^r). \end{aligned} \quad (51)$$

Combining the above expression with (41) for computing Ru_{q-1}, it is clear that all the computation only involves local communication and local gradient computation.

The above observation also suggests that for a general choice of parameter matrix $\Sigma^2 \succ 0$, xFILTER is in class $\mathcal{A}$. Further, if Σ^2 is a multiple of identity matrix (i.e., there exists $\sigma^2 > 0$ such that $\Sigma^2 = \sigma^2 I_E$), then the computations in (51) only involve the sum of neighboring iterates, therefore the algorithm belongs to class $\mathcal{A}'$ as well.

V. The Convergence Rate Analysis

In this section we provide the analysis of the convergence rate of xFILTER. All the proofs can be found in the appendix. For convenience, the algorithm will be analyzed based on the primal-dual interpretation in Remark 4.1.

Step 1. We first analyze the dynamics of the dual variable.

Lemma 5.1: Suppose that $f(x)$ is in class $\mathcal{P}_L^M$. Then, for all $r \geq 0$, the iterates of xFILTER satisfy

$$\|\lambda^{r+1}-\lambda^r\|^2_{\Sigma^{-2}} \leq \widetilde{\kappa}\bigg(\frac{3}{M^2}\|\Upsilon^{-1}L(x^r-x^{r-1})\|^2 + 3\|w^{r+1}\|^2_{\Upsilon^2} + 3\|\Upsilon R(\epsilon^{r+1}-\epsilon^r)\|^2\bigg). \tag{52}$$

where we have defined the following

$$\widetilde{\kappa} := \frac{1}{\underline{\lambda}_{\min}(\Sigma F\Upsilon^{-2}F^\top\Sigma)} = \frac{1}{\underline{\lambda}_{\min}(\mathcal{L}_G)} \tag{53a}$$

$$w^{r+1} := (x^{r+1}-x^r)-(x^r-x^{r-1}). \tag{53b}$$

Step 2. In this step we analyze the dynamics of the AL (38).

Lemma 5.2: For all $r \geq 0$, the iterates of xFILTER satisfy

$$\begin{aligned}\mathsf{AL}^{r+1}-\mathsf{AL}^r \leq& -\frac{1}{2}\|x^{r+1}-x^r\|^2_{\Upsilon^2R-\frac{L}{M}} \\ &+\langle\Upsilon^2R\epsilon^{r+1},x^{r+1}-x^r\rangle+\frac{3\widetilde{\kappa}}{M^2}\|\Upsilon^{-1}L(x^r-x^{r-1})\|^2 \\ &+3\widetilde{\kappa}\|w^{r+1}\|^2_{\Upsilon^2}+3\widetilde{\kappa}\|\Upsilon R(\epsilon^{r+1}-\epsilon^r)\|^2.\end{aligned} \tag{54}$$

Step 3. In this step, we analyze the error sequences $\{\epsilon^{r+1}\}$ generated by the xFILTER. First we have the following well-known result on the behavior of the Chebyshev iteration; see, e.g., [48, Chapter 6] and [49, Theorem 1, Chapter 7].

Lemma 5.3: Consider using the Chebyshev iteration (46) *to solve $Rx=d^r$. Define $x_*^{r+1}=R^{-1}d^r$, with*

$$R := \Upsilon^{-2}(F^\top\Sigma^2F+\Upsilon^2). \tag{55}$$

Define the following constants:

$$\begin{aligned}&\xi(R) := \frac{\lambda_{\min}(R)}{\lambda_{\max}(R)} \leq 1,\ \xi(\Upsilon^2) := \frac{\lambda_{\min}(\Upsilon^2)}{\lambda_{\max}(\Upsilon^2)} \leq 1,\\ &\theta(R) := \lambda_{\min}(R)+\lambda_{\max}(R).\end{aligned} \tag{56}$$

Choose the following parameters:

$$\tau = \frac{2}{\theta(R)},\ \alpha_1=2,\ \alpha_{t+1}=\frac{4}{4-\rho_0^2\alpha_t},\ \rho_0=\frac{1-\xi(R)}{1+\xi(R)}.$$

Then for any $\eta \in (0,1)$, achieving the following accuracy

$$\|u_Q-x_*^{r+1}\|^2_{\Upsilon^2} \leq \eta\|u_0-x_*^{r+1}\|^2_{\Upsilon^2}, \tag{57}$$

requires the following number of iterations

$$Q \geq -\frac{1}{4}\ln(\eta/4)\sqrt{1/\xi(R)}.$$

Recall that in Algorithm 1 the initial and final solutions for the Chebyshev iteration are assigned to x^r and x^{r+1}, respectively. Define $\widetilde{\epsilon}^r := u_0-x_*^{r+1}=x^r-x_*^{r+1}$, which is the error before running the C-iteration. We have

$$Rx^r = Ru_0 = R(u_0-x_*^{r+1})+Rx_*^{r+1} := R\widetilde{\epsilon}^r+d^r, \forall\ r\geq -1.$$

Plugging in the definition of d^r in (44), we obtain

$$R\widetilde{\epsilon}^r = Rx^r+\Upsilon^{-2}(\nabla f(x^r)+F^\top\lambda^r-\Upsilon^2x^r). \tag{58}$$

Using the definition of ϵ^{r+1} in (45b), and the fact that R is invertible, we obtain the following key relationship

$$\epsilon^{r+1}-\widetilde{\epsilon}^r = x^{r+1}-x^r,\ \forall\ r\geq -1. \tag{59}$$

Recall that $\epsilon^{r+1} := x^{r+1}-x_*^{r+1}$, and $x^{r+1}=u_Q$, $x^r=u_0$, then (57) implies

$$\|\epsilon^{r+1}\|^2_{\Upsilon^2} \leq \eta\|\widetilde{\epsilon}^r\|^2_{\Upsilon^2}. \tag{60}$$

By combining Lemma 5.3, (59) and (60), the following result provides some essential relationships between the error sequences $\{\epsilon^{r+1}\}$ with the outer-loop iterates $\{x^{r+1}\}$.

Lemma 5.4: Choose the inner iteration of xFILTER as

$$Q = -\frac{1}{4}\ln\bigg(\frac{\theta^2}{16+128M\max\{\lambda_{\max}(\Upsilon^2R),1\}}\bigg)\sqrt{1/\xi(R)}. \tag{61}$$

where $\theta := \xi(\Upsilon^2R)\xi(\Upsilon^2)\times\min\{1,\lambda_{\min}(\Upsilon^2)\}$. Then we have the following inequalities

$$\|\Upsilon^2R\epsilon^{r+1}\|^2 \leq \frac{1}{16M}\|x^{r+1}-x^r\|^2_{\Upsilon^2R}, \tag{62a}$$

$$\|\epsilon^{r+1}\|^2_{\Upsilon^2R} \leq \frac{1}{16M}\|x^{r+1}-x^r\|^2_{\Upsilon^2R}, \tag{62b}$$

$$\|\Upsilon R\epsilon^{r+1}\|^2 \leq \frac{1}{16M}\|x^{r+1}-x^r\|^2_{\Upsilon^2R}, \tag{62c}$$

$$\langle\Upsilon^2R\epsilon^{r+1},x^{r+1}-x^r\rangle \leq \frac{3}{16}\|x^{r+1}-x^r\|^2_{\Upsilon^2R}, \tag{62d}$$

$$\langle\Upsilon^2R\epsilon^r,x^{r+1}-x^r\rangle \leq \frac{1}{8}\|x^r-x^{r-1}\|^2_{\Upsilon^2R}+\frac{1}{16}\|x^{r+1}-x^r\|^2_{\Upsilon^2R}. \tag{62e}$$

Clearly, using the Chebyshev iteration is one critical step that ensures fast reduction of the error $\{\epsilon^{r+1}\}$. In particular, to achieve constant reduction of error, the total number of required Chebyshev iteration is proportional to $\sqrt{1/\xi(R)}$, rather than $1/\xi(R)$ in conventional iterative scheme such as the Richardson's iteration [48]. Such a choice enables the final bound to be dependent on $\sqrt{1/\xi(\mathcal{G})}$, rather than $1/\xi(\mathcal{G})$.

Step 4. Let us construct the following potential functions (parameterized by constants $\widetilde{c}>0$)

$$\begin{aligned}\widetilde{P}_{\widetilde{c}}(x^{r+1},x^r,\lambda^{r+1}) :=& \mathsf{AL}^{r+1}+\frac{3\widetilde{\kappa}}{M^2}\|\Upsilon^{-1}L(x^{r+1}-x^r)\|^2 \\ &+\frac{3\widetilde{\kappa}}{8}\|x^{r+1}-x^r\|^2_{\Upsilon^2R}+\frac{\widetilde{c}}{2}\|\Sigma Fx^{r+1}\|^2+\frac{\widetilde{c}}{2}\|x^{r+1}-x^r\|^2_{\Upsilon^2+\frac{\Upsilon^2R}{4}+\frac{L}{M}}.\end{aligned} \tag{63}$$

For notational simplicity we will denote it as $\widetilde{P}^{r+1}$. Next we show that when the algorithm parameters are chosen properly, the potential functions will decrease along the iterations.

Lemma 5.5: Suppose that $f(x)$ is in class $\mathcal{P}_L^M$, Q is chosen according to (61)*, and the rest of the parameters of xFILTER are chosen as below*

$$\tilde{c} = 8\tilde{\kappa} = \frac{8}{\underline{\lambda}_{\min}(\Sigma F\Upsilon^{-2}F^\top\Sigma)},\ \Upsilon^2 \succeq \frac{L\Upsilon^{-2}L}{M^2}, \tag{64a}$$

$$(1/4-3\tilde{\kappa}-\tilde{c})\Upsilon^2R-(1+2\tilde{c})L/M-\frac{6\tilde{\kappa}}{M^2}L\Upsilon^{-2}L \succeq 0. \tag{64b}$$

Then for all $r\geq 0$, we have

$$\widetilde{P}^r-\widetilde{P}^{r+1} \geq \frac{1}{8}\|x^{r+1}-x^r\|^2_{\Upsilon^2R}+\tilde{\kappa}\|w^{r+1}\|^2_{\Upsilon^2}. \tag{65}$$

Step 5. Next we show the boundedness of $\{\widetilde{P}^{r+1}\}$.

Lemma 5.6: Suppose that $f(x)$ is in class $\mathcal{P}_L^M$ and the parameters are chosen according to (64) *and* (61)*. Then the*

sequence $\{\widetilde{P}^{r+1}\}$ *generated by xFILTER satisfies*

$$\widetilde{P}^{r+1} \geq \underline{f},\ \forall\ r>0, \quad \widetilde{P}^{0} \leq f(x^0)+\frac{5}{M}d_0^\top L^{-1}d_0. \quad (66)$$

where $\underline{f}$ *and* d_0 *are defined in* (6) *and* (33)*, respectively.*

Step 6. We are ready to derive the final bounds for the convergence rate of the proposed algorithm.

Theorem 5.1: Suppose that $f(x)$ *is in class* $\mathcal{P}_L^M$ *and the parameters are chosen according to* (64) *and and* (61)*. Let* T_r *denote the outer iteration index in which xFILTER satisfies*

$$e(T_r):=\min_{r\in[T_r]}\left\|1/M\sum_{i=1}^{M}\nabla f_i(x_i^r)\right\|^2+\|\Sigma F x^r\|^2\leq\epsilon. \quad (67)$$

Then we have the following bound for the error:

$$\epsilon\leq\widetilde{C}_1\times\frac{\widetilde{C}_2}{T_r}, \quad (68)$$

with the following constants

$$\widetilde{C}_1:=f(x^0)-\underline{f}+\frac{5}{M}d_0^\top L^{-1}d_0 \quad (69a)$$

$$\widetilde{C}_2:=128\left(\sum_{i=1}^{M}\beta_i^2+3+\frac{1}{32\widetilde{\kappa}}\right). \quad (69b)$$

VI. Rate Bounds and Tightness

In this section we provide explicit choices of various parameters, and discuss the tightness of the resulting bounds.

A. Parameter Selection and Rate Bounds for xFILTER

First, recall that the matrices $\widetilde{L}$ and $\widehat{\mathcal{L}}$ are defined in (21)-(22). Below we will provide two choices of parameters.

Choice I. We focus on a class of graphs such that there exists an *absolute* constant $k>0$ such that the following holds :

$$kP\succeq\bar{d}I_M \quad (70)$$

where $\bar{d}$ is the averaged degree (7). Condition (70) says that the degrees of the nodes are not very different from their average. For example the following graphs satisfy (70): Complete graph ($k=1$), star graph ($k=2$), grid graph ($k=2$), cubic graph ($k=1$), path graph ($k=2$), and any regular graph ($k=1$).

For the class of graphs satisfies (70), let us pick the parameters for xFILTER as follows:

$$\Sigma^2=\frac{48\times 96k}{\sum_i d_i\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}K, \quad \Upsilon^2=\frac{96k}{\sum_i d_i}P^{1/2}LP^{1/2}. \quad (71)$$

Using the above choice, we have

$$\beta_i^2=\frac{96L_id_ik}{\sum_i d_i}=\frac{96L_id_ik}{M\bar{d}} \quad (72)$$

and that the matrix Υ satisfies the following

$$\Upsilon^2=\frac{96k}{\sum_i d_i}P^{1/2}LP^{1/2}\succeq\frac{96}{M}L. \quad (73)$$

Plugging these choices to $\mathcal{L}_G$ in (19) we obtain

$$\begin{aligned}\mathcal{L}_G&=\Upsilon^{-1}F^\top\Sigma^2F\Upsilon^{-1} \quad (74)\\ &=\frac{48}{\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}L^{-1/2}P^{-1/2}F^\top KFP^{-1/2}L^{-1/2}\\ &=\frac{48}{\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}\widetilde{\mathcal{L}}.\end{aligned}$$

Therefore by (53a) we have

$$\tilde{\kappa}\ =\frac{\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}{48\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}=\frac{1}{48}. \quad (75)$$

Also in this case we have

$$\begin{aligned}R&=\Upsilon^{-2}F^\top\Sigma^2F+I\\ &=\frac{48}{\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}P^{-1/2}L^{-1}P^{-1/2}F^\top KF+I.\end{aligned}$$

By noting that the matrix $P^{-1/2}L^{-1}P^{-1/2}F^\top KF$ and $\widetilde{\mathcal{L}}$ share the same set of eigenvalues, we obtain

$$\lambda_{\max}(R)\leq\left(\frac{48\lambda_{\max}(\widetilde{\mathcal{L}})}{\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}+1\right)\leq\frac{50}{\xi(\widetilde{\mathcal{L}})},\ \lambda_{\min}(R)=1, \quad (76a)$$

$$\xi(R)\geq 1/\left(\frac{48\lambda_{\max}(\widetilde{\mathcal{L}})}{\underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}+1\right)\geq\frac{\xi(\widetilde{\mathcal{L}})}{50}. \quad (76b)$$

Choice II. For general graphs not necessarily satisfying (70), let us pick the parameters for xFILTER as follows

$$\Sigma^2=\frac{48\times 96}{M\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}K, \quad \Upsilon^2=\frac{96}{M}L. \quad (77)$$

Using the above choice, we have

$$\beta_i^2=\frac{96L_i}{M}. \quad (78)$$

We have that

$$\mathcal{L}_G=\frac{48}{\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}L^{-1/2}F^\top KFL^{-1/2}=\frac{48}{\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}\widehat{\mathcal{L}}. \quad (79)$$

Therefore by (53a) we have

$$\tilde{\kappa}\ =\frac{\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}{48\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}=\frac{1}{48}. \quad (80)$$

Also in this case we have

$$R=\Upsilon^{-2}F^\top\Sigma^2F+I=\frac{48}{\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}L^{-1}F^\top KF+I.$$

By noting that the matrix $L^{-1}F^\top KF$ and $\widehat{\mathcal{L}}$ share the same set of eigenvalues, we obtain

$$\lambda_{\max}(R)\leq\left(\frac{48\lambda_{\max}(\widehat{\mathcal{L}})}{\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}+1\right)\leq\frac{50}{\xi(\widehat{\mathcal{L}})},\ \lambda_{\min}(R)=1, \quad (81a)$$

$$\xi(R)\geq 1/\left(\frac{48\lambda_{\max}(\widehat{\mathcal{L}})}{\underline{\lambda}_{\min}(\widehat{\mathcal{L}})}+1\right)\geq\frac{\xi(\widehat{\mathcal{L}})}{50}. \quad (81b)$$

Remark 6.1: **(Choices of Parameters)** The above two choices of parameters differ on whether Υ^2 is scaled with the degree matrix or not. The resulting bounds are also dependent on the spectral gap for $\widetilde{L}$ and $\widehat{\mathcal{L}}$, one inversely scaled with the degree matrix, and the other does not. Note that the spectral gap of $\widetilde{L}$ and $\widehat{\mathcal{L}}$ may not be the same. For example for a star graph with $L_i=L_j$, $\xi(\widehat{\mathcal{L}})=\mathcal{O}(1/M)$ but $\xi(\widetilde{\mathcal{L}})=\mathcal{O}(1)$. Therefore one has to be careful in choosing these parameters so that $\xi(R)$ is made as large as possible.

Additionally, since we are mainly interested in choosing the parameters so that the resulting rate bounds will be optimal

in their dependency on problem parameters, the absolute constants in the above parameter choices have not been optimized.

The following result is a consequence of Theorem 5.1.

Theorem 6.1: Consider using xFILTER to solve problems in class $(\mathcal{P}_L^M, \mathcal{N})$, then the following holds.

Case I. Further restricting $\mathcal{N}_D^M$ to a subclass satisfying (70). If parameters in (71) is used, then the condition (64b) will be satisfied. Further, to achieve $e(T) \leq \epsilon$, it requires at most the following number of iterations (where T denotes the *total* iterations of the xFILTER algorithm)

$$T \leq \frac{1}{\epsilon}\left(f(x^0) - \underline{f} + \frac{5}{M}\|d_0\|^2_{L^{-1}}\right) \times \widetilde{C}_2 \times \frac{1}{4}\ln\left(\frac{(ML_{\max}/L_{\min})^4 \times (16+6400M)}{\xi^3(\widetilde{\mathcal{L}})}\right)\sqrt{50/\xi(\widetilde{\mathcal{L}})} \tag{82}$$

where $\widetilde{C}_2$ is given by

$$\widetilde{C}_2 \leq 128\left(\frac{96k}{\sum_{i=1}^M d_i}\sum_{i=1}^M d_i L_i + 19\right). \tag{83}$$

Case II. Suppose parameters in (77) are used. Then the condition (64b) will be satisfied. Further, to achieve $e(T) \leq \epsilon$, it requires at most the following number of iterations

$$T \leq \frac{1}{\epsilon}\left(f(x^0) - \underline{f} + \frac{5}{M}\|d_0\|^2_{L^{-1}}\right) \times \widetilde{C}_2 \times \frac{1}{4}\ln\left(\frac{(ML_{\max}/L_{\min})^4 \times (16+6400M)}{\xi^3(\widehat{\mathcal{L}})}\right)\sqrt{50/\xi(\widehat{\mathcal{L}})} \tag{84}$$

where $\widetilde{C}_2$ is given by

$$\widetilde{C}_2 \leq 128\left(\frac{96}{M}\sum_{i=1}^M L_i + 19\right). \tag{85}$$

We note that compared with (68) in Theorem 5.1, the additional multiplicative term in (82) accounts for the Chebyshev iterations that are needed for every outer iteration r.

B. Tightness of the Upper Rate Bounds

We present some tightness results of the upper rate bounds for xFILTER. In particular, we compare the expressions derived in Theorem 6.1, and the lower bounds derived in Section III, over different kinds of graphs and for different problems. We will mainly focus on the case with uniform Lipschitz constants, i.e., $L_i = U,\ \forall\ i$. We will briefly discuss the case of non-uniform Lipschitz constants at the end of this section.

First, consider the class $\mathcal{P}_U^M$ with the following properties:

$$L_1 = L_2 = \cdots L_M = \frac{1}{M}\sum_{i=1}^M L_i := U, \quad L = UI_M. \tag{86}$$

It follows that in this case $\widetilde{\mathcal{L}} = \mathcal{L}$, and $\widehat{\mathcal{L}} = P^{1/2}\mathcal{L}P^{1/2}$. Let us first make some useful observations.

Remark 6.2: Let us specialize the parameter choices for xFILTER algorithm in (71) and derive the bounds for $\widetilde{C}_2 \times 1/\sqrt{\xi(\widetilde{\mathcal{L}})}$ in (83) for the following special graphs.

Complete graph. Complete graphs satisfy (70) with $k = 1$. It also satisfies $\underline{\lambda}_{\min}(\widetilde{\mathcal{L}}) = M/(M-1) \geq 1$. Therefore using the expression (83) we obtain the following:

$$\widetilde{C}_2^{\text{comp}} \times \frac{1}{\sqrt{\xi(\widetilde{\mathcal{L}})}} \leq 12500U + 2560. \tag{87}$$

Grid graph. Grid graphs satisfy (70) with $k = 2$. It also satisfies $\underline{\lambda}_{\min}(\widetilde{\mathcal{L}}) \geq 1/M$. Therefore using the expression (83) we obtain the following:

$$\widetilde{C}_2^{\text{grid}} \times \frac{1}{\sqrt{\xi(\widetilde{\mathcal{L}})}} \leq (12500U + 2560) \times \sqrt{M}. \tag{88}$$

Star graph. Star graphs satisfy (70) with $k = 2$. It also has $\xi(\widetilde{\mathcal{L}}) = 1/2$. Therefore using the expression (83) we obtain the following:

$$\widetilde{C}_2^{\text{star}} \times \frac{1}{\sqrt{\xi(\widetilde{\mathcal{L}})}} \leq (12500U + 2560) \times \sqrt{2}. \tag{89}$$

Random Geometric graph. If the radius Ra satisfies (25), then with high probability $\xi(\widetilde{\mathcal{L}}) = \mathcal{O}\left(\frac{\log(M)}{M}\right)$. Further, from the proof of [50, Lemma 10], for any ϵ and $c > 0$, if

$$Ra = \Omega\left(\sqrt{\log^{1+\epsilon}(M)/(M\pi)}\right) \tag{90}$$

then with probability at least $1 - 2/M^{c-1}$, the following holds

$$\log^{1+\epsilon} M - \sqrt{2}c\log M \leq d_i \leq \log^{1+\epsilon} M + \sqrt{2}c\log M,\ \forall\ i.$$

This means that (70) is satisfied (with $k = \mathcal{O}(1)$) with high probability (also see discussion at the end of [44, Section V]). Therefore using the expression (83) we obtain the following:

$$\widetilde{C}_2^{\text{geometric}} \times \frac{1}{\sqrt{\xi(\widetilde{\mathcal{L}})}} \leq (12500U + 2560) \times \mathcal{O}\left(\frac{\sqrt{M}}{\sqrt{\log(M)}}\right).$$

Cycle/Path graph. Cycle/path graphs satisfy (70) with $k = 2$. We also have $\underline{\lambda}_{\min}(\widetilde{\mathcal{L}}) \geq 1/M^2$. Therefore using the expression (83) we obtain the following:

$$\widetilde{C}_2^{\text{cycle}} \times \frac{1}{\sqrt{\xi(\widetilde{\mathcal{L}})}} \leq (12500U + 2560) \times M. \tag{91}$$

We also note that for the xFILTER algorithm, the fact that $L_i = U,\ \forall\ i$ implies that the matrix Σ^2 given in (71) is a multiple of identity matrix. Therefore by Remark 4.2, we can conclude that in this case xFILTER belongs to both $\mathcal{A}$ and $\mathcal{A}'$.

Now we are ready to present our tightness analysis.

Theorem 6.2: Consider the problem class P_U^M, and a subclass of $\mathcal{N}$ satisfying (70). Then the convergence rate in (82) is tight (up to a polylog factor).

Proof. When $L_i = L_j,\ \forall\ i \neq j$, and when (70) is satisfied, it is easy to verify that the following holds

$$h_T^* \leq e(T), \text{ and } \widetilde{\mathcal{L}} = \mathcal{L}. \tag{92}$$

To bound the total number of iteration required to achieve $h_T^* \leq \epsilon$, note that when (70) is satisfied, we can apply the

bound (82) in Theorem 6.1 and obtain

$$T \leq \frac{1}{\epsilon}\left(f(x^0) - \underline{f} + \frac{5}{MU}\|d_0\|^2\right) \times 128\,(96kU + 19) \times \frac{1}{4}\ln\left(\frac{M^4 \times (16 + 6400M)}{\xi^3(\mathcal{G})}\right)\sqrt{50/\xi(\mathcal{G})}. \quad (93)$$

Comparing with the lower bound in Theorem 3.1, it is clear that except for the multiplicative $\ln(\cdot)$ term, the remaining bound is in the same order as the lower bound (36). **Q.E.D.**

Remark 6.3: **(Optimal Number of Gradient Evaluations)** It is important to note that the "outer" iteration of the xFILTER required to achieve ϵ-local solution scales with $\mathcal{O}(U/\epsilon)$, which is independent of the network size. Because local gradients are only evaluated in the outer iterations, the above fact suggests that the total number of gradients $\nabla f(x^r)$ required is also in $\mathcal{O}(U/\epsilon)$. This is an optimal order because it is the same as what is needed for the centralized gradient descent.

Remark 6.4: **(Performance Gap Compared with Existing Methods)** An existing algorithm called distributed gradient primal-dual algorithm (D-GPDA) has also been developed recently, which has explicit characterization of various convergence rates [51]. In particular, this algorithm is not optimal, in the sense that at each iteration, $\mathcal{O}(1)$ local communication and gradient computation are to be carried out, and the total number of iterations scale with $\mathcal{O}\left(\frac{1}{\epsilon} \times \frac{1}{\xi(\mathcal{G})}\right)$. Obviously the D-GPDA algorithm costs a lot more compared with xFILTER, for example for path/star graph, it requires $\mathcal{O}(M^2)$ times more gradient computation effort, and $\mathcal{O}(M)$ times more communication effort than what is required by the xFILTER.

Remark 6.5: **(Non-uniform Lipschitz Constants)** We comment that for the general case $L_i \neq L_j$, $\forall\ i, j$, we can use similar steps to verify that the bound (84) derived in Theorem 6.1 is optimal, in the sense that they achieve the lower bound (37) predicted in Corollary 3.1.

VII. Numerical Results

This section presents numerical examples to show the effectiveness of the proposed algorithms. Two kinds of problems are considered, distributed binary classification and distributed neural networks training. We use the former one to demonstrate the behavior and scalability of our algorithm and use the latter one to show the practical performance.

A. Simulation Setup

In our simulations, all algorithms are implemented in MATLAB R2017a for binary classification problem and implemented in Python 3.6 for training neural networks, running on a computer node with two 12-core Intel Haswell processors and 128 GB of memory (unless otherwise specified). Both synthetic and real data are used for performance comparison. For synthetic data, the feature vector is randomly generated with standard normal distribution with zero mean and unit variance. The label vector is randomly generated with uniformly distributed pseudorandom integers taking the values $\{-1, 1\}$. For real data, we use the breast cancer dataset [1] for binary classification and MNIST[2] for training neural network. The breast cancer dataset contains a total of 569 samples each with 30 real positive features. The MNIST dataset contains a total of 60,000 handwritten digits, each with a 28×28 gray scale image and a label from ten categories.

B. Distributed Binary Classification

We consider a non-convex distributed binary classification problem [52], where each component function f_i is given by

$$f_i(x_i) = \frac{1}{B}\sum_{j=1}^{B}\log\left(1 + \exp(-y_{ij}x_i^\top v_{ij})\right) + \sum_{s=1}^{S}\frac{\lambda\alpha x_{i,s}^2}{1 + \alpha x_{i,s}^2}.$$

Here $v_{ij} \in \mathbb{R}^S$ denotes the feature vector with dimension S, $y_{ij} \in \{1, -1\}$ denotes the label for the jth date point in ith agent, and there are total B data points for each agent. Unless otherwise noted, the graph $\mathcal{E}$ used in our simulation is generated using the random geometric graph and the graph parameter Ra is set to 0.5. The regularization parameter is set to $\lambda = 0.001, \alpha = 1$.

To compare the convergence performance of the proposed algorithms, we randomly generated MB data points with dimension K and distribute them into M nodes, i.e. each node contains B data points with K features. Then we compare xFILTER with the D-GPDA [51], the distributed subgradient (DSG) method [53], the Push-sum algorithm [54], and the NEXT algorithm [14]. The parameters for NEXT are chosen as $\tau = 1, \alpha[0] = 0.1$ and $\mu = 0.01$ as suggested by [14], while the parameters for xFILTER are chosen based on (71).

Simulation results on synthetic data for different M, B, K averaged over 30 realizations are investigated and shown in Fig. 2 to Fig. 3, where the x-axis denotes the total rounds of communications required, and the y-axis denotes the quality measure (16). Note that the curves xFILTER (outer) included in these figures show the number of communication rounds required for xFILTER to perform the "outer" iterations (which is equivalent to r in Algorithm 1, since in each outer iteration only one round of communication is required in Step **S3**). The performance evaluated on real data is also characterized in Fig. 4, in which we choose $M = 10$, $B = 56$, and $K = 30$. These results show that the proposed algorithms perform well in all parameter settings compared with existing methods.

We further note that these figures also show (rough) comparison about computation efficiency of different algorithms. Specifically, for GPDA, DSG and Push Sum (resp. NEXT), the total rounds of communication is the same as (resp. twice as) the total number of gradient evaluations per node. In contrast, the total rounds of communication in the *outer loop* of xFILTER is the same as the local gradient evaluations. Therefore, the comparison between xFILTER (outer) and other algorithms in Fig. 2 to Fig. 3 shows the relative computational efficiency of these algorithms. Clearly, xFILTER has a significant advantage over the rest of the algorithms.

Further, we compare the scalability performance of the proposed algorithms with increased network dimension M. In particular, in Fig. 5 we compare the total communication rounds required for NEXT and the xFILTER for reaching

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic) [2] http://yann.lecun.com/exdb/mnist/

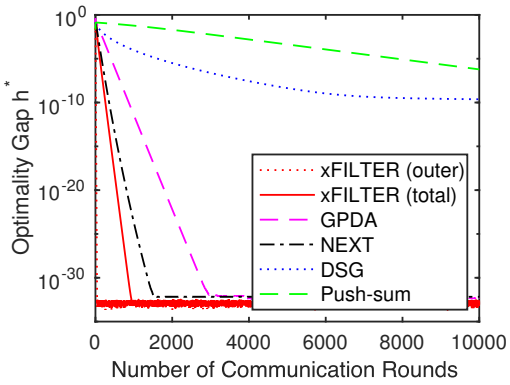


Fig. 2: $M = 20, B = 200, K = 10$

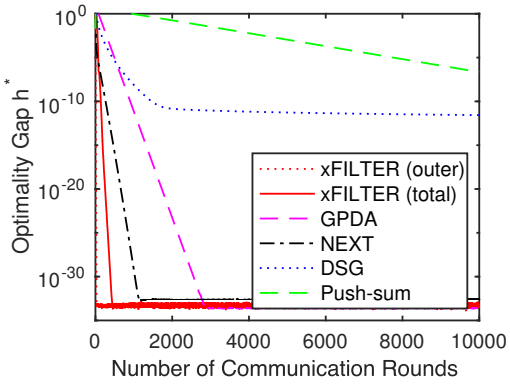


Fig. 3: $M = 50, B = 2000, K = 10$

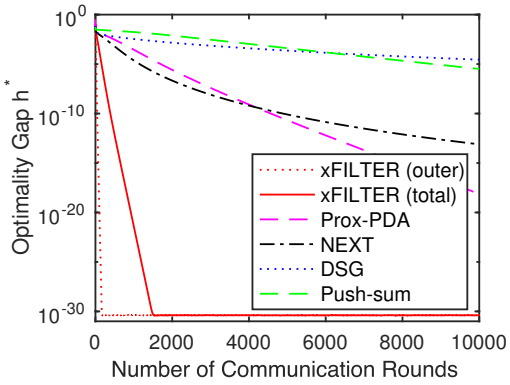
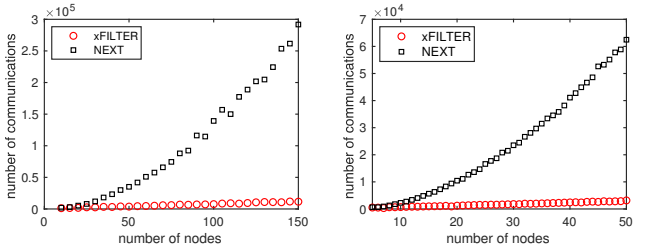


Fig. 4: $M = 10, B = 56, K = 30$

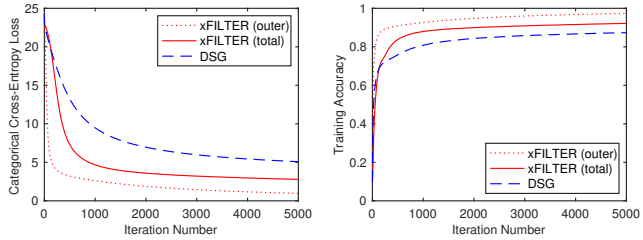
$h_T^* \leq 10^{-10}$ and $h_T^* \leq 10^{-15}$, over path graphs with increasing number of nodes. Overall, we see that the xFILTER performs reasonably fast and exhibits the desired linear scaling.

We do want to point out that although the proposed algorithms compare relatively favorably with NEXT in our numerical tests, NEXT can in fact handle a larger class of problems because it is designed for nonsmooth and constrained nonconvex problems. Further, for all the algorithms we have used, we did not tune the parameters: For xFILTER and D-GPDA, we use the theoretical upper bound suggested in Theorem 5.1, and for NEXT we use the parameters suggested in the paper [14]. It could be possible to fine-tune the stepsizes to make them faster, but since this paper is mostly on the theoretical properties of rate optimal algorithms, we choose not to go down that path.



(a) $B = 10, K = 10, \epsilon = 10^{-10}$ (b) $B = 200, K = 10, \epsilon = 10^{-15}$

Fig. 5: Comparison of NEXT and xFILTER over path graphs with increasing number of nodes ($M \in [10,\ 150]$ in (a) and $M \in [5,\ 50]$ in (b)). Each point in the figure represents the total number of communication needed to reach $h_T^* \leq \epsilon$.



(a) Training Loss (b) Training Accuracy

Fig. 6: Comparison of DSG and xFILTER over path graphs on distributed training neural networks; Plot (a) shows the dynamic of the categorical cross-entropy loss, and plot (b) shows the training classification accuracy. The parameters are chosen based on their best practical performance through grid search. The curves xFILTER (outer) and xFILTER (total) again represent the number of outer iteration, and the total number of iterations required for xFILTER.

C. Distributed Neural Network Training

In our second experiment, we present some numerical results under a more realistic setting. We consider training a neural network model for fitting the MNIST data set. The dataset is first randomly partitioned into 10 subsets, and then gets distributed over 10 machines. A fully connected neural network with one hidden layer is used in the experiment. The number of neurons for the hidden layer and the output layer are set as 128 and 10, respectively. The initial weights for the neural network are drawn from a truncated normal distribution centered at zero with variance scaled with the number of input units. The algorithms are written in Python, and the communication protocol is implemented using the Message Passing Interface (MPI). The empirical performance of the xFILTER is evaluated and compared with the DSG algorithm [53]. Fig. 6 shows that, compared with DSG, the proposed algorithm achieves better communication and computation efficiency, and has improved classification accuracy.

Note that despite the fact that some global parameters (such as the Lipschitz constants) are unknown, the rules provided in (71) or (77) still can help us roughly estimate a set of good parameters. For example, we choose the following parameters

$$\Sigma^2 = \frac{\sigma}{\sum_i d_i \underline{\lambda}_{\min}(\widetilde{\mathcal{L}})}, \quad \Upsilon^2 = \frac{\beta P}{\sum_i d_i}, \tag{94}$$

and tune the parameter β and σ by search from the sets $\{0.1, 0.2, 0.5, 1, 2, 5, \cdots, 100, 200, 500\}$. Based on the best practical performance over 10 runs, we choose $\beta = 100$ and $\sigma = 20$ for xFILTER and $\alpha = 0.1$ for DSG.

VIII. Conclusion and Future Works

This paper represents the first work that investigates the performance of optimal first-order algorithms for non-convex distributed optimization problems. We provide a lower complexity bound that characterizes the worst case performance for any algorithm in class $\mathcal{A}$, and propose an algorithm capable of (nearly) achieving the lower bound in various settings. In Fig. 7, we illustrate various bounds discussed in this work by using a path graph.

To the best of our knowledge, the proposed algorithm is the first and the only available distributed non-convex algorithm in class $\mathcal{A}$ that can achieve the (near) optimal rate performance for problem/network classes $(\mathcal{P},\mathcal{N})$. However, they still require some global information to initialize the parameters, so it will be of interest to design global information free algorithms that only require local structures to set parameters (just like in the convex case, see discussions in [55]). It will also be desirable to consider the problem where only the average function $\bar{f}$ has Lipschitz gradient, but not the local f_i's.

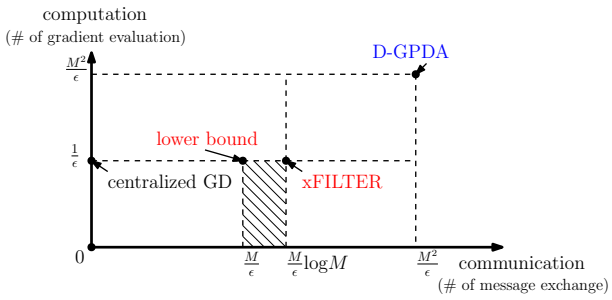


Fig. 7: Graphical comparison of various bounds analyzed in this work, illustrated over a path graph with M nodes.

IX. Appendix

A. Proof of Lemma 5.1

Proof. For simplicity we will denote $g^r := \nabla f(x^r)$. First note that $\forall\ r \geq -1$ the following holds according to (45b),

$$g^r + F^\top(\lambda^r + \Sigma^2 F x^{r+1}) + \Upsilon^2(x^{r+1} - x^r) = \Upsilon^2 R\epsilon^{r+1}. \quad (95)$$

Second, by using (95) and the update (43), we obtain

$$F^\top\lambda^{r+1} = -g^r - \Upsilon^2(x^{r+1} - x^r) + \Upsilon^2 R\epsilon^{r+1}. \quad (96)$$

Then subtracting the previous iteration leads to

$$F^\top(\lambda^{r+1} - \lambda^r) = -(g^r - g^{r-1}) - \Upsilon^2 w^{r+1} + \Upsilon^2 R(\epsilon^{r+1} - \epsilon^r),\ \forall\ r \geq 0.$$

Note that the matrix $\Upsilon^2 \succ 0$, $\Sigma^2 \succ 0$, then we have

$$\Upsilon^{-1}(\Sigma F)^\top\Sigma^{-1}(\lambda^{r+1} - \lambda^r) = -\Upsilon^{-1}(g^r - g^{r-1}) - \Upsilon w^{r+1} + \Upsilon R(\epsilon^{r+1} - \epsilon^r). \quad (97)$$

Then using the fact that

$$\Sigma^{-1}(\lambda^{r+1} - \lambda^r) = \Sigma F x^{r+1} \in \mathrm{col}(\Sigma F),$$

we can square both sides and obtain the following

$$\begin{aligned}
&\underline{\lambda}_{\min}(\Sigma F\Upsilon^{-2}F^\top\Sigma)\|\Sigma^{-1}(\lambda^{r+1} - \lambda^r)\|^2 \\
&\leq 3\|g^r - g^{r-1}\|^2_{\Upsilon^{-2}} + 3\|w^{r+1}\|^2_{\Upsilon^2} + 3\|\Upsilon R(\epsilon^{r+1} - \epsilon^r)\|^2 \\
&\overset{(18)}{\leq} \frac{3}{M^2}\|\Upsilon^{-1}L(x^r - x^{r-1})\|^2 \\
&+ 3\|w^{r+1}\|^2_{\Upsilon^2} + 3\|\Upsilon R(\epsilon^{r+1} - \epsilon^r)\|^2,\ \forall\ r \geq 0. \quad (98)
\end{aligned}$$

This concludes the proof. **Q.E.D.**

B. Proof of Lemma 5.2

Proof. Consider (50), using the Lipschitz gradient assumption (18), we have

$$\begin{aligned}
&\mathsf{AL}(x^{r+1}, \lambda^r) - \mathsf{AL}(x^r, \lambda^r) \\
&\leq \langle \nabla f(x^r) + F^\top\lambda^r + F^\top\Sigma^2 F x^r, x^{r+1} - x^r\rangle \\
&\quad + \frac{1}{2M}\|x^{r+1} - x^r\|^2_L + \frac{1}{2}\|\Sigma F(x^{r+1} - x^r)\|^2 \\
&= \langle \nabla f(x^r) + F^\top\lambda^r + F^\top\Sigma^2 F x^{r+1}, x^{r+1} - x^r\rangle \\
&+ \langle \Upsilon^2(x^{r+1} - x^r), x^{r+1} - x^r\rangle + \frac{1}{2M}\|x^{r+1} - x^r\|^2_L \\
&+ \frac{1}{2}\|\Sigma F(x^{r+1} - x^r)\|^2 - \|x^{r+1} - x^r\|^2_{\Upsilon^2 + F^\top\Sigma^2 F} \\
&\overset{(95)}{\leq} -(x^{r+1} - x^r)^\top\left(\frac{\Upsilon^2 R}{2} - \frac{L}{2M}\right)(x^{r+1} - x^r) \\
&+ \langle \Upsilon^2 R\epsilon^{r+1}, x^{r+1} - x^r\rangle. \quad (99)
\end{aligned}$$

Using the update rule of the dual variable, and combine the above inequality, we obtain

$$\begin{aligned}
\mathsf{AL}^{r+1} - \mathsf{AL}^r &\leq -\frac{1}{2}\|x^{r+1} - x^r\|^2_{\Upsilon^2 R - \frac{L}{M}} \\
&+ \langle \Upsilon^2 R\epsilon^{r+1}, x^{r+1} - x^r\rangle + \langle \lambda^{r+1} - \lambda^r, F x^{r+1}\rangle \\
&= -\frac{1}{2}\|x^{r+1} - x^r\|^2_{\Upsilon^2 R - \frac{L}{M}} \\
&+ \langle \Upsilon^2 R\epsilon^{r+1}, x^{r+1} - x^r\rangle + \|\Sigma^{-1}(\lambda^{r+1} - \lambda^r)\|^2.
\end{aligned}$$

Combined with Lemma 5.1 we complete the proof. **Q.E.D.**

C. Proof of Lemma 5.4

Proof. Let us choose

$$\eta = \theta^2/(4 + 32M\max\{\lambda_{\max}(\Upsilon^2 R), 1\}). \quad (100)$$

Then from Lemma 5.3, it is clear that if Q satisfies (61), then

$$\|\epsilon^{r+1}\|^2_{\Upsilon^2} \leq \eta\|\tilde{\epsilon}^r\|^2_{\Upsilon^2}. \quad (101)$$

Note that $\Upsilon^2 R = F^\top\Sigma^2 F + \Upsilon^2 \succ 0$, then it follows that

$$\begin{aligned}
&\|\Upsilon^2 R\epsilon^{r+1}\|^2 \leq \frac{\lambda_{\max}(R\Upsilon^2\Upsilon^2 R)}{\lambda_{\min}(\Upsilon^2)}\|\epsilon^{r+1}\|^2_{\Upsilon^2} \\
&\overset{(60)}{\leq} \frac{\eta\lambda_{\max}(R\Upsilon^2\Upsilon^2 R)}{\lambda_{\min}(\Upsilon^2)}\|\tilde{\epsilon}^r\|^2_{\Upsilon^2} \leq \frac{\eta\lambda_{\max}(R\Upsilon^2\Upsilon^2 R)\lambda_{\max}(\Upsilon^2)}{\lambda_{\min}(\Upsilon^2)}\|\tilde{\epsilon}^r\|^2 \\
&\leq \frac{\eta\lambda_{\max}(R\Upsilon^2\Upsilon^2 R)\lambda_{\max}(\Upsilon^2)}{\lambda_{\min}(R\Upsilon^2\Upsilon^2 R)\lambda_{\min}(\Upsilon^2)}\|\Upsilon^2 R\tilde{\epsilon}^r\|^2 \leq \eta\theta^{-2}\|\Upsilon^2 R\tilde{\epsilon}^r\|^2.
\end{aligned}$$

Using the above relation, we can then obtain the following

$$\begin{aligned}
\|\Upsilon^2 R\epsilon^{r+1}\|^2 &\leq 2\eta\theta^{-2}(\|\Upsilon^2 R\epsilon^{r+1}\|^2 + \|\Upsilon^2 R(\epsilon^{r+1} - \tilde{\epsilon}^r)\|^2) \\
&\overset{(59)}{\leq} 2\eta\theta^{-2}(\|\Upsilon^2 R\epsilon^{r+1}\|^2 + \|\Upsilon^2 R(x^{r+1} - x^r)\|^2).
\end{aligned}$$

Therefore, we obtain

$$\|\Upsilon^2 R\epsilon^{r+1}\|^2 \leq 2\eta\theta^{-2}/(1 - 2\eta\theta^{-2})\|\Upsilon^2 R(x^{r+1} - x^r)\|^2.$$

Plugging the definition of η in (100), we have

$$\|\Upsilon^2 R\epsilon^{r+1}\|^2 \le \lambda_{\max}(\Upsilon^2 R)2\eta\theta^{-2}/(1-2\eta\theta^{-2})\|x^{r+1}-x^r\|^2_{\Upsilon^2 R} \overset{(100)}{\le} 1/(16M)\|x^{r+1}-x^r\|^2_{\Upsilon^2 R}, \quad \forall\ r\ge -1.$$

To obtain the second inequality, notice that

$$\|\epsilon^{r+1}\|^2_{\Upsilon^2 R} \le \theta^{-1}\eta\|\widetilde{\epsilon}^r\|^2_{\Upsilon^2 R} \le \theta^{-2}\eta\|\widetilde{\epsilon}^r\|^2_{\Upsilon^2 R} \tag{102}$$

where the last inequality is due to the fact that $\theta \le 1$. Then repeating the above derivation we can obtain the desired result. The third inequality in (62) can be derived in a similar way, and the last two in (62) can be obtained by using Cauchy-Swartz inequality. **Q.E.D.**

D. Proof of Lemma 5.5

Proof. Notice that the following identities hold true

$$\langle a-b, c-d\rangle \le \frac{1}{2}\|a-b\|^2 + \frac{1}{2}\|c-d\|^2 \tag{103}$$

$$\langle a-b, Bb\rangle = \frac{1}{2}\|a\|^2_B - \frac{1}{2}\|b\|^2_B + \frac{1}{2}\|a-b\|^2_B, \text{ with } B\succeq 0. \tag{104}$$

Using the optimality condition from (96) we have

$$\langle F^\top\lambda^{r+1} + \nabla f(x^r) + \Upsilon^2(x^{r+1}-x^r) - \Upsilon^2 R\epsilon^{r+1}, x^{r+1}-x^r\rangle = 0$$
$$\langle F^\top\lambda^{r} + \nabla f(x^{r-1}) + \Upsilon^2(x^{r}-x^{r-1}) - \Upsilon^2 R\epsilon^{r}, x^{r}-x^{r-1}\rangle = 0,$$

Subtract the above two equations, use (103) – (104), and apply the bounds (62d)(62e), we obtain

$$\begin{aligned}&\frac{1}{2}\|\Sigma F x^{r+1}\|^2 + \frac{1}{2}\|x^{r+1}-x^r\|^2_{\Upsilon^2} \\ &\le \frac{1}{2}\|\Sigma F x^{r}\|^2 + \frac{1}{2}\|x^{r}-x^{r-1}\|^2_{\Upsilon^2} - \frac{1}{2}\|w^{r+1}\|^2_{\Upsilon^2} \\ &\quad + 1/(2M)\|x^{r+1}-x^r\|^2_L + 1/(2M)\|x^r-x^{r-1}\|^2_L \\ &\quad + 1/4\|x^{r+1}-x^r\|^2_{\Upsilon^2 R} + 1/4\|x^r-x^{r-1}\|^2_{\Upsilon^2 R}, \quad \forall\ r\ge 0.\end{aligned} \tag{105}$$

By using the potential function defined in (63), we have

$$\begin{aligned}\widetilde{P}^{r+1} - \widetilde{P}^r &= \mathsf{AL}^{r+1} - \mathsf{AL}^r + \frac{3\widetilde{\kappa}}{M^2}\|\Upsilon^{-1}L(x^{r+1}-x^r)\|^2 \\ &\quad - \frac{3\widetilde{\kappa}}{M^2}\|\Upsilon^{-1}L(x^r-x^{r-1})\|^2 \\ &\quad + \frac{3\widetilde{\kappa}}{8}(\|x^{r+1}-x^r\|^2_{\Upsilon^2 R} - \|x^r-x^{r-1}\|^2_{\Upsilon^2 R}) \\ &\quad + \frac{\widetilde{c}}{2}\left(\|\Sigma F x^{r+1}\|^2 + \|x^{r+1}-x^r\|^2_{\Upsilon^2+\frac{\Upsilon^2 R}{4}+\frac{L}{M}}\right) \\ &\quad - \frac{\widetilde{c}}{2}\left(\|\Sigma F x^{r}\|^2 + \|x^{r}-x^{r-1}\|^2_{\Upsilon^2+\frac{\Upsilon^2 R}{4}+\frac{L}{M}}\right).\end{aligned}$$

Multiplying (105) with $\widetilde{c}$, then adding to (54), and use the estimate of the size of ϵ in (62c) and (62d), we can obtain

$$\widetilde{P}^{r+1} - \widetilde{P}^r \le -\frac{1}{2}(x^{r+1}-x^r)^\top V(x^{r+1}-x^r) - \left(\frac{\widetilde{c}}{2} - 3\widetilde{\kappa}\right)\|w^{r+1}\|^2,$$

with the matrix V defined as follows

$$V := \left(\Upsilon^2 R - (1+2\tilde{c})\frac{L}{M} - \frac{6\tilde{\kappa}}{M^2}L\Upsilon^{-2}L - \frac{\Upsilon^2 R(24\tilde{\kappa}+6+16\tilde{c})}{16}\right).$$

Therefore in order to make the potential function decrease, we need to follow (64). **Q.E.D.**

E. Proof of Lemma 5.6

Proof. We can express the AL as (for all $r\ge 0$)

$$\mathsf{AL}^{r+1} - f(x^{r+1}) = \langle\lambda^{r+1}, \Sigma^{-2}(\lambda^{r+1}-\lambda^r)\rangle + \frac{1}{2}\|\Sigma F x^{r+1}\|^2$$
$$= \frac{1}{2}\big(\|\Sigma^{-1}\lambda^{r+1}\|^2 - \|\Sigma^{-1}\lambda^r\|^2 + \|\Sigma^{-1}(\lambda^{r+1}-\lambda^r)\|^2 + \|\Sigma F x^{r+1}\|^2\big).$$

Since $\inf_x f(x) = \underline{f}$ is lower bounded, let us define

$$\widehat{\mathsf{AL}}^{r+1} := \mathsf{AL}^{r+1} - \underline{f},\ \widehat{f}(x) := f(x) - \underline{f} \ge 0,\ \widehat{P}^{r+1} := \widetilde{P}^{r+1} - \underline{f}.$$

Therefore, summing over $r = -1\cdots, T$, we obtain

$$\sum_{r=-1}^{T}\widehat{\mathsf{AL}}^{r+1} = \frac{1}{2}\left(\|\Sigma^{-1}\lambda^{T+1}\|^2 - \|\Sigma^{-1}\lambda^{-1}\|^2\right) + \sum_{r=-1}^{T}\left(\widehat{f}(x^{r+1}) + \frac{1}{2}\|\Sigma F x^{r+1}\|^2 + \frac{1}{2}\|\Sigma^{-1}(\lambda^{r+1}-\lambda^r)\|^2\right).$$

Using the initialization $\lambda^{-1}=0$, then the above sum is lower bounded by zero. This fact implies that the sum of $\widehat{P}^{r+1}$ is also lower bounded by zero (since besides $\widehat{\mathsf{AL}}$, the remaining terms in $\widehat{P}$ are all nonnegative)

$$\sum_{r=0}^{T}\widehat{P}^{r+1} \ge 0, \quad \forall\ T>0,$$

Note that if the parameters of the system are chosen according to (64), then $\widetilde{P}^{r+1}$ is nonincreasing, which implies that its shifted version $\widehat{P}^{r+1}$ is also nonincreasing. Combined with the nonnegativity of the sum of the shifted potential function, we can conclude that

$$\widehat{P}^{r+1} \ge 0, \quad \text{and} \quad \widetilde{P}^{r+1} \ge \inf f(x), \quad \forall\ r\ge 0. \tag{106}$$

Next we compute $\widetilde{P}^0$. By letting $r=-1$, and use $x^{-1}=0$ and $\lambda^{-1}=0$, we obtain

$$\mathsf{AL}^0 - f(x^0) = \frac{1}{2}\big(2\|\Sigma^{-1}\lambda^0\|^2 + \|\Sigma F x^0\|^2\big) = \frac{3}{2}\|\Sigma^{-1}\lambda^0\|^2. \tag{107}$$

Then we have

$$\widetilde{P}^0 = \mathsf{AL}^0 + \frac{3\widetilde{\kappa}}{M^2}\|\Upsilon^{-1}Lx^0\|^2 + \frac{3}{8}\tilde{\kappa}\|x^0\|^2_{\Upsilon^2 R} + \frac{\widetilde{c}}{2}\left(\|\Sigma F x^0\|^2 + \|x^0\|^2_{\Upsilon^2+\Upsilon^2 R/4+L/M}\right), \tag{108a}$$

$$\mathsf{AL}^0 \le f(x^0) + 2\|\Sigma F x^0\|^2,\ x^{-1}=0,\ \lambda^{-1}=0, \tag{108b}$$

$$x^0 \overset{(45b)}{=} R^{-1}\Upsilon^{-2}\nabla f(0) - \epsilon^0, \quad \widetilde{\epsilon}^{-1} \overset{(58)}{=} R^{-1}\Upsilon^{-2}\nabla f(0). \tag{108c}$$

Use the above relation, we have

$$\widetilde{P}^0 \le f(x^0) + (x^0)^\top\widetilde{Z}x^0, \text{ with } Z \text{ defined as}$$

$$\widetilde{Z} = \frac{3\tilde{\kappa}}{M^2}L\Upsilon^{-2}L + \left(\frac{3}{8}\tilde{\kappa}+\tilde{c}\right)\Upsilon^2 R + \frac{\tilde{c}L}{2M} + 2F\Sigma^2 F \preceq 3\Upsilon^2 R$$

where the last inequality follows from our choice of parameters in (64b). Therefore we have

$$\begin{aligned}(x^0)^\top\widetilde{Z}x^0 &\le 3(x^0)^\top\Upsilon^2 R x^0 \\ &\le 3(\nabla f(0) - \Upsilon^2 R\epsilon^0)^\top R^{-1}\Upsilon^{-2}(\nabla f(0) - \Upsilon^2 R\epsilon^0) \\ &\overset{(i)}{\le} 6(\nabla f(0))^\top R^{-1}\Upsilon^{-2}\nabla f(0) + 6(\epsilon^0)^\top\Upsilon^2 R\epsilon^0 \\ &\le 3M(\nabla f(0))^\top L^{-1}\nabla f(0) + \frac{3}{8M}\|x^0\|^2_{\Upsilon^2 R}\end{aligned}$$

where in (i) we have used the Cauchy-Swartz inequality; the last inequality uses (62), the choice of the parameters (64b) (which implies $\Upsilon^2 R \geq 4L/M$). The above series of inequalities imply that

$$2\|x^0\|^2_{\Upsilon^2 R} \leq \left(3 - \frac{3}{8M}\right)\|x^0\|^2_{\Upsilon^2 R} \leq 3M(\nabla f(0))^\top L^{-1}\nabla f(0).$$

Therefore overall we have

$$(x^0)^\top \widetilde{Z} x^0 \leq 3(x^0)^\top \Upsilon^2 R x^0 \leq 5M(\nabla f(0))^\top L^{-1}\nabla f(0).$$

By observing $\frac{1}{M^2} d_0^\top d_0 = \|\nabla f(0)\|^2$, the desired result is obtained. **Q.E.D.**

F. Proof of Theorem 5.1

To show the result, we consider the optimality condition (95), and multiply both sides of it by the all one vector, and use the fact that $F\mathbb{1} = 0$ to obtain

$$\mathbb{1}^\top \nabla f(x^r) + \mathbb{1}^\top \Upsilon^2 (x^{r+1} - x^r) = \mathbb{1}^\top \Upsilon^2 R \epsilon^{r+1}.$$

Squaring both sides and rearranging terms we have

$$\begin{aligned}
\left\| \frac{1}{M}\sum_{i=1}^M \nabla f_i(x_i^r)\right\|^2 &\leq 2(x^{r+1} - x^r)^\top \Upsilon^2 \mathbb{1}\mathbb{1}^\top \Upsilon^2 (x^{r+1} - x^r) \\
&\quad + 2(\epsilon^{r+1})^\top \Upsilon^2 R \mathbb{1}\mathbb{1}^\top \Upsilon^2 R \epsilon^{r+1} \\
&\overset{(62)}{\leq} 2(x^{r+1} - x^r)^\top \Upsilon^2 (x^{r+1} - x^r) \times \mathbb{1}^\top \Upsilon^2 \mathbb{1} \\
&\quad + M/(4M)\|x^{r+1} - x^r\|^2_{\Upsilon^2 R} \\
&\overset{(i)}{\leq} \|x^{r+1} - x^r\|^2_{\Upsilon^2 R} \times 2\left(1 + \sum_{i=1}^M \beta_i^2\right), \\
&\overset{(65)}{\leq} 64(\widetilde{P}^r - \widetilde{P}^{r+1}) \times 2\left(1 + \sum_{i=1}^M \beta_i^2\right), \ \forall\ r \geq 0.
\end{aligned}$$

where in (i) we used $\Upsilon^2 R = \Upsilon^2 + F^\top \Sigma^2 F \succeq \Upsilon^2$.

To bound the consensus error, we first use (62) and obtain

$$\|\Upsilon^2 R(\epsilon^{r+1} - \epsilon^r)\|^2 \leq \frac{1}{4M}\|x^{r+1} - x^r\|^2_{\Upsilon^2 R} + \frac{1}{4M}\|x^r - x^{r-1}\|^2_{\Upsilon^2 R}.$$

Then we apply Lemma 5.1 to obtain

$$\begin{aligned}
&\|\Sigma F x^{r+1}\|^2 \qquad (109)\\
&\leq 3\widetilde{\kappa}\left(\|x^{r+1} - x^r\|^2_{\frac{\Upsilon^2 R}{4M}} + \|w^{r+1}\|^2_{\Upsilon^2} + \|x^r - x^{r-1}\|^2_{\frac{\Upsilon^2 R}{4M} + \frac{L\Upsilon^{-2}L}{M^2}}\right) \\
&\overset{(i)}{\leq} 2\|x^{r+1} - x^r\|^2_{\Upsilon^2 R} + 3\widetilde{\kappa}\|w^{r+1}\|^2_{\Upsilon^2} + 2\|x^r - x^{r-1}\|^2_{\Upsilon^2 R}, \ \forall\ r \geq 0
\end{aligned}$$

where (i) is a consequence of (64b), which implies

$$2\Upsilon^2 R \succeq 3\widetilde{\kappa}\left(\frac{L\Upsilon^{-2}L}{M^2} + \Upsilon^2 R\right). \qquad (110)$$

By combining (109) and the following inequality

$$\|\Sigma F x^r\|^2 \leq 2\|\Sigma F(x^{r+1} - x^r)\|^2 + 2\|\Sigma F x^{r+1}\|^2,$$

we have

$$\begin{aligned}
\|\Sigma F x^r\|^2 &\leq 4\|x^{r+1} - x^r\|^2_{\Upsilon^2 R + F^\top \Sigma^2 F} + 6\tilde{\kappa}\|w^{r+1}\|^2_{\Upsilon^2} \\
&\quad + 4\|x^r - x^{r-1}\|^2_{\Upsilon^2 R} \\
&\overset{(65)}{\leq} 64(\widetilde{P}^r - \widetilde{P}^{r+1}) + 64(\widetilde{P}^{r-1} - \widetilde{P}^r), \quad \forall\ r \geq 1 \\
\|\Sigma F x^0\|^2 &\leq 64(\widetilde{P}^0 - \widetilde{P}^1) + 4\|x^0\|^2_{\Upsilon^2 R}.
\end{aligned}$$

So overall we have that

$$\begin{aligned}
&\sum_{r=0}^{T_r}\left(\left\|\frac{1}{M}\sum_{i=1}^M \nabla f_i(x_i^r)\right\|^2 + \|\Sigma F x^r\|^2\right) \\
&\leq 64\left(1 + \sum_{i=1}^M \beta_i^2 + 1\right)\sum_{r=1}^{T_r}((\widetilde{P}^r - \widetilde{P}^{r+1}) + (\widetilde{P}^{r-1} - \widetilde{P}^r)) \\
&\quad + 64(\widetilde{P}^0 - \widetilde{P}^1) + 4\|x^0\|^2_{\Upsilon^2 R} \\
&\leq 128\left(1 + \sum_{i=1}^M \beta_i^2 + 2\right)(\widetilde{P}^0 - \underline{f}) + 4\|x^0\|^2_{\Upsilon^2 R}. \qquad (111)
\end{aligned}$$

where the last inequality utilizes the descent property of $\widetilde{P}^r$ in Lemma 5.5, and the boundedness property in Lemma 5.6. Note that from (107), (108a) and use $\widetilde{c} = 8\widetilde{\kappa}$ in (64a), we obtain

$$\widetilde{P}^0 \geq f(x^0) + \widetilde{\kappa}\|x^0\|^2_{\Upsilon^2 R}. \qquad (112)$$

Therefore From (66) and Lemma 5.6 we have that

$$\begin{aligned}
4\|x^0\|^2_{\Upsilon^2 R} &\leq \frac{4\left(\widetilde{P}^0 - f(x^0)\right)}{\widetilde{\kappa}} \\
&\overset{(66)}{\leq} \frac{4\left(f(x^0) + \frac{5}{M} d_0^\top L^{-1} d_0 - \underline{f}\right)}{\widetilde{\kappa}} := \frac{4\widetilde{C}_1}{\widetilde{\kappa}}.
\end{aligned}$$

Combining the above two relations leads to

$$\begin{aligned}
&\frac{1}{T_r}\sum_{r=0}^{T_r}\left(\|\frac{\sum_{i=1}^M \nabla f_i(x_i^r)}{M}\|^2 + \|\Sigma F x^r\|^2\right) \\
&\leq 128\left(\sum_{i=1}^M \beta_i^2 + 3 + \frac{1}{32\widetilde{\kappa}}\right)\widetilde{C}_1/T_r.
\end{aligned}$$

This completes the proof. **Q.E.D.**

REFERENCES

[1] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms," in *Proceedings of the 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018.

[2] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2017.

[3] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 707–724, Aug 2011.

[4] T.-H. C. H.-T. Wai and A. Scaglione, "A consensus-based decentralized algorithm for non-convex optimization with application to dictionary learning," in *the Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.

[5] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[6] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[7] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2014.

[8] D. Jakovetić, J. M. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented lagrangian algorithms," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 922–936, 2015.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[10] I. Schizas, G. Mateos, and G. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing,," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365 – 2382, 2009.

[11] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, 2013.

[12] M. Zhu and S. Martínez, "An approximate dual subgradient algorithm for distributed non-convex constrained optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1534–1539, June 2013.

[13] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal On Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[14] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[15] D. Hajinezhad and M. Hong, "Perturbed proximal primal dual algorithm for nonconvex nonsmooth optimization," *Mathematical Programming*, vol. 176, no. 1-2, pp. 207–245, July 2019.

[16] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *the Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[17] D. Hajinezhad, M. Hong, and A. Garcia, "Zone: Zeroth order nonconvex multi-agent optimization over networks," *IEEE Transactions on Automatic Control*, 2019.

[18] A. Daneshmand, G. Scutari, and F. Facchinei, "Distributed dictionary learning," in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*, Nov. 6–9, 2016.

[19] A. Daneshmand, Y. Sun, G. Scutari, and F. Facchinei, "Distributed dictionary learning over networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, March 5-9 2017.

[20] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2834–2848, June 2018.

[21] Z. Jiang, A. Balu, C. Hegde, and S. Sarkar, "Collaborative deep learning in fixed topology networks," in *Advances in Neural Information Processing Systems*, 2017.

[22] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments–part i: Agreement at a linear rate," *arXiv preprint arXiv:1907.01848*, 2019.

[23] ——, "Distributed learning in non-convex environments–part ii: Polynomial escape from saddle-points," *arXiv preprint arXiv:1907.01849*, 2019.

[24] B. Swenson, S. Kar, H. V. Poor, and J. Moura, "Annealing for distributed global optimization," *arXiv preprint arXiv:1903.07258*, 2019.

[25] M. Hong, J. D. Lee, and M. Razaviyayn, "Gradient primal-dual algorithm converges to second-order stationary solutions for nonconvex distributed optimization," in *the Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[26] A. Daneshmand, G. Scutari, and V. Kungurtsev, "Second-order guarantees of gradient algorithms over networks," in *Proceedings of the 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018.

[27] B. Swenson, R. Murray, H. V. Poor, and S. Kar, "Distributed gradient descent: Nonconvergence to saddle points and the stable-manifold theorem," *arXiv preprint arXiv:1908.02747*, 2019.

[28] C. Duenner, A. Lucchi, M. Gargiani, A. Bian, T. Hofmann, and M. Jaggi, "A distributed second-order algorithm you can trust," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[29] C.-H. Fang, S. B. Kylasa, F. Roosta-Khorasani, M. W. Mahoney, and A. Grama, "Distributed second-order convex optimization," *arXiv preprint arXiv:1807.07132*, 2018.

[30] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.

[31] A. Nemirovsky and D. Yudin, "Problem complexity and method efficiency in optimization," in *Interscience Series in Discrete Mathematics*. Wiley, 1983.

[32] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer, 2004.

[33] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imgaging Science*, vol. 2, no. 1, pp. 183 – 202, 2009.

[34] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," 2008, preprint.

[35] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[36] K. Scaman, F. Bach, S. Bubeck, Y. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," *arXiv preprint arXiv:1702.08704*, 2017.

[37] C. Uribe, S. Lee, A. Gasnikov, and A. Nedić, "Optimal algorithms for distributed optimization," *arXiv preprint arXiv:1712.00232*, 2017.

[38] K. Scaman, F. Bach, S. Bubeck, L. Massoulié, and Y. T. Lee, "Optimal algorithms for non-smooth distributed optimization in networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 2740–2749.

[39] C. Cartis, N. Gould, and P. Toint, "On the complexity of steepest descent, newton's and regularized newton's methods for nonconvex unconstrained optimization problems," *SIAM journal on optimization*, vol. 20, no. 6, pp. 2833–2852, 2010.

[40] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, "Lower bounds for finding stationary points i," *Mathematical Programming*, Jun 2019.

[41] A. Daneshmand, Y. Sun, and G. Scutari, "Convergence rate of distributed convex and nonconvex optimization methods with gradient tracking," 2018, purdue University, Tech. Rep.

[42] F. R. K. Chung, *Spectral Graph Theory*. The American Mathematical Society, 1997.

[43] S. Butler, *Algebraic aspects of the normalized Laplacian*. Cham: Springer International Publishing, 2016, pp. 295–315.

[44] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, March 2012.

[45] Y. Nesterov, "Smooth minimization of nonsmooth functions," *Mathematical Programming*, vol. 103, pp. 127–152, 2005.

[46] D. Tian, H. Mansour, A. Knyazev, and A. Vetro, "Chebyshev and conjugate gradient filters for graph image denoising," in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2014.

[47] A. Gadde, S. K. Narang, and A. Ortega, "Bilateral filter: Graph spectral interpretation and extensions," in *Proceedings of the IEEE International Conference on Image Processing*.

[48] V. S. Ryaben'kii and S. V. Tsynkov, *A Theoretical Introduction to Numerical Analysis*. CRC Press, 2007.

[49] A. A. Samarskij and E. S. Nikolaev, *Numerical Methods for Grid Equations Volume II Iterative Methods*. Springer, 1989.

[50] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

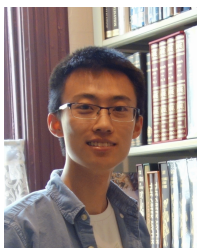
[51] H. Sun and M. Hong, "Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms (online version)," *IEEE Transactions on Signal processing*, July 2019, accepted for publication.

[52] A. Antoniadis, I. Gijbels, and M. Nikolova, "Penalized likelihood regression for generalized linear models with non-quadratic penalties," *Annals of the Institute of Statistical Mathematics*, vol. 63, no. 3, pp. 585–615, 2011.

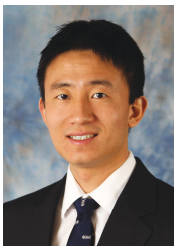
[53] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2506–2517, 2009.

[54] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, 2017.

[55] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, Jan 2015.



Haoran Sun received the B.S. degree in Automatic Control from Beijing Institute of Technology, Beijing, China, in 2015, and the M.S. degree in Industrial Engineering from Iowa State University, Ames, IA, USA, in 2017. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. His research interests include optimization, machine learning, and its applications in signal processing and wireless communications. He won the third place in the Student Paper Contest at the 52nd Asilomar Conference on Signals, Systems, and Computers.



Mingyi Hong received his Ph.D. degree from University of Virginia in 2011. Since August 2017, he has been an Assistant Professor in the Department of Electrical and Computer Engineering, University of Minnesota. From 2014-2017 he has been Assistant Professor with the Department of Industrial and Manufacturing Systems Engineering, Iowa State University. He is serving on the IEEE Signal Processing for Communications and Networking (SPCOM), and Machine Learning for Signal Processing (MLSP) Technical Committees. His research interests are primarily in the fields of optimization theory and applications in signal processing and machine learning.

X. Supplemental Material: The Complexity Analysis

A. Proof of Lemma 3.1

Proof. Property 1) is obviously true.

To prove Property 2), note that following holds for $w > 0$:

$$\Psi(w) = 1 - e^{-w^2}, \quad \Psi'(w) = 2e^{-w^2}w, \quad \Psi''(w) = 2e^{-w^2} - 4e^{-w^2}w^2, \ \forall\ w > 0. \tag{113}$$

Obviously, $\Psi(w)$ is an increasing function over $w > 0$, therefore the lower and upper bounds are $\Psi(0) = 0, \Psi(\infty) = 1$; $\Psi'(w)$ is increasing on $[0, \frac{1}{\sqrt{2}}]$ and decreasing on $[\frac{1}{\sqrt{2}}, \infty]$, where $\Psi''(\frac{1}{\sqrt{2}}) = 0$, therefore the lower and upper bounds are $\Psi'(0) = \Psi'(\infty) = 0, \Psi'(\frac{1}{\sqrt{2}}) = \sqrt{\frac{2}{e}}$; $\Psi''(w)$ is decreasing on $(0, \sqrt{\frac{3}{2}}]$ and increasing on $[\sqrt{\frac{3}{2}}, \infty)$ [this can be verified by checking the signs of $\Psi'''(w) = 4e^{-w^2}w(2w^2 - 3)$ in these intervals]. Therefore the lower and upper bounds are $\Psi''(\sqrt{\frac{3}{2}}) = -\frac{4}{e^{\frac{3}{2}}}, \Psi''(0^+) = 2$, i.e.,

$$0 \le \Psi(w) < 1, \quad 0 \le \Psi'(w) \le \sqrt{\frac{2}{e}}, \quad -\frac{4}{e^{\frac{3}{2}}} \le \Psi''(w) \le 2, \quad \forall w > 0.$$

Further, for all $w \in \mathbb{R}$, the following holds:

$$\Phi(w) = 4\arctan w + 2\pi, \quad \Phi'(w) = \frac{4}{w^2 + 1}, \quad \Phi''(w) = -\frac{8w}{(w^2 + 1)^2}. \tag{114}$$

Similarly, as above, we can obtain the following bounds:

$$0 < \Phi(w) < 4\pi, \quad 0 < \Phi'(w) \le 4, \quad -\frac{3\sqrt{3}}{2} \le \Phi''(w) \le \frac{3\sqrt{3}}{2}, \quad \forall w \in \mathbb{R}.$$

We refer the readers to Fig. 1 for illustrations of these functions.

To show Property 3), note that for all $w \ge 1$ and $|v| < 1$,

$$\Psi(w)\Phi'(v) > \Psi(1)\Phi'(1) = 2(1 - e^{-1}) > 1$$

where the first inequality is true because $\Psi(w)$ is strictly increasing and $\Phi'(v)$ is strictly decreasing for all $w > 0$, and that $\Phi'(v) = \Phi'(|v|)$.

Next we show Property 4). Note that $0 \le \Psi(w) < 1$ and $0 < \Phi(w) < 4\pi$. Therefore we have $h(0) = -\Psi(1)\Phi(0) < 0$ and using the construction in (28)

$$\inf_{x_i} h_i(x_i) \ge -\Psi(1)\Phi(x_i[1]) - 3\sum_{j=1}^{\lfloor T/2 \rfloor} \Psi(w)\Phi(v) \ge -4\pi - 6\pi T \ge -10\pi T \tag{115}$$

where the first inequality follows $\Psi(w)\Phi(v) > 0$ and second follows $\Psi(w)\Phi(v) < 4\pi$, we reach the conclusion.

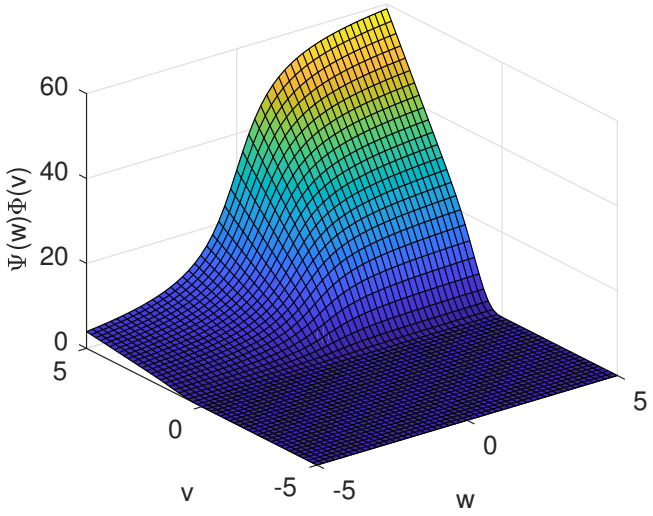


Fig. 8: The functional value for $\Theta(w, v) = \Psi(w)\Phi(v)$.

Finally we show Property 5), using the fact that a function is Lipschitz if it is piecewise smooth with bounded derivative. From construction (28), the first-order partial derivative of $h_q(y)$ can be expressed below.

Case I) If i is even, we have

$$\frac{\partial h_q}{\partial y[i]} = \begin{cases} 3\left(-\Psi\left(-y[i-1]\right)\Phi'\left(-y[i]\right) - \Psi\left(y[i-1]\right)\Phi'\left(y[i]\right)\right), & q \in [1, \frac{M}{3}] \\ 0, & q \in [\frac{M}{3}+1, \frac{2M}{3}] \\ 3\left(-\Psi'\left(-y[i]\right)\Phi\left(-y[i+1]\right) - \Psi'\left(y[i]\right)\Phi\left(y[i+1]\right)\right), & q \in [\frac{2M}{3}+1, M] \end{cases}. \tag{116}$$

Case II) If i is odd but not 1, we have

$$\frac{\partial h_q}{\partial y[i]} = \begin{cases} 3\left(-\Psi'\left(-y[i]\right)\Phi\left(-y[i+1]\right) - \Psi'\left(y[i]\right)\Phi\left(y[i+1]\right)\right), & q \in [1, \frac{M}{3}] \\ 0, & q \in [\frac{M}{3}+1, \frac{2M}{3}] \\ 3\left(-\Psi\left(-y[i-1]\right)\Phi'\left(-y[i]\right) - \Psi\left(y[i-1]\right)\Phi'\left(y[i]\right)\right), & q \in [\frac{2M}{3}+1, M] \end{cases}. \tag{117}$$

Case III) If $i = 1$, we have

$$\frac{\partial h_q}{\partial y[1]} = \begin{cases} -\Psi(1)\Phi'(y[1]) + 3\left(-\Psi'\left(-y[1]\right)\Phi\left(-y[2]\right) - \Psi'\left(y[1]\right)\Phi\left(y[2]\right)\right), & q \in [1, \frac{M}{3}] \\ -\Psi(1)\Phi'(y[1]), & q \in [\frac{M}{3}+1, M] \end{cases}. \tag{118}$$

Obviously, $\frac{\partial h_q}{\partial y[i]}$ is a piecewise smooth function for any i, q, and it either equals zero or is separated at the non-differentiable point $y[i] = 0$ because of the function Ψ.

Further, fix a point $y \in \mathbb{R}^T$ and a unit vector $v \in \mathbb{R}^T$ where $\sum_{i=1}^T v[i]^2 = 1$. Define

$$g_q(\theta; y, v) := h_q(y + \theta v)$$

to be the directional projection of h_q on to the direction v at point y. We will show that there exists $\ell > 0$ such that $|{g_q}''(0; y, v)| \le \ell$ for all $y \neq 0$ (where the second-order derivative is taken with respect to θ).

First we can compute ${g_q}''(0; y, v)$ as follows:

$$g_q^{''}(0; y, v) = \sum_{i_1, i_2 = 1}^{T} \frac{\partial^2}{\partial y[i_1]\partial y[i_2]} h_q(y)\, v[i_1] v[i_2] = \sum_{\delta \in \{0,1,-1\}} \sum_{i=1}^{T} \frac{\partial^2}{\partial y[i]\partial y[i+\delta]} h_q(y)\, v[i] v[i+\delta],$$

where we take $v[0] := 0$ and $v[T+1] := 0$.

The second-order partial derivative of $h_q(y)$ ($\forall y \neq 0$) is given as follows when i is even:

$$\frac{\partial^2 h_q}{\partial y[i]\partial y[i]} = \begin{cases} 3\left(\Psi\left(-y[i-1]\right)\Phi''\left(-y[i]\right) - \Psi\left(y[i-1]\right)\Phi''\left(y[i]\right)\right), & q \in [1, \frac{M}{3}] \\ 0, & q \in [\frac{M}{3}+1, \frac{2M}{3}] \\ 3\left(\Psi''\left(-y[i]\right)\Phi\left(-y[i+1]\right) - \Psi''\left(y[i]\right)\Phi\left(y[i+1]\right)\right), & q \in [\frac{2M}{3}+1, M] \end{cases} \tag{119}$$

$$\frac{\partial^2 h_q}{\partial y[i]\partial y[i+1]} = \begin{cases} 0, & q \in [1, \frac{2M}{3}] \\ 3\left(\Psi'\left(-y[i]\right)\Phi'\left(-y[i+1]\right) - \Psi'\left(y[i]\right)\Phi'\left(y[i+1]\right)\right), & q \in [\frac{2M}{3}+1, M] \end{cases} \tag{120}$$

$$\frac{\partial^2 h_q}{\partial y[i]\partial y[i-1]} = \begin{cases} 3\left(\Psi'\left(-y[i-1]\right)\Phi'\left(-y[i]\right) - \Psi'\left(y[i-1]\right)\Phi'\left(y[i]\right)\right), & q \in [1, \frac{M}{3}] \\ 0, & q \in [\frac{M}{3}+1, M] \end{cases}. \tag{121}$$

By applying Lemma 3.1 – i) [i.e., $\Psi(w) = \Psi'(w) = \Psi''(w) = 0$ for $\forall\, w \le 0$], we immediately obtain that at least one of the terms $\Psi\left(-y[i-1]\right)\Phi''\left(-y[i]\right)$ or $-\Psi\left(y[i-1]\right)\Phi''\left(y[i]\right)$ is zero. It follows that

$$\Psi\left(-y[i-1]\right)\Phi''\left(-y[i]\right) - \Psi\left(y[i-1]\right)\Phi''\left(y[i]\right) \le \sup_w |\Psi(w)| \sup_v |\Phi''(v)|.$$

Similarly,

$$\Psi''\left(-y[i]\right)\Phi\left(-y[i+1]\right) - \Psi''\left(y[i]\right)\Phi\left(y[i+1]\right) \le \sup_w |\Psi''(w)| \sup_v |\Phi(v)|$$

$$\Psi'\left(-y[i]\right)\Phi'\left(-y[i+1]\right) - \Psi'\left(y[i]\right)\Phi'\left(y[i+1]\right) \le \sup_w |\Psi'(w)| \sup_v |\Phi'(v)|.$$

Therefore, take the maximum over equations (119) to (121) and plug in the above inequalities, we obtain

$$\begin{aligned} \left|\frac{\partial^2 h_q}{\partial y[i_1]\partial y[i_2]}\right| &\le 3\max\{\sup_w |\Psi''(w)| \sup_v |\Phi(v)|, \sup_w |\Psi(w)| \sup_v |\Phi''(v)|, \sup_w |\Psi'(w)| \sup_v |\Phi'(v)|\} \\ &= 3\max\left\{8\pi, \frac{3\sqrt{3}}{2}, 4\sqrt{\frac{2}{e}}\right\} < 25\pi, \quad \forall\ i_1 \text{ being even}, \ \forall\ i_2 \end{aligned}$$

where the equality comes from Lemma 3.1 – ii).

We can also verify that the above bound for i being odd but not 1 is exactly the same.

When $i = 1$ we have following:

$$\frac{\partial^2 h_q}{\partial y[1]\partial y[1]} = \begin{cases} -\Psi(1)\Phi''(y[1]) + 3\left(-\Psi''\left(-y[1]\right)\Phi\left(-y[2]\right) - \Psi''\left(y[1]\right)\Phi\left(y[2]\right)\right), & q \in [1, \frac{M}{3}] \\ -\Psi(1)\Phi''(y[1]), & q \in [\frac{M}{3}+1, M] \end{cases}$$

$$\frac{\partial^2 h_q}{\partial y[1]\partial y[2]} = \begin{cases} 3\left(-\Psi'\left(-y[1]\right)\Phi'\left(-y[2]\right) - \Psi'\left(y[1]\right)\Phi'\left(y[2]\right)\right), & q \in [1, \frac{M}{3}] \\ 0, & q \in [\frac{M}{3}+1, M] \end{cases}$$

Again by applying Lemma 3.1 – i) and ii),

$$\begin{aligned} \left|\frac{\partial^2 h_q}{\partial y[1]\partial y[i_2]}\right| &\leq \max\{\sup_w |\Psi(1)\Phi''(w)| + 3\sup_w |\Psi''(w)| \sup_v |\Phi(v)|, 3\sup_w |\Psi'(w)| \sup_v |\Phi'(v)|\} \\ &= \max\left\{\frac{3\sqrt{3}}{2}(1-e^{-1}) + 24\pi, 12\sqrt{\frac{2}{e}}\right\} < 25\pi, \ \forall\ i_2. \end{aligned}$$

Summarizing the above results, we obtain:

$$\begin{aligned} \left|g_q''(0;y,v)\right| &= \left|\sum_{\delta\in\{0,1,-1\}}\sum_{i=1}^{T} \frac{\partial^2}{\partial y[i]\partial y[i+\delta]} h_q(y)\, v[i]v[i+\delta]\right| \\ &\leq 25\pi \sum_{\delta\in\{0,1,-1\}} \left|\sum_{i=1}^{T} v[i]v[i+\delta]\right| \\ &= 25\pi\left(\left|\sum_{i=1}^{T} v[i]^2\right| + 2\left|\sum_{i=1}^{T} v[i]v[i+1]\right|\right) \\ &\leq 75\pi \sum_{i=1}^{T} \left|v[i]^2\right| = 75\pi. \end{aligned}$$

Overall, the first-order derivatives of h_q are Lipschitz continuous for any q with constant $\ell = 75\pi$.

To show the same result for the function $\bar{h}$, we can apply (17). This completes the proof. **Q.E.D.**

B. Proof of Lemma 3.2

Proof. To show that property 1) is true, note that from the definition of $f_i(x_i)$ we have

$$\nabla f_i(x_i) = \sqrt{2\epsilon} \times \nabla h_i\left(\frac{x_i U}{75\pi\sqrt{2\epsilon}}\right).$$

Therefore the following holds:

$$\frac{1}{M}\|d_0\|^2 = \frac{2\epsilon}{M}\sum_{i=1}^{M}\|\nabla h_i(0)\|^2 = \frac{2\epsilon}{M}\sum_{i=1}^{M}|\Psi(1)\Phi'(0)|^2 = 32\epsilon(1-\exp(-1))^2. \tag{122}$$

Therefore we have the following:

$$f(0) - \inf_x f(x) + \frac{\|d_0\|^2}{MU} = \frac{150\pi\epsilon}{U}\left(h(0) - \inf_x h(x) + \frac{16(1-\exp(-1))^2}{75\pi}\right).$$

Then by applying Lemma 3.1 we have that for any $T \geq 1$, the following holds

$$f(0) - \inf_x f(x) + \frac{\|d_0\|^2}{MU} \leq \frac{150\pi\epsilon}{U} \times (10\pi T + 0.03) \leq \frac{150\pi\epsilon}{U} \times 11\pi T.$$

Property 2) is true due to the definition of $\bar{f}$.

Property 3) is true because the following

$$\|\nabla\bar{f}(z) - \nabla\bar{f}(y)\| = \sqrt{2\epsilon}\left\|\nabla\bar{h}\left(\frac{zU}{75\pi\sqrt{2\epsilon}}\right) - \nabla\bar{h}\left(\frac{yU}{75\pi\sqrt{2\epsilon}}\right)\right\| \leq U\|z-y\|$$

where the last inequality comes from Lemma 3.1 – (5). This completes the proof. **Q.E.D.**

C. Proof of Lemma 3.3

Proof. The first inequality holds for all $k \in [T]$, since $\frac{1}{M}\sum_{i=1}^{M}\frac{\partial}{\partial y[k]}h_i(y)$ is one element of $\frac{1}{M}\sum_{i=1}^{M}\nabla h_i(y)$.

We divide the proof for second inequality into two cases.

Case 1. Suppose $|y[j-1]| < 1$ for all $2 \le j \le k$. Therefore, we have $|y[1]| < 1$. Using (118), we have the following inequalities:

$$\frac{\partial}{\partial y[1]} h_i(y) \overset{\text{(i)}}{\le} -\Psi(1)\Phi'(y[1]) \overset{\text{(ii)}}{<} -1, \forall i \tag{123}$$

where (i) is true because $\Psi'(w), \Phi(w)$ are all non-negative from Lemma 3.1 -(2); (ii) is true due to Lemma 3.1 – (3). Therefore, we have the following

$$\left\|\nabla \bar{h}(y)\right\| = \left\| \frac{1}{M} \sum_{i=1}^{M} \nabla h_i(y) \right\| \ge \left| \frac{1}{M} \sum_{i=1}^{M} \frac{\partial}{\partial y[1]} h_i(y) \right| > 1.$$

Case 2) Suppose there exists $2 \le j \le k$ such that $|y[j-1]| \ge 1$.

We choose j so that $|y[j-1]| \ge 1$ and $|y[j]| < 1$. Therefore, depending on the choices of (i, j) we have three cases

$$\frac{\partial h_i(y)}{\partial y[j]} = \begin{cases} -3\left(\Psi\left(-y[i-1]\right)\Phi'\left(-y[j]\right) + \Psi\left(y[i-1]\right)\Phi'\left(y[j]\right)\right), & q \in [1, \frac{M}{3}] \\ 0, & q \in [\frac{M}{3}+1, \frac{2M}{3}] \\ -3\left(\Psi'\left(-y[j]\right)\Phi\left(-y[i+1]\right) + \Psi'\left(y[j]\right)\Phi\left(y[i+1]\right)\right), & q \in [\frac{2M}{3}+1, M] \end{cases}.$$

If $q \in [1, \frac{M}{3}]$, because $|y[j-1]| \ge 1$ and $|y[j]| < 1$, using Lemma 3.1 – (3), and the fact that the negative part is zero for Ψ, and Φ' is even function, the expression further equals to

$$-3 \cdot \Psi(|y[j-1]|)\Phi'\left(|y[j]|\right) \overset{(32)}{<} -3, \tag{124}$$

If $q \in [\frac{2M}{3}+1, M]$ the expression is obviously non-positive because both Ψ' and Φ are nonnegative. Overall, we have

$$\left| \frac{1}{M} \sum_{i=1}^{M} \frac{\partial h_i(y)}{\partial y[j]} \right| > \left| \frac{1}{M} \sum_{i=1}^{M/3} 3 \right| = 1.$$

This completes the proof. **Q.E.D.**

D. Proof of Lemma 3.4

Proof. First let us derive a useful property. Define $d := [d_1; d_2; \cdots; d_M]$ where d_i is the degree for node i; further define

$$\bar{x} := \frac{1}{M} \sum_{i=1}^{M} x_i, \quad \tilde{x_i} := x_i - \bar{x}, \quad \tilde{x} := [\tilde{x}_1; \tilde{x}_2; \cdots; \tilde{x}_M].$$

It is easy to observe that :

$$\tilde{x}^\top \mathbb{1} = 0, \quad \text{and} \quad \tilde{x} \notin \text{Null}(F^\top F).$$

Then the following holds:

$$x^\top F^\top F x = \sum_{(i,j):i\sim j} \|x_i - x_j\|^2 = \sum_{(i,j):i\sim j} \|\tilde{x}_i - \tilde{x}_j\|^2 = \tilde{x}^\top F^\top F \tilde{x} \ge \underline{\lambda}_{\min}(F^\top F)\|\tilde{x}\|^2. \tag{125}$$

Therefore the following holds:

$$\sum_{i=1}^{M} \|\bar{x} - x_i\|^2 \le \frac{1}{\underline{\lambda}_{\min}(F^\top F)} \sum_{(i,j):i\sim j} \|x_i - x_j\|^2 = \frac{1}{\underline{\lambda}_{\min}(P^{1/2}\mathcal{L}P^{1/2})} \sum_{(i,j):i\sim j} \|x_i - x_j\|^2. \tag{126}$$

Based on the above property, we have the following series of inequalities

$$\begin{aligned} \left\|\nabla \bar{f}(\bar{x})\right\|^2 &\le 2\left\| \frac{1}{M} \sum_{i=1}^{M} \left(\nabla f_i(\bar{x}) - \nabla f_i(x_i)\right) \right\|^2 + 2\left\| \frac{1}{M} \sum_{i=1}^{M} \nabla f_i(x_i) \right\|^2 \\ &\overset{\text{(i)}}{\le} \frac{2}{M} \sum_{i=1}^{M} \left\| \nabla f_i(\frac{1}{M} \sum_{j=1}^{M} x_j) - \nabla f_i(x_i) \right\|^2 + 2\left\| \frac{1}{M} \sum_{i=1}^{M} \nabla f_i(x_i) \right\|^2 \\ &\overset{\text{(ii)}}{\le} \frac{2}{M} \sum_{i=1}^{M} U^2 \left\| \frac{1}{M} \sum_{j=1}^{M} x_j - x_i \right\|^2 + 2\left\| \frac{1}{M} \sum_{i=1}^{M} \nabla f_i(x_i) \right\|^2 \\ &\overset{\text{(iii)}}{\le} \frac{2U}{M\underline{\lambda}_{\min}(P^{1/2}\mathcal{L}P^{1/2})} \sum_{(i,j):i\sim j} \|x_j - x_i\|^2 + 2\left\| \frac{1}{M} \sum_{i=1}^{M} \nabla f_i(x_i) \right\|^2 \end{aligned}$$

where in (i) and (iii) we have used the convexity of the function $\|\cdot\|^2$; in (ii) we used Lemma 3.2 – (3); in (iii) we have also used the assumption that $U \in (0,1)$ and (126). Overall we have

$$\left\|\frac{1}{M}\sum_{i=1}^{M}\nabla f_i(x_i)\right\|^2 + \frac{U}{M\underline{\lambda}_{\min}(P^{1/2}\mathcal{L}P^{1/2})}\sum_{(i,j):i\sim j}\|x_i - x_j\|^2 \geq \frac{1}{2}\left\|\nabla f(\bar{x})\right\|^2.$$

This completes the proof. **Q.E.D.**

E. Proof of Lemma 3.5

Proof. For a given $k \geq 2$, suppose that $x_i[k], x_i[k+1], ..., x_i[T] = 0$, $\forall i$, that is, support$\{x_i\} \subseteq \{1, 2, 3, ..., k-1\}$ for all i. Then $\Psi'(x_i[k]) = \Psi'(-x_i[k]) = 0$ for all i, and h_i has the following partial derivative when k is even:

$$\frac{\partial h_i(x_i)}{\partial x_i[k]} = \begin{cases} -3\left(\Psi\left(-x_i[k-1]\right)\Phi'\left(-x_i[k]\right)\right) + 3\left(\Psi\left(x_i[k-1]\right)\Phi'\left(x_i[k]\right)\right), & i \in [1, \frac{M}{3}] \\ 0, & i \in [\frac{M}{3}+1, M] \end{cases} \tag{127}$$

and the following partial derivative when k is odd and $k \geq 3$:

$$\frac{\partial h_i(x_i)}{\partial x_i[k]} = \begin{cases} 0, & i \in [1, \frac{2M}{3}] \\ -3\left(\Psi\left(-x_i[k-1]\right)\Phi'\left(-x_i[k]\right)\right) + 3\left(\Psi\left(x_i[k-1]\right)\Phi'\left(x_i[k]\right)\right), & i \in [\frac{2M}{3}+1, M] \end{cases}. \tag{128}$$

Recall that for any algorithm in class $\mathcal{A}$ or $\mathcal{A}'$, each agent is only able to compute linear combination of historical gradient and neighboring iterates [cf. (14) and (15)]. Therefore, for a given node i, as long as the kth element of the gradient as well as that of its neighbors have never been updated once, $x_i[k]$ remains to be zero. Combining this observation with the above two expressions for $\frac{\partial h_i(x_i)}{\partial x_i[k]}$, we can conclude that when support$\{x_i\} \subseteq \{1, 2, 3, ..., k-1\}$ for all i, then in the next iteration $x_i[k]$ will be possibly non-zero on the node $i \in [1, \frac{M}{3}]$ for even k and $i \in [\frac{2M}{3}+1, M]$ for odd k, and all other nodes still have $x_j[k] = 0$, $\forall\ j \neq i$.

Now suppose that the initial solution is $x_i[k] = 0$ for all (i,k). Then at the first iteration only $\frac{\partial h_i(x_i)}{\partial x_i[1]}$ is non-zero for all i, due to the fact that $\frac{\partial h_i(x_i)}{\partial x_i[1]} = \Psi(1)\Phi'(0) = 4(1-e^{-1})$ for all i from (118). If follows that even if every node is able to compute its local gradient, and can communicate with their neighbors, it is only possible to have $x_i[1] \neq 0, \forall i$. At the second iteration, we can use (127) to conclude that it is only possible to have $\frac{\partial h_r(x_r)}{\partial x_r[k]} \neq 0$ for some $r \in [1,\ M/3]$, therefore when using an algorithm in class $\mathcal{A}$, we can conclude that $x_i[2] = 0$ for all $i \notin [1,\ M/3]$.

Then following our construction (28), we know the nodes in the set $[1, \frac{M}{3}]$ and the set $[\frac{2M}{3}+1, M]$ have minimum distance $M/3$. It follows that using an algorithm in $\mathcal{A}$ or $\mathcal{A}'$, it takes at least $M/3$ iterations for the non-zero $x_r[2]$ and the corresponding gradient vector to propagate to at least one node in set $[2M/3+1, M]$. Once we have $x_j[2] \neq 0$ for some $j \in [2M/3+1,\ M]$, then according to (128), it is possible to have $\frac{\partial h_j(x_j)}{\partial x_j[3]} \neq 0$, and once this gradient becomes non-zero, the corresponding variable $x_j[3], j \in [2M/3+1,\ M]$ can become nonzero.

Following the above procedure, it is clear that we need at least $\frac{MT}{3}$ iterates and T computations to make $x_i[T]$ possibly non-zero. **Q.E.D.**

F. Proof of Theorem 3.1

Now we are ready to prove our first main result.

Proof of Theorem 3.1. By Lemma 3.5 we have $\bar{x}[T] = 0$ for all $t < \frac{M+3}{3}T$. Then by applying Lemma 3.2 – (2) and Lemma 3.3, we can conclude that the following holds

$$\left\|\nabla\bar{f}(\bar{x}[T])\right\| = \sqrt{2\epsilon}\left\|\nabla\bar{h}\left(\frac{\bar{x}[T]U}{75\pi\sqrt{2\epsilon}}\right)\right\| > \sqrt{2\epsilon}, \tag{129}$$

where the second inequality follows that there exists $k \in [T]$ such that $|\frac{\bar{x}[k]U}{75\pi\sqrt{2\epsilon}}| = 0 < 1$, then we can directly apply Lemma 3.3. Then by applying Lemma 3.4 gives $h^*_{(M+3)T/3} > \epsilon$, where h^*_T is defined in (16).

The third part of Lemma 3.2 ensures that f_i's are U-Lipschitz continuous gradient, and the first part shows

$$f(0) - \inf_x f(x) + \frac{\|d_0\|^2}{MU} \leq \frac{1650\pi^2\epsilon}{U}T,$$

Therefore we obtain

$$T \geq \left\lfloor \frac{\left(f(0) - \inf_x f(x) + \frac{\|d_0\|^2}{MU}\right)U}{1650\pi^2}\epsilon^{-1} \right\rfloor. \tag{130}$$

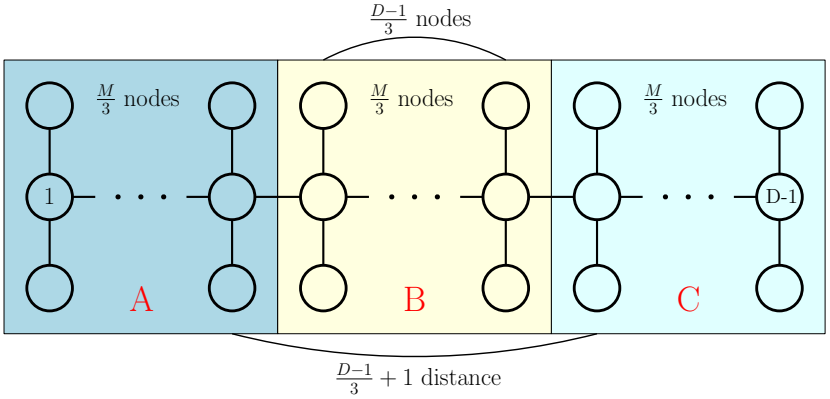


Fig. 9: The path-star graph used in our construction.

Summarizing the above argument, we have

$$t \geq \frac{M+3}{3} T \geq \frac{M+3}{3} \left\lfloor \frac{\left(f(0) - \inf_x f(x) + \frac{\|d_0\|^2}{MU}\right) U}{1650\pi^2} \epsilon^{-1} \right\rfloor.$$

By noting that for path graph $\xi(\mathcal{G}) \geq 1/M^2$, this completes the proof. **Q.E.D.**

G. Generalization

1) Uniform L_i, Fixed D and M: In this subsection, we would like to generalize Theorem 3.1 to a slightly wider class of networks (beyond the path graph used in our construction). Towards this end, consider a *path-star graph* shown in Fig. 9. The graph contains a path graph with $D-1$ nodes, and the remaining nodes are divided into $D-1$ groups, each with either $\lfloor M/(D-1) - 1 \rfloor$ or $\lfloor M/(D-1) - 1 \rfloor + 1$ nodes, and each group is connected to the nodes in the path graph by using a star topology. We have the following corollary to Theorem 3.1.

Corollary 10.1: Let $U \in (0,1)$ and ϵ be positive, and fix any D and M such that $D \leq M-1$. For any algorithm in class $\mathcal{A}$ or $\mathcal{A}'$, there exists a problem in class $\mathcal{P}_U^M$ and a network in class $\mathcal{N}_D^M$, so that to achieve accuracy $h_t^* < \epsilon$, it requires at least the following number iterations

$$t \geq \frac{D}{3} \left\lfloor \frac{\left(f(0) - \inf_x f(x) + \frac{\|d_0\|^2}{MU}\right) U}{1650\pi^2} \epsilon^{-1} \right\rfloor.$$

Alternatively, the above bound can be expressed as the following

$$t \geq \frac{\sqrt{D/(3M)}}{3\sqrt{\xi(\mathcal{G})}} \left\lfloor \frac{\left(f(0) - \inf_x f(x) + \frac{\|d_0\|^2}{MU}\right) U}{1650\pi^2} \epsilon^{-1} \right\rfloor.$$

Proof. Fix any D and M such that $D \leq M-1$, we can construct a path-star graph as described in Fig.9, whose diameter is D.

To show the lower bounds for such a graph, we split all M nodes into three sets $\mathcal{A}, \mathcal{B}, \mathcal{C}$ based on the main path, each with $\frac{M}{3}$ nodes (assume M is a multiple of 3), where $\mathcal{A}$ and $\mathcal{C}$ has minimum $\frac{D+2}{3}$ distance (assume $D-1$ is a multiple of 3). Then we construct the component functions h_i's as follows.

$$h_i(x_i) = \begin{cases} \Theta(x_i, 1) + 3 \sum_{j=1}^{\lfloor T/2 \rfloor} \Theta(x_i, 2j), & i \in \mathcal{A} \\ \Theta(x_i, 1), & i \in \mathcal{B} \\ \Theta(x_i, 1) + 3 \sum_{j=1}^{\lfloor T/2 \rfloor} \Theta(x_i, 2j+1), & i \in \mathcal{C} \end{cases} \tag{131}$$

Since the graph has diameter D in the above construction, and the distance between any two elements in $\mathcal{A}$ and $\mathcal{C}$ is at least $\frac{D+2}{3}$ (assume $D-1$ is a multiple of 3), by a similar step in Lemma 3.5 we can conclude that we need at least $(\frac{D+2}{3}+1)T$ iterations to achieve $x_i[T] \neq 0$. By applying (130), we can obtain the desired result.

To show the second result, note that from (24) we have

$$\sum_i d_i D \geq \frac{1}{\underline{\lambda}_{\min}(\mathcal{L})} \tag{132}$$

For the path-star graph under consideration, we have

$$\sum_i d_i = 2(D-1) - 2 + M \leq 3M$$

so the following holds:

$$D^2 \geq \frac{D/3M}{\underline{\lambda}_{\min}(\mathcal{L})}.$$

The desired result is then immediate. **Q.E.D.**

Finally, for the problem class with non-uniform Lipschitz constants, we can extend the previous result to any network in class $\mathcal{N}$ (by properly assigning different values of L_i's to different nodes). In this case the lower bound will be dependent on the spectral property of $\widehat{\mathcal{L}}$ as defined in (22) (expressed below for easy reference)

$$\widehat{\mathcal{L}} := L^{-1/2} F^\top K F L^{-1/2}. \tag{133}$$

H. Sketch of Proof for Corollary 3.1

To prove this result, we select the values of the coefficient set $\{L_i\}_{i=1}^M$, so that the "effective" network topology becomes a path. In particular, for any given network in $\mathcal{N}$, we can construct local functions as follows: First, along the longest path of size D, we distributed the functions into three sets $\mathcal{A}, \mathcal{B}, \mathcal{C}$, where $\mathcal{A}$ and $\mathcal{C}$ denotes the first and last $\frac{D}{3}$ nodes on the path respectively, and $\mathcal{B}$ denotes the rest nodes on the path. Second, for the rest of the functions not on the path, denoted as set $\mathcal{D}$, set their local functions to zero (or equivalently, set the corresponding L_i's to zero). Then, the local function belongs to each set can be expressed as:

$$h_i(x_i) = \begin{cases} \frac{M}{D}\Theta(x_i, 1) + \frac{3M}{D}\sum_{j=1}^{\lfloor T/2 \rfloor} \Theta(x_i, 2j), & i \in \mathcal{A} \\ \frac{M}{D}\Theta(x_i, 1), & i \in \mathcal{B} \\ \frac{M}{D}\Theta(x_i, 1) + \frac{3M}{D}\sum_{j=1}^{\lfloor T/2 \rfloor} \Theta(x_i, 2j+1), & i \in \mathcal{C} \\ 0, & i \in \mathcal{D} \end{cases} \tag{134}$$

This way the network reduces to a path graph. Note that the Lipschitz constant for the gradient of $h(y) = \frac{1}{M}\sum_{i=1}^M h_i(y)$ is still 1, and we can use the similar constructions and proof steps leading to Theorem 3.1 to prove the claim.