This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2017.2698261, IEEE Transactions on Control of Network Systems

Harnessing Smoothness to Accelerate Distributed Optimization

Guannan Qu, Na Li

***Abstract*—There has been a growing effort in studying the distributed optimization problem over a network. The objective is to optimize a global function formed by a sum of local functions, using only local computation and communication. Literature has developed consensus-based distributed (sub)gradient descent (DGD) methods and has shown that they have the same convergence rate $O(\frac{\log t}{\sqrt{t}})$ as the centralized (sub)gradient methods (CGD) when the function is convex but possibly nonsmooth. However, when the function is convex and smooth, under the framework of DGD, it is unclear how to harness the smoothness to obtain a faster convergence rate comparable to CGD's convergence rate. In this paper, we propose a distributed algorithm that, despite using the same amount of communication per iteration as DGD, can effectively harnesses the function smoothness and converge to the optimum with a rate of $O(\frac{1}{t})$. If the objective function is further strongly convex, our algorithm has a linear convergence rate. Both rates match the convergence rate of CGD. The key step in our algorithm is a novel gradient estimation scheme that uses history information to achieve fast and accurate estimation of the average gradient. To motivate the necessity of history information, we also show that it is impossible for a class of distributed algorithms like DGD to achieve a linear convergence rate without using history information even if the objective function is strongly convex and smooth.**

I. Introduction

Given a set of agents $\mathcal{N} = \{1, 2, \ldots, n\}$, each of which has a local convex cost function $f_i(x) : \mathbb{R}^N \to \mathbb{R}$, the objective of distributed optimization is to find x that minimizes the average of all the functions,

$$\min_{x\in\mathbb{R}^N} f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

using local communication and local computation. The local communication is defined through an undirected communication graph $\mathcal{G} = (V, E)$, where the nodes $V = \mathcal{N}$ and the edges $E \subset V \times V$. Agent i and j can send information to each other if and only if i and j are connected in graph $\mathcal{G}$. The local computation means that each agent can only make his decision based on the local function f_i and the information obtained from his neighbors.

This problem has recently received much attention and has found various applications in multi-agent control, distributed state estimation over sensor networks, large scale computation in machine/statistical learning [2]–[4], etc. As a concrete example, in the setting of distributed statistical learning, x is the parameter to infer, and f_i is the empirical loss function of the local dataset of agent i. Then minimizing f means empirical loss minimization that uses datasets of all the agents.

The early work of this problem can be found in [5], [6]. Recently, [7] (see also [8]) proposes a consensus-based distributed (sub)gradient descent (DGD) method where each agent performs a consensus step and then a descent step along the local (sub)gradient direction of f_i. Reference [9] applies a similar idea to develop a distributed dual averaging algorithm. Extensions of these work have been proposed that deal with various realistic conditions, such as stochastic subgradient errors [10], directed or random communication graphs [11]–[13], linear scaling in network size [14], heterogeneous local constraints [15], [16]. Overall speaking, these DGD (or DGD-like) algorithms are designated for nonsmooth functions and they achieve the same convergence speed $O(\frac{\log t}{\sqrt{t}})$ [17] as centralized subgradient descent. They can also be applied to smooth functions, but when doing so they either do not guarantee exact convergence when using a constant step size [13], [18], or have a convergence rate of at most $\Omega(\frac{1}{t^{2/3}})$ when using a diminishing step size [19], slower than the normal Centralized Gradient Descent (CGD) method's $O(\frac{1}{t})$ [20]. Therefore, DGD does not fully exploit the function smoothness and has a slower convergence rate compared with CGD. In fact, we prove in this paper that for strongly convex and smooth functions, it is impossible for DGD-like algorithms to achieve the same linear convergence rate as CGD (Theorem 6). Alternatively, [19], [21] suggest that it is possible to achieve faster convergence for smooth functions, by performing multiple consensus steps after each gradient evaluation. However, it places a larger communication burden. These drawbacks pose the need for distributed algorithms that effectively harness the smoothness to achieve faster convergence, using only *one* communication step per gradient evaluation iteration.

In this paper, we propose a distributed algorithm that can effectively harness the smoothness, and achieve a convergence rate that matches CGD, using only one communication step per gradient evaluation. Specifically, our algorithm achieves a $O(\frac{1}{t})$ rate for smooth and convex functions (Theorem 3), and a linear convergence rate ($O(\gamma^t)$ for some $\gamma \in (0, 1)$) for smooth and strongly convex functions (Theorem 1).[1] The convergence rates match the convergence rates of CGD, but with worse constants due to the distributed nature of the problem. Our algorithm is a combination of gradient descent and a novel gradient estimation scheme that utilizes history information to achieve fast and accurate estimation of the average gradient. To show the necessity of history information, we also prove that it is impossible for a class of distributed algorithms like DGD

Guannan Qu and Na Li are affiliated with John A. Paulson School of Engineering and Applied Sciences at Harvard University. Email: gqu@g.harvard.edu, nali@seas.harvard.edu. This work is supported under NSF ECCS 1608509 and NSF CAREER 1553407.

Some preliminary work has been presented in the conference version of the paper [1]. This paper includes more theoretical results and detailed proofs compared to [1].

[1] A recent paper [22] also achieves similar convergence rate results using a different algorithm. A detailed comparison between our algorithm and [22] will be given in Section III-C.

to achieve a linear convergence rate without using history information even if we restrict the class of objective functions to be strongly convex and smooth (Theorem 6).

Moreover, our scheme can be cast as a general method for obtaining distributed versions of many first-order optimization algorithms, like Nesterov gradient descent [20]. We expect the distributed algorithm obtained in this way will have a similar convergence rate as its centralized counterpart. Some preliminary results on applying the scheme to Nesterov gradient descent can be found in our recent work [23].

Besides [22] that studies an algorithm with similar performance, we note that variants of the algorithm in this paper have appeared in a few recent work, but these work has different focus compared to ours. Reference [24] focuses on uncoordinated step sizes. It proves the convergence of the algorithm but does not provide convergence rate results. Reference [25], [26] focus on (possibly) nonconvex objective functions and thus have different step size rules and convergence results compared to ours. More recently, [27], [28] prove a linear convergence rate of the same algorithm for strongly convex and smooth functions, with [27] focusing on time-varying graphs and [28] focusing on uncoordinated step sizes. At last, [27] and [29] have studied a variant of the algorithm for directed graphs. To the best of our knowledge, our work is the first to study the $O(\frac{1}{t})$ convergence rate of the algorithm without the strongly convex assumption (with only the convex and smooth assumption). Also, our way of proving the linear convergence rate is inherently different from that of [27], [28].

At last, we would like to emphasize that the focus of this paper is the consensus-based, first-order distributed algorithms. We note that there are other types of distributed optimization algorithms, like second-order distributed algorithms [30]–[32], Alternating Direction Method of Multipliers (ADMM) [33], [34] etc. However these methods are inherently different from the algorithms studied in this paper, and thus are not discussed in this paper.

The rest of the paper is organized as follows. Section II formally defines the problem and presents our algorithm and results. Section III reviews previous methods, introduces an impossibility result and motivates our approach. Section IV proves the convergence of our algorithm. Lastly, Section V provides numerical simulations and Section VI concludes the paper.

Notation. Throughout the rest of the paper, n is the number of agents, and N is the dimension of the domain of f_i. Notation $i, j \in \{1, 2, \ldots, n\}$ are indices for the agents, while $t, k, \ell \in \mathbb{N}$ are indices for iteration steps. Notation $\|\cdot\|$ denotes 2-norm for vectors, and Frobenius norm for matrices. Notation $\langle \cdot, \cdot \rangle$ denotes inner product for vectors. Notation $\rho(\cdot)$ denotes spectral radius for square matrices, and $\mathbf{1}$ denotes an n-dimensional all one column vector. All vectors, when having dimension N (the dimension of the domain of f_i), will be regarded as row vectors. As a special case, all gradients, $\nabla f_i(x)$ and $\nabla f(x)$ are interpreted as N-dimensional row vectors. Notation '$\leq$', when applied to vectors of the same dimension, denotes element wise 'less than or equal to'.

II. Problem and Algorithm

A. Problem Formulation

Consider n agents, $\mathcal{N} = \{1, 2, \ldots, n\}$, each of which has a convex function $f_i : \mathbb{R}^N \to \mathbb{R}$. The objective of distributed optimization is to find x to minimize the average of all the functions, i.e.

$$\min_{x \in \mathbb{R}^N} f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(x) \tag{1}$$

using local communication and local computation. The local communication is defined through an *undirected* and *connected* communication graph $\mathcal{G} = (V, E)$, where the nodes $V = \mathcal{N}$ and edges $E \subset V \times V$. Agent i and j can send information to each other if and only if $(i, j) \in E$. The local computation means that each agent can only make its decision based on the local function f_i and the information obtained from its neighbors.

Throughout the paper, we assume that the set of minimizers of f is non-empty. We denote x^* as one of the minimizers and f^* as the minimal value. We will study the case where each f_i is convex and β-smooth (Assumption 1) and also the case where each f_i is in addition α-strongly convex (Assumption 2). The definition of β-smooth and α-strongly convex are given in Definition 1 and 2 respectively.

Definition 1. *A function $\xi : \mathbb{R}^N \to \mathbb{R}$ is β-smooth if ξ is differentiable and its gradient is β-Lipschitz continuous, i.e. $\forall x, y \in \mathbb{R}^N$,*

$$\|\nabla \xi(x) - \nabla \xi(y)\| \leq \beta \|x - y\|.$$

Definition 2. *A function $\xi : \mathbb{R}^N \to \mathbb{R}$ is α-strongly convex if $\forall x, y \in \mathbb{R}^N$,*

$$\xi(y) \geq \xi(x) + \langle \nabla \xi(x), y - x \rangle + \frac{\alpha}{2} \|y - x\|^2.$$

Assumption 1. *$\forall i, f_i$ is convex and β-smooth.*

Assumption 2. *$\forall i$, f_i is α-strongly convex.*

Since f is an average of the f_i's, Assumption 1 also implies f is convex and β-smooth, and Assumption 2 also implies f is α-strongly convex.

B. Algorithm

The algorithm we will describe is a consensus-based distributed algorithm. Each agent weighs its neighbors' information to compute its local decisions. To model the weighting process, we introduce a consensus weight matrix, $W = [w_{ij}] \in \mathbb{R}^{n \times n}$, which satisfies the following properties:[2]

(a) For any $(i, j) \in E$, we have $w_{ij} > 0$. For any $i \in \mathcal{N}$, we have $w_{ii} > 0$. For other (i, j), we have $w_{ij} = 0$.

(b) Matrix W is doubly stochastic, i.e. $\sum_{i'} w_{i'j} = \sum_{j'} w_{ij'} = 1$ for all $i, j \in \mathcal{N}$.

As a result, $\exists \sigma \in (0, 1)$ which is the spectral norm of $W - \frac{1}{n}\mathbf{1}\mathbf{1}^T$,[3] such that for any $\omega \in \mathbb{R}^{n \times 1}$, we have $\|W\omega - \mathbf{1}\bar{\omega}\| =$

[2] The selection of the consensus weights is an intensely studied problem, see [35], [36].

[3] To see why $\sigma \in (0, 1)$, note that WW^T is a doubly stochastic matrix. Then since the graph is connected, WW^T has a unique largest eigenvalue, which is 1 and it is simple, with eigenvector $\mathbf{1}$. Hence all the eigenvalues of $(W - \frac{1}{n}\mathbf{1}\mathbf{1}^T)(W - \frac{1}{n}\mathbf{1}\mathbf{1}^T)^T = WW^T - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ are strictly less than 1, which implies $\sigma \in (0, 1)$.

$\|(W - \frac{1}{n}\mathbf{1}\mathbf{1}^T)(\omega - \mathbf{1}\bar{\omega})\| \leq \sigma\|\omega - \mathbf{1}\bar{\omega}\|$ where $\bar{\omega} = \frac{1}{n}\mathbf{1}^T\omega$ (the average of the entries in ω) [36]. This 'averaging' property will be frequently used in the rest of the paper.

In our algorithm, each agent i keeps an estimate of the minimizer $x_i(t) \in \mathbb{R}^{1\times N}$, and another vector $s_i(t) \in \mathbb{R}^{1\times N}$ which is designated to estimate the average gradient, $\frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i(t))$. The algorithm starts with an arbitrary $x_i(0)$, and with $s_i(0) = \nabla f_i(x_i(0))$. The algorithm proceeds using the following update,

$$x_i(t+1) = \sum_{j=1}^n w_{ij}x_j(t) - \eta s_i(t) \tag{2}$$

$$s_i(t+1) = \sum_{j=1}^n w_{ij}s_j(t) + \nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t)) \tag{3}$$

where $[w_{ij}]_{n\times n}$ are the consensus weights and $\eta > 0$ is a fixed step size. Because $w_{ij} = 0$ when $(i,j) \notin E$, each node i only needs to send $x_i(t)$ and $s_i(t)$ to its neighbors. Therefore, the algorithm can be operated in a fully distributed fashion, with only local communication. Note that the two consensus weight matrices in step (2) and (3) can be chosen differently. We use the same matrix W to carry out our analysis for the purpose of easy exposition.

The update equation (2) is similar to the algorithm in [7] (see also (5) in Section III), except that the gradient is replaced with $s_i(t)$ which follows the update rule (3). In Section III and IV-B, we will discuss the motivation and the intuition behind this algorithm.

Remark 1. *The key of our algorithm is the gradient estimation scheme (3) and it can be used to obtain distributed versions of many other gradient-based algorithms. For example, suppose a centralized algorithm is in the following form,*

$$x(t+1) = \mathcal{F}_t(x(t), \nabla f(x(t)))$$

where $x(t)$ is the state, $\mathcal{F}_t$ is the update equation. We can write down a distributed algorithm as

$$x_i(t+1) = \mathcal{F}_t(\sum_j w_{ij}x_j(t), s_i(t))$$

$$s_i(t+1) = \sum_j w_{ij}s_j(t) + \nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t)).$$

Our conjecture is that for a broad range of centralized algorithms, the distributed algorithm obtained as above will have a similar convergence rate as the centralized one. Our ongoing work includes applying the above scheme to other centralized algorithms like Nesterov gradient method. Some of our preliminary results are in [23].

C. Convergence of the Algorithm

To state the convergence results, we need to define the following average sequences.

$$\bar{x}(t) = \frac{1}{n}\sum_{i=1}^n x_i(t) \in \mathbb{R}^{1\times N}, \bar{s}(t) = \frac{1}{n}\sum_{i=1}^n s_i(t) \in \mathbb{R}^{1\times N}$$

$$g(t) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i(t)) \in \mathbb{R}^{1\times N}$$

We also define the gradient of f evaluated at $\bar{x}(t)$, $h(t) = \nabla f(\bar{x}(t)) \in \mathbb{R}^{1\times N}$. We summarize our convergence results here.

Theorem 1. *Under the smooth and strongly convex assumptions (Assumption 1 and 2), when η is such that the matrix*

$$G(\eta) = \begin{bmatrix} (\sigma+\beta\eta) & \beta(\eta\beta+2) & \eta\beta^2 \\ \eta & \sigma & 0 \\ 0 & \eta\beta & \lambda \end{bmatrix}$$
$$\textit{where } \lambda = \max(|1-\alpha\eta|, |1-\beta\eta|)$$

has spectral radius $\rho(G(\eta)) < 1$, then $\forall i$, $\|\bar{x}(t) - x^*\|$ (distance to the optimizer), $\|x_i(t) - \bar{x}(t)\|$ (consensus error), and $\|s_i(t) - g(t)\|$ (gradient estimation error) are all decaying with rate $O(\rho(G(\eta))^t)$. Moreover, we have $f(x_i(t)) - f^*$ (objective error) is decaying with rate $O(\rho(G(\eta))^{2t})$.

The following lemma provides an explicit upper bound on the convergence rate $\rho(G(\eta))$.

Lemma 2. *When $0 < \eta < \frac{1}{\beta}$, we have $\rho(G(\eta)) \leq \max(1 - \frac{\alpha\eta}{2}, \sigma + 5\sqrt{\eta\beta}\sqrt{\frac{\beta}{\alpha}})$. Specifically, when $\eta = \frac{\alpha}{\beta^2}(\frac{1-\sigma}{6})^2$, $\rho(G(\eta)) \leq 1 - \frac{1}{2}(\frac{\alpha}{\beta}\frac{1-\sigma}{6})^2 < 1$.*

If we drop the strongly convex assumption, we have the following result.

Theorem 3. *Under the smooth assumption (Assumption 1), when $0 < \eta \leq \frac{(1-\sigma)^2}{160\beta}$, the following is true.*

(a) The average objective error satisfy,

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^n \left[f(\hat{x}_i(t+1)) - f^*\right] \\ &\leq \frac{1}{t+1}\left\{\frac{\|\bar{x}(0) - x^*\|^2}{2\eta}\right. \\ &\left. + \frac{36\beta}{(1-\sigma)^2}\left[\frac{1}{\beta\sqrt{n}}\|s(0) - \mathbf{1}g(0)\| + \frac{2}{\sqrt{n}}\|x(0) - \mathbf{1}\bar{x}(0)\|\right]^2\right\}\end{aligned}$$

where $\hat{x}_i(t+1)$ is the running average of agent i defined to be $\hat{x}_i(t+1) = \frac{1}{t+1}\sum_{k=0}^t x_i(k+1)$.

(b) The consensus error satisfy,

$$\begin{aligned}&\min_{0\leq k\leq t}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \leq \\ &\frac{1}{t}\times\left\{\frac{1740}{(1-\sigma)^4}\left[\frac{1}{\beta}\|s(0) - \mathbf{1}g(0)\| + 2\|x(0) - \mathbf{1}\bar{x}(0)\|\right]^2\right. \\ &\qquad\left. + \frac{24}{(1-\sigma)^2}\|\mathbf{1}\bar{x}(0) - \mathbf{1}x^*\|^2\right\}.\end{aligned}$$

Since the objective error is nonnegative, we have for each $i \in \mathcal{N}$, $f(\hat{x}_i(t+1)) - f^* \leq n \times \frac{1}{n}\sum_{j=1}^n[f(\hat{x}_j(t+1)) - f^*]$. This leads to the following simple corollary of Theorem 3 regarding the individual objective errors $f(\hat{x}_i(t+1)) - f^*$.

Corollary 4. *Under the conditions of Theorem 3, we have $\forall i = 1, \ldots, n$,*

$$\begin{aligned}&f(\hat{x}_i(t+1)) - f^* \\ &\leq \frac{1}{t+1}\left\{\frac{\|\mathbf{1}\bar{x}(0) - \mathbf{1}x^*\|^2}{2\eta}\right.\end{aligned}$$

$$+\frac{36\beta}{(1-\sigma)^2}\Big[\frac{1}{\beta}\|s(0)-\mathbf{1}g(0)\|+2\|x(0)-\mathbf{1}\bar{x}(0)\|\Big]^2\Big\}.$$

Remark 2. *Our algorithm preserves the convergence rate of CGD, in the sense that it has a linear convergence rate when each f_i is strongly convex and smooth, and a convergence rate of $O(\frac{1}{t})$ when each f_i is just smooth. However, we note that the linear convergence rate constant $\rho(G(\eta))$ is usually worse than CGD; and moreover, in both cases, our algorithm has a worse constant in the big O terms. Moreover, compared to CGD, the step size rules depend on the consensus matrix W (Lemma 2 and Theorem 3).*

Remark 3. *By Lemma 2 and Theorem 3(a), the convergence rate of our algorithm does not explicitly depend on n but depends on n through the second largest singular value σ of the consensus matrix W.*[4] *The relationship between σ and n is studied in [36] for a general class of W, and in Lemma 4 of [9] when W is selected using the Lapalcian method (to be introduced in Section V-A).*

In Lemma 2 and Theorem 3, the step sizes depend on the parameter σ which requires global knowledge of graph $\mathcal{G}$ to compute. To make the algorithm fully distributed, we now relax the step size rules such that each agent only needs to know an upper bound U on the number of agents n, i.e. $U \geq n$. To achieve this, we require each agent select the weights W to be the lazy Metropolis matrix [14], i.e.

$$w_{ij}=\begin{cases}\frac{1}{2\max(d_i,d_j)} & \text{if } i\neq j,\ (i,j) \text{ connected.}\\ 1-\sum_{q\in N_i}\frac{1}{2\max(d_i,d_q)} & \text{if } i=j.\\ 0 & \text{elsewhere.}\end{cases} \tag{4}$$

In (4), d_i denotes the degree of agent i in graph $\mathcal{G}$, and N_i denotes the set of neighbors of agent i. Note that the W in (4) can be computed distributedly, since each agent only needs to know its own degree and its neighbor's degree to compute w_{ij}. Moreover, Lemma 2.1 in [14] shows that if W is selected according to (4), $\sigma < 1-\frac{1}{71n^2}\leq 1-\frac{1}{71U^2}$. Combining this with our original step size rules in Lemma 2 and Theorem 3, we have the following corollary in which the step size rules are relaxed to only depend on U.

Corollary 5. (a) *Under the assumptions of Theorem 1, if W is chosen according to (4) and $\eta=\frac{\alpha}{\beta^2}(\frac{1}{426U^2})^2$, then $\rho(G(\eta))\leq 1-\frac{1}{2}(\frac{\alpha}{\beta}\frac{1}{426U^2})^2$ and the convergence results in Theorem 1 hold.*

(b) *Under the assumptions of Theorem 3, if W is chosen according to (4) and $0<\eta\leq\frac{1}{160\beta(71U^2)^2}$, then the statements in Theorem 3 and Corollary 4 hold.*

III. Algorithm development: Motivation

In this section, we will briefly review distributed first-order optimization algorithms that are related to our algorithm and discuss their limitations which motivates our algorithm development. In particular, we will formally provide an impossibility result regarding the limitations. Lastly we will discuss the literature that motivates the idea of harnessing the smoothness from history information.

A. Review of Distributed First-Order Optimization Algorithms

To solve the distributed optimization problem (1), consensus-based DGD (Distributed (sub)gradient descent) methods have been developed, e.g., [7], [9]–[14], [17]–[19], [21], [22], that combine a consensus algorithm and a first order optimization algorithm. For a review of consensus algorithms and first order optimization algorithms, we refer to references [36] and [20], [37], [38] respectively. For the sake of concrete discussion, we focus on the algorithm in [7], where each agent i keeps an local estimate of the solution to (1), $x_i(t)$ and it updates $x_i(t)$ according to,

$$x_i(t+1)=\sum_j w_{ij}x_j(t)-\eta_t g_i(t) \tag{5}$$

where $g_i(t)\in\partial f_i(x_i(t))$ is a subgradient of f_i at $x_i(t)$ (f_i is possibly nonsmooth), and η_t is the step size, and w_{ij} are the consensus weights. Algorithm (5) is essentially performing a consensus step followed by a standard subgradient descent along the local subgradient direction $g_i(t)$. Results in [17] show that the running best of the objective $f(x_i(t))$ converges to the minimum f^* with rate $O(\frac{\log t}{\sqrt{t}})$ if using a diminishing step size $\eta_t=\Theta(\frac{1}{\sqrt{t}})$. This is the same rate as the centralized subgradient descent algorithm up to a $\log t$ factor.

When the f_i's are smooth, the subgradient $g_i(t)$ will equal the gradient $\nabla f_i(x_i(t))$. However, as shown in [19], even in this case the convergence rate of (5) can not be better than $\Omega(\frac{1}{t^{2/3}})$ when using a vanishing step size. In contrast, the CGD (centralized gradient descent) method,

$$x(t+1)=x(t)-\eta\nabla f(x) \tag{6}$$

converges to the optimum with rate $O(\frac{1}{t})$ if the stepsize η is a small enough constant. Moreover, when f is further strongly convex, CGD (6) converges to the optimal solution with a linear rate. If a fixed step size η is used in DGD (5), though the algorithm runs faster, the method only converges to a neighborhood of the optimizer [13], [18]. This is because even if $x_i(t)=x^*$ (the optimal solution), $\nabla f_i(x_i(t))$ is not necessarily zero.

To fix this problem of non-convergence, it has been proposed to use multiple consensus steps after each gradient descent [19], [21]. One example is provided as follows:

$$\begin{aligned} y_i(t,0) &= x_i(t)-\eta\nabla f_i(x_i(t)) && \text{(7a)}\\ y_i(t,k) &= \sum_j w_{ij}y_j(t,k-1), k=1,2,\ldots,c_t && \text{(7b)}\\ x_i(t+1) &= y_i(t,c_t). && \text{(7c)}\end{aligned}$$

For each gradient descent step (7a), after c_t consensus steps ($c_t=\Theta(\log t)$ in [19], and $c_t=\Theta(t)$ in [21]), the agents' estimates $x_i(t+1)$ are sufficiently averaged, and it is as if each agent has performed a descent along the average gradient $\frac{1}{n}\sum_i\nabla f_i(x_i(t))$. As a result, algorithm (7) addresses the non-convergence problem mentioned above. However, it places a large communication burden on the agents: the further

[4] In Theorem 3(a) we consider the quantity $\frac{1}{\sqrt{n}}\|s(0)-\mathbf{1}g(0)\|$ (and similarly $\frac{1}{\sqrt{n}}\|x(0)-\mathbf{1}\bar{x}(0)\|$) to be *not* explicitly dependent on n, since this quantity equals $\sqrt{\frac{1}{n}\sum_{i=1}^n\|s_i(0)-g(0)\|^2}$ and is essentially an average of some initial condition across the agents.

the algorithm proceeds, the more consensus steps after each gradient step are required. In addition, even if the algorithm already reaches the optimizer $x_i(t) = x^*$, because of (7a) and because $\nabla f_i(x^*)$ might be non-zero, $y_i(t,0)$ will deviate from the optimizer, and then a large number of consensus steps in (7b) are needed to average out the deviation. All these drawbacks pose the need for alternative distributed algorithms that effectively harness the smoothness to achieve faster convergence, using only *one* (or a constant number of) communication step(s) per gradient evaluation.

B. An Impossibility Result

To compliment the preceding discussion, here we provide an impossibility result for a class of distributed first-order algorithms which include algorithms like (5). We use notation $-i$ to denote the set $\mathcal{N}/\{i\}$. The class of algorithms we consider obey the following updating rule, $\forall i \in \mathcal{N}$

$$x_i(t) = \mathcal{F}(\mathcal{H}(x_i(t-1), x_{-i}(t-1), \mathcal{G}), \eta_t \nabla f_i(x_i(t-1))). \quad (8)$$

Here both $\mathcal{H}$ and $\mathcal{F}$ denote general functions with the following properties. Function $\mathcal{H}$ captures how agents use their neighbors' information, and $\mathcal{H}$ is assumed to be a continuous function of the component $x_j(t)$, $j \in \mathcal{N}$. Note that $\mathcal{H}$ can be interpreted as the consensus step. Function $\mathcal{F}$ is a function of $\mathcal{H}$ and the scaled gradient direction $\eta_t \nabla f_i(x_i(t-1))$, and $\mathcal{F}(\cdot,\cdot)$ is assumed to be L-Lipschitz continuous in its second variable (when fixing the first variable). Note that $\mathcal{F}$ can be interpreted as a first-order update rule, such as the (projected) gradient descent, mirror descent, etc. Parameter η_t can be considered as the step size, and we assume it has a limit η^* as $t \to \infty$. We will show that for strongly convex and smooth cost functions, any algorithm belonging to this class will not have a linear convergence rate, which is in contrast to the linear convergence of the centralized methods.

Theorem 6. *Consider a simple case where $\mathcal{N} = \{1,2\}$, i.e. there are only two agents. Assume the objective functions $f_1, f_2 : \mathbb{R}^N \to \mathbb{R}$ are α-strongly convex and β-smooth. Suppose for any $f_1, f_2, x_1(0), x_2(0)$, $\lim_{t\to\infty} x_i(t) = x^*$ under algorithm (8), where x^* is the minimizer of $f_1 + f_2$. Then there exist $f_1, f_2, x_1(0), x_2(0)$ such that for any $\delta \in (0,1)$ and $T \geq 0$, there exist $t \geq T$, s.t. $\|x_i(t+1) - x^*\| \geq \delta \|x_i(t) - x^*\|$.*

Proof: We first show $\eta^* = 0$. Assume the contrary holds, $\eta^* \neq 0$, then for any objective functions f_1, f_2, and any starting point, we have $x_1(t), x_2(t) \to x^*$, which implies $\mathcal{F}(\mathcal{H}(x_1(t), x_2(t)), \eta_t \nabla f_1(x_1(t))) \to x^*$. By the continuity of $\mathcal{F}$ and $\mathcal{H}$ and ∇f_1, we have $x^* = \mathcal{F}(\mathcal{H}(x^*, x^*), \eta^* \nabla f_1(x^*))$. We can choose f_1, f_2 to be simple quadratic functions such that $(x^*, \nabla f_1(x^*))$ can be any point in $\mathbb{R}^N \times \mathbb{R}^N$. Hence, since $\eta^* \neq 0$, we have, for any $x, y \in \mathbb{R}^N$, $x = \mathcal{F}(\mathcal{H}(x,x), y)$. This is impossible, because if we let the objective functions be $f_1(x) = f_2(x) = \frac{\alpha}{2}\|x\|^2$, and we start from $x_1(0) = x_2(0) \neq 0$, we will have that the trajectory $x_i(t)$ stays fixed $x_1(t) = x_2(t) = x_1(0) = x_2(0)$, not converging to the minimizer 0. This is a contradiction. Hence, $\eta^* = 0$.

In the rest of the proof we focus on a restricted scenario in which $f_1 = f_2$ and $x_1(0) = x_2(0)$. In this scenario, we can easily check $x_1(t)$ always equals $x_2(t)$ by induction.[5] In light of this, we introduce notation $x(t) \triangleq x_1(t) = x_2(t)$ and also $f \triangleq f_1 = f_2$. Using the new notation, the update equation for $x(t)$ becomes

$$\begin{aligned} x(t+1) &= \mathcal{F}(\mathcal{H}(x(t), x(t)), \eta_t \nabla f(x(t))) \\ &\triangleq \tilde{\mathcal{F}}(x(t), \eta_t \nabla f(x(t))) \end{aligned}$$

where we have defined $\tilde{\mathcal{F}}(u,v) = \mathcal{F}(\mathcal{H}(u,u), v)$. By the continuity of $\mathcal{F}$ and $\mathcal{H}$, we have $\tilde{F}$ is continuous. Since $\mathcal{F}(\cdot,\cdot)$ is L-Lipschitz continuous in its second variable, we have $\tilde{F}(\cdot,\cdot)$ is also L-Lipschitz continuous in its second variable. Under the new notation, the assumption of the Theorem ($x_1(t)$ and $x_2(t)$ converge to the minimizer of $f_1 + f_2$) can be rephrased as that $x(t)$ converges to the minimizer of f.

We now claim that $u = \tilde{\mathcal{F}}(u,0)$ for any $u \in \mathbb{R}^N$. To see this, we fix $u \in \mathbb{R}^N$ and consider a specific case of the function, $f(x) = \frac{\alpha}{2}\|x - u\|^2$. Then by the assumption of the Theorem, $x(t)$ will converge to the minimizer of f, which in this case is u. The fact $x(t) \to u$ also implies $\eta_t \nabla f(x(t)) \to \eta^* \nabla f(u) = 0$. Now let $t \to \infty$ in the update equation $x(t+1) = \tilde{\mathcal{F}}(x(t), \eta_t \nabla f(x(t)))$. By the continuity of $\tilde{\mathcal{F}}$, we have that $u = \tilde{\mathcal{F}}(u,0)$. Since we can arbitrarily pick u, we have $u = \tilde{F}(u,0)$ for all $u \in \mathbb{R}^N$.

Now we are ready to prove the Theorem. Notice that for any objective function f, if we start from $x(0) \neq x^*$ (x^* is the unique minimizer of f), then the generated sequence $x(t)$ satisfies

$$\begin{aligned} \|x(t+1) - x^*\| &= \|\tilde{\mathcal{F}}(x(t), \eta_t \nabla f(x(t))) - x^*\| \\ &\overset{(a)}{\geq} \|\tilde{\mathcal{F}}(x(t), 0) - x^*\| \\ &\quad - \|\tilde{\mathcal{F}}(x(t), \eta_t \nabla f(x(t))) - \tilde{\mathcal{F}}(x(t), 0)\| \\ &\overset{(b)}{\geq} \|x(t) - x^*\| - L\eta_t \|\nabla f(x(t))\| \\ &\overset{(c)}{\geq} (1 - \eta_t L \beta)\|x(t) - x^*\|. \end{aligned}$$

where (a) is from triangle inequality; (b) is because $\tilde{\mathcal{F}}(u,0) = u, \forall u \in \mathbb{R}^N$ and $\tilde{\mathcal{F}}(\cdot,\cdot)$ is L-Lipschitz continuous in its second variable; (c) is because f is β-smooth. The Theorem then follows from the fact that $\eta_t L\beta \to 0$. □

C. Harnessing Smoothness via History Information

Motivated by the previous discussion and the impossibility result, we seek for alternative methods to exploit smoothness to develop faster distributed algorithms. Firstly we note that one major reason for the slow convergence of DGD is the decreasing step size η_t. This motivates us to use a constant step size η in our algorithm (2). But we have discussed that a constant η will lead to optimization error due to the fact that $\nabla f_i(x_i(t))$ could be very different from the average gradient $g(t) = \frac{1}{n}\sum_i \nabla f_i(x_i(t))$. However, because of smoothness, $\nabla f_i(x_i(t+1))$ and $\nabla f_i(x_i(t))$ would be close (as well as $g(t+1)$ and $g(t)$) if $x_i(t+1)$ and $x_i(t)$ are close, which is exactly the case when the algorithm is coming close to the minimizer x^*. This motivates the second step of our

[5] Firstly $x_1(0) = x_2(0)$. Then if assuming $x_1(t) = x_2(t)$, using $f_1 = f_2$ we have $x_1(t+1) = \mathcal{F}(\mathcal{H}(x_1(t), x_2(t)), \eta_t \nabla f_1(x_1(t))) = \mathcal{F}(\mathcal{H}(x_2(t), x_1(t)), \eta_t \nabla f_2(x_2(t))) = x_2(t+1)$.

algorithm (3), using history information to get an accurate estimation of the average gradient $g(t)$ which is a better descent direction than $\nabla f_i(x_i(t))$. Similar ideas of using history information trace back to [39], in which the previous gradient is used to narrow down the possible values of the current gradient to reduce communication complexity for a two-agent optimization problem.

A recent paper [22] proposes an algorithm that achieves convergence results similar to our algorithm. The algorithm in [22] can be interpreted as adding an integration type correction term to (5) while using a fixed step size. This correction term also involves history information in a certain way, which is consistent with our impossibility result. Our algorithm and [22] are similar in the sense that they are both dynamical systems with order $2nN$, and take difference of gradients as inputs. But they are different in the sense that they are dynamical systems with different parameters, which result in different behaviors. The differences between our algorithm and [22] are summarized below. Firstly, in our algorithm, the state $s_i(t)$ serves as an estimator of the average gradient and can be used as a stopping criterion, like in many centralized methods where the norm of gradients is used as a stopping criterion. Secondly, in our algorithm, the update rule can be clearly separated into two parts, the first part being the update corresponding to centralized gradient descent, and the second part being the gradient estimator. With the separation, our algorithm can be easily extended to other centralized methods (see also Remark 1). Thirdly, the two consensus matrices in [22] need to be symmetric and also satisfy a predefined spectral relationship, whereas our algorithm has a looser requirement on the consensus matrices. Fourthly, without assuming the strong convexity, [22] achieves a $O(\frac{1}{t})$ convergence rate in terms of the optimality residuals, which can be loosely defined as $\|\nabla f(x_i(t))\|^2$ and $\|x_i(t) - \bar{x}(t)\|^2$. Our algorithm not only achieves $O(\frac{1}{t})$ for the optimality residuals, but also achieves $O(\frac{1}{t})$ in terms of the objective error $f(\hat{x}_i(t)) - f^*$, which is a more direct measure of optimality. At last, one downside of our current results is that [22] gives a step size bound that only depends on β, whereas our step size bounds also depend on W (Lemma 2 and Theorem 3).

IV. Convergence Analysis

In this section, we prove our main convergence results Theorem 1, Lemma 2, and Theorem 3.

A. Analysis Setup

We first stack the $x_i(t)$, $s_i(t)$ and $\nabla f_i(x_i(t))$ in (2) and (3) into matrices. Define $x(t), s(t), \nabla(t) \in \mathbb{R}^{n \times N}$ as,[6]

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}, s(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_n(t) \end{bmatrix}, \nabla(t) = \begin{bmatrix} \nabla f_1(x_1(t)) \\ \nabla f_2(x_2(t)) \\ \vdots \\ \nabla f_n(x_n(t)) \end{bmatrix}.$$

We can compactly write the update rule in (2) and (3) as

$$x(t+1) = Wx(t) - \eta s(t) \quad (9a)$$
$$s(t+1) = Ws(t) + \nabla(t+1) - \nabla(t) \quad (9b)$$

and also $s(0) = \nabla(0)$. We start by introducing two straightforward lemmas. Lemma 7 derives update equations that govern the average sequence $\bar{x}(t)$ and $\bar{s}(t)$. Lemma 8 gives several auxiliary inequalities. Lemma 7 is a direct consequence of the fact W is doubly stochastic and $s(0) = \nabla(0)$, and Lemma 8 is a direct consequence of the β-smoothness of f_i. The proofs of the two lemmas are postponed to Appendix-A.

Lemma 7. *The following equalities hold.*

(a) $\bar{s}(t+1) = \bar{s}(t) + g(t+1) - g(t) = g(t+1)$
(b) $\bar{x}(t+1) = \bar{x}(t) - \eta\bar{s}(t) = \bar{x}(t) - \eta g(t)$

Lemma 8. *Under Assumption 1, the following inequalities hold.*

(a) $\|\nabla(t) - \nabla(t-1)\| \leq \beta\|x(t) - x(t-1)\|$
(b) $\|g(t) - g(t-1)\| \leq \beta\frac{1}{\sqrt{n}}\|x(t) - x(t-1)\|$
(c) $\|g(t) - h(t)\| \leq \beta\frac{1}{\sqrt{n}}\|x(t) - \mathbf{1}\bar{x}(t)\|$

B. Why the Algorithm Works: An Intuitive Explanation

We provide our intuition that partially explains why the algorithm (9) can achieve a linear convergence rate for strongly convex and smooth functions. In fact we can prove the following proposition.

Proposition 9. *The following is true.*

- *Assuming $\|s(t) - \mathbf{1}g(t)\|$ decays at a linear rate, then $\|x(t) - \mathbf{1}x^*\|$ also decays at a linear rate.*
- *Assuming $\|x(t) - \mathbf{1}x^*\|$ decays at a linear rate, then $\|s(t) - \mathbf{1}g(t)\|$ also decays at a linear rate.*

The proof of the above proposition is postponed to Appendix-B. The above proposition tells that the linear decaying rates of the gradient estimation error $\|s(t) - \mathbf{1}g(t)\|$ and the distance to optimizer $\|x(t) - \mathbf{1}x^*\|$ imply each other. Though this circular argument does not prove the linear convergence rate of our algorithm, it illustrates how the algorithm works: the gradient descent step (9a) and the gradient estimation step (9b) facilitate each other to converge fast in a reciprocal manner. This mutual dependence is distinct from many previous methods, where one usually bounds the consensus error at first, and then use the consensus error to bound the objective error, and there is no mutual dependence between the two. In the next two subsections, we will rigorously prove the convergence.

C. Convergence Analysis: Strongly Convex

We start by introducing a lemma that is adapted from standard optimization literature, e.g. [37]. Lemma 10 states that if we perform a gradient descent step with a fixed step size for a strongly convex and smooth function, then the distance to optimizer shrinks by at least a fixed ratio. For completeness we give a proof of Lemma 10 in Appendix-C.

Lemma 10. *$\forall x \in \mathbb{R}^N$, define $x^+ = x - \eta\nabla f(x)$ where $0 < \eta < \frac{2}{\beta}$ and f is α-strongly convex and β-smooth, then*

$$\|x^+ - x^*\| \leq \lambda\|x - x^*\|$$

where $\lambda = \max(|1 - \eta\alpha|, |1 - \eta\beta|)$.

Now we are ready to prove Theorem 1.

[6] In section II and III, x and $x(t)$ have been used as centralized decision variables. Here we abuse the use of notation $x(t)$ without causing any confusion.

Proof of Theorem 1: Our strategy is to bound $\|s(k)-\mathbf{1}g(k)\|$, $\|x(k)-\mathbf{1}\bar{x}(k)\|$, and $\|\bar{x}(k)-x^*\|$ in terms of linear combinations of their past values, and in this way obtain a linear system inequality, which will imply linear convergence.

Step 1: Bound $\|s(k)-\mathbf{1}g(k)\|$. By the update rule (9b),

$$\begin{aligned} s(k)-\mathbf{1}g(k) &= [Ws(k-1)-\mathbf{1}g(k-1)] \\ &\quad + [\nabla(k)-\nabla(k-1)] - [\mathbf{1}g(k)-\mathbf{1}g(k-1)]. \end{aligned}$$

Taking the norm, noticing that the column-wise average of $s(k-1)$ is just $g(k-1)$ by Lemma 7(a), and using the averaging property of the consensus matrix W, we have

$$\begin{aligned} &\|s(k)-\mathbf{1}g(k)\| \\ &\le \|Ws(k-1)-\mathbf{1}g(k-1)\| \\ &\quad + \Big\|[\nabla(k)-\nabla(k-1)] - [\mathbf{1}g(k)-\mathbf{1}g(k-1)]\Big\| \\ &\le \sigma\|s(k-1)-\mathbf{1}g(k-1)\| \\ &\quad + \Big\|[\nabla(k)-\nabla(k-1)] - [\mathbf{1}g(k)-\mathbf{1}g(k-1)]\Big\|. \qquad (10) \end{aligned}$$

It is easy to verify

$$\begin{aligned} &\Big\|[\nabla(k)-\nabla(k-1)] - [\mathbf{1}g(k)-\mathbf{1}g(k-1)]\Big\|^2 \\ &= \|\nabla(k)-\nabla(k-1)\|^2 + n\|g(k)-g(k-1)\|^2 \\ &\quad - 2\sum_{i=1}^{n}\langle \nabla f_i(x_i(k)) - \nabla f_i(x_i(k-1)), g(k)-g(k-1)\rangle \\ &= \|\nabla(k)-\nabla(k-1)\|^2 + n\|g(k)-g(k-1)\|^2 \\ &\quad - 2\langle ng(k)-ng(k-1), g(k)-g(k-1)\rangle \\ &\le \|\nabla(k)-\nabla(k-1)\|^2. \end{aligned}$$

Combining this with (10) and using Lemma 8 (a), we get

$$\begin{aligned} &\|s(k)-\mathbf{1}g(k)\| \\ &\le \sigma\|s(k-1)-\mathbf{1}g(k-1)\| + \beta\|x(k)-x(k-1)\|. \qquad (11) \end{aligned}$$

Step 2: Bound $\|x(k)-\mathbf{1}\bar{x}(k)\|$. Considering update rule (9a) and using Lemma 7(b) and the property of W, we have

$$\begin{aligned} \|x(k)-\mathbf{1}\bar{x}(k)\| &\le \sigma\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| \\ &\quad + \eta\|s(k-1)-\mathbf{1}g(k-1)\|. \qquad (12) \end{aligned}$$

Step 3: Bound $\|\bar{x}(k)-x^*\|$. Notice by Lemma 7(b), the update rule for $\bar{x}(k)$ is

$$\bar{x}(k) = \bar{x}(k-1) - \eta h(k-1) - \eta[g(k-1)-h(k-1)].$$

Since the gradient of f at $\bar{x}(k-1)$ is $h(k-1)$, therefore, by Lemma 10 and Lemma 8(c), we have

$$\begin{aligned} &\|\bar{x}(k)-x^*\| \\ &\le \lambda\|\bar{x}(k-1)-x^*\| + \eta\|g(k-1)-h(k-1)\| \\ &\le \lambda\|\bar{x}(k-1)-x^*\| + \eta\frac{\beta}{\sqrt{n}}\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| \qquad (13) \end{aligned}$$

where $\lambda = \max(|1-\eta\alpha|, |1-\eta\beta|)$.

Step 4: Bound $\|x(k)-x(k-1)\|$. Notice that by Assumption 1,

$$\|h(k-1)\| = \|\nabla f(\bar{x}(k-1))\| \le \beta\|\bar{x}(k-1)-x^*\|.$$

Combining the above and Lemma 8(c), we have

$$\begin{aligned} &\|s(k-1)\| \\ &\le \|s(k-1)-\mathbf{1}g(k-1)\| \\ &\quad + \|\mathbf{1}g(k-1)-\mathbf{1}h(k-1)\| + \|\mathbf{1}h(k-1)\| \\ &\le \|s(k-1)-\mathbf{1}g(k-1)\| + \beta\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| \\ &\quad + \beta\sqrt{n}\|\bar{x}(k-1)-x^*\|. \end{aligned}$$

Hence

$$\begin{aligned} &\|x(k)-x(k-1)\| \\ &= \|Wx(k-1)-x(k-1)-\eta s(k-1)\| \\ &= \|(W-I)(x(k-1)-\mathbf{1}\bar{x}(k-1)) - \eta s(k-1)\| \\ &\le 2\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| + \eta\|s(k-1)\| \\ &\le \eta\|s(k-1)-\mathbf{1}g(k-1)\| \\ &+ (\eta\beta+2)\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| + \eta\beta\sqrt{n}\|\bar{x}(k-1)-x^*\|. \end{aligned} \qquad (14)$$

Step 5: Derive a linear system inequality. We combine the previous four steps into a big linear system inequality. Plugging (14) into (11), we have

$$\begin{aligned} \|s(k)-\mathbf{1}g(k)\| &\le (\sigma+\beta\eta)\|s(k-1)-\mathbf{1}g(k-1)\| \\ &\quad + \beta(\eta\beta+2)\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| \\ &\quad + \eta\beta^2\sqrt{n}\|\bar{x}(k-1)-x^*\|. \qquad (15) \end{aligned}$$

Combining (15), (12) and (13), we get

$$\begin{aligned} &\overbrace{\begin{bmatrix} \|s(k)-\mathbf{1}g(k)\| \\ \|x(k)-\mathbf{1}\bar{x}(k)\| \\ \sqrt{n}\|\bar{x}(k)-x^*\| \end{bmatrix}}^{\triangleq z(k)\in\mathbb{R}^3} \le \overbrace{\begin{bmatrix} (\sigma+\beta\eta) & \beta(\eta\beta+2) & \eta\beta^2 \\ \eta & \sigma & 0 \\ 0 & \eta\beta & \lambda \end{bmatrix}}^{\triangleq G(\eta)\in\mathbb{R}^{3\times 3}} \\ &\qquad \cdot \overbrace{\begin{bmatrix} \|s(k-1)-\mathbf{1}g(k-1)\| \\ \|x(k-1)-\mathbf{1}\bar{x}(k-1)\| \\ \sqrt{n}\|\bar{x}(k-1)-x^*\| \end{bmatrix}}^{\triangleq z(k-1)\in\mathbb{R}^3} \qquad (16) \end{aligned}$$

where '$\le$' means element wise less than or equal to. Since $z(k)$ and $G(\eta)$ have nonnegative entries, we can actually expand (16) recursively, and get

$$z(k) \le G(\eta)^k z(0).$$

Since $G(\eta)$ has nonnegative entries and $G(\eta)^2$ has all positive entries, by Theorem 8.5.1 and 8.5.2 of [40], each entry of $G(\eta)^k$ will be $O(\rho(G(\eta))^k)$. Hence, the three entries of $z(k)$, $\|s(k)-\mathbf{1}g(k)\|$, $\|x(k)-\mathbf{1}\bar{x}(k)\|$, and $\|\bar{x}(k)-x^*\|$ will converge to 0 in the order of $\rho(G(\eta))^k$. By β-smoothness of f, we have

$$f(\bar{x}(k)) \le f^* + \langle \nabla f(x^*), \bar{x}(k)-x^*\rangle + \frac{\beta}{2}\|\bar{x}(k)-x^*\|^2.$$

Since $\nabla f(x^*) = 0$, the above implies that $f(\bar{x}(k)) - f^* = O(\rho(G(\eta))^{2k})$. Again by β-smoothness and Cauchy-Schwarz inequality,

$$\begin{aligned} &f(x_i(k)) \\ &\le f(\bar{x}(k)) + \langle \nabla f(\bar{x}(k)), x_i(k)-\bar{x}(k)\rangle + \frac{\beta}{2}\|x_i(k)-\bar{x}(k)\|^2 \\ &\le f(\bar{x}(k)) + \frac{1}{2\beta}\|\nabla f(\bar{x}(k))\|^2 + \beta\|x_i(k)-\bar{x}(k)\|^2. \end{aligned}$$

Since $\|\nabla f(\bar{x}(k))\| = \|\nabla f(\bar{x}(k)) - \nabla f(x^*)\| \le \beta\|\bar{x}(k) - x^*\| = O(\rho(G(\eta))^k)$, and $\|x_i(k)-\bar{x}(k)\| \le \|x(k)-\mathbf{1}\bar{x}(k)\| = O(\rho(G(\eta))^k)$, the above inequality implies that $f(x_i(k)) -$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2017.2698261, IEEE Transactions on Control of Network Systems

$f(\bar{x}(k)) = O(\rho(G(\eta))^{2k})$. This further leads to $f(x_i(k)) - f^* = f(x_i(k)) - f(\bar{x}(k)) + f(\bar{x}(k)) - f^* = O(\rho(G(\eta))^{2k})$. □

We now prove Lemma 2.

Proof of Lemma 2: Since $\eta < \frac{1}{\beta}$, it is easy to check $1-\alpha\eta \geq 1-\beta\eta > 0$, and hence $\lambda = 1-\alpha\eta$. We first write down the charasteristic polynomial $p(\zeta)$ of $G(\eta)$,

$$p(\zeta) = p_0(\zeta)[\zeta - (1-\alpha\eta)] - \eta^3\beta^3$$

where $p_0(\zeta) = (\zeta-\sigma-\eta\beta)(\zeta-\sigma)-\eta\beta(\eta\beta+2)$. The two roots of p_0, ζ_1 and ζ_2 are $\frac{2\sigma+\eta\beta\pm\sqrt{5\eta^2\beta^2+8\eta\beta}}{2}$. Since $0 < \eta\beta < 1$, both roots are real numbers less than $\sigma + 3\sqrt{\eta\beta}$. This implies

$$\begin{aligned} p_0(\zeta) &= (\zeta-\zeta_1)(\zeta-\zeta_2) \\ &\geq (\zeta-\sigma-3\sqrt{\eta\beta})^2 \text{ when } \zeta > \sigma+3\sqrt{\eta\beta}. \end{aligned} \tag{17}$$

Let $\zeta^* = \max(1-\frac{\alpha\eta}{2}, \sigma + 5\sqrt{\eta\beta}\sqrt{\frac{\beta}{\alpha}}) > \sigma + 3\sqrt{\eta\beta}$, then

$$\begin{aligned} p(\zeta^*) &\geq \frac{\alpha\eta}{2}(\zeta^*-\sigma-3\sqrt{\eta\beta})^2 - \eta^3\beta^3 \\ &\geq \frac{\alpha\eta}{2}\eta\beta\frac{4\beta}{\alpha} - \eta^3\beta^3 \geq 0. \end{aligned}$$

Since $p(\zeta)$ is a strictly increasing function on $[\max(1-\alpha\eta, \sigma+3\sqrt{\eta\beta}), +\infty)$ (this interval includes ζ^*), $p(\zeta)$ does not have real roots on (ζ^*, ∞). Since $G(\eta)$ is a nonnegative matrix, by Perron-Frobenius Theorem (Page 503, Theorem 8.3.1 of [40]), $\rho(G(\eta))$ is an eigenvalue of $G(\eta)$. Hence $\rho(G(\eta))$ is a real root of $p(\zeta)$, so we have $\rho(G(\eta)) \leq \zeta^*$. □

Remark 4. *We now comment on how β-smoothness of f_i is used in the proof of Theorem 1 and not used in the proof of DGD-like algorithms, e.g. [9], and how this difference would affect the convergence rates of the two algorithms. In DGD-like algorithms, (sub)gradients are usually assumed to be bounded. Whenever a (sub)gradient is encountered in the proof, it is replaced by its bound and the resulting inequalities usually involve many additive constant terms. To control the constant terms, a vanishing step size is required, which slows down the convergence. In the proof of Theorem 1, whenever gradients appear, they appear in the form of the difference of two gradients (like (10)). Therefore we can bound it using the β-smoothness assumption. The resulting inequalities (like (11)) do not involve constant terms, but linear combinations of some variables instead. After carefully arranging these inequalities, we can get a contraction inequality (16) and hence the linear convergence rate.*

D. Convergence Analysis: Non-strongly Convex Case

Proof of Theorem 3: The proof will be divided into 4 steps. In step 1, we derive a linear system inequality (18) similar to (16), but this time with input. In step 2, we use the linear system inequality (18) to bound the consensus error. In step 3, we show that $g(t)$, is actually an inexact gradient [41] of f at $\bar{x}(t)$ with the inexactness being characterized by the consensus error. Therefore, the update equation for the average sequence $\bar{x}(t)$ (Lemma 7(b)) is essentially inexact gradient descent. In step 4, we apply the analysis method for CGD to the average sequence $\bar{x}(t)$ and show that the $O(\frac{1}{t})$ convergence rate is preserved in spite of the inexactness.

Step 1: A linear system inequality. We prove the following inequality,

$$\begin{aligned} \overbrace{\begin{bmatrix} \|s(k)-\mathbf{1}g(k)\| \\ \|x(k)-\mathbf{1}\bar{x}(k)\| \end{bmatrix}}^{\triangleq \tilde{z}(k)\in\mathbb{R}^2} \leq \overbrace{\begin{bmatrix} (\sigma+\beta\eta) & 2\beta \\ \eta & \sigma \end{bmatrix}}^{\triangleq \tilde{G}(\eta)\in\mathbb{R}^{2\times 2}} \\ \cdot \overbrace{\begin{bmatrix} \|s(k-1)-\mathbf{1}g(k-1)\| \\ \|x(k-1)-\mathbf{1}\bar{x}(k-1)\| \end{bmatrix}}^{\triangleq \tilde{z}(k-1)\in\mathbb{R}^2} \\ + \overbrace{\begin{bmatrix} \eta\beta\sqrt{n}\|g(k-1)\| \\ 0 \end{bmatrix}}^{\triangleq \tilde{b}(k-1)}. \end{aligned} \tag{18}$$

It is easy to check that (11) and (12) (copied below as (19) and (20)) still holds if we remove the strongly convex assumption.

$$\begin{aligned} \|s(k)-\mathbf{1}g(k)\| \leq \sigma\|s(k-1)-\mathbf{1}g(k-1)\| \\ + \beta\|x(k)-x(k-1)\| \end{aligned} \tag{19}$$

$$\begin{aligned} \|x(k)-\mathbf{1}\bar{x}(k)\| \leq \eta\|s(k-1)-\mathbf{1}g(k-1)\| \\ + \sigma\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| \end{aligned} \tag{20}$$

Notice we have

$$\|s(k-1)\| \leq \|s(k-1)-\mathbf{1}g(k-1)\| + \|\mathbf{1}g(k-1)\|. \tag{21}$$

Also notice

$$\begin{aligned} &\|x(k)-x(k-1)\| \\ &= \|Wx(k-1)-x(k-1)-\eta s(k-1)\| \\ &= \|(W-I)(x(k-1)-\mathbf{1}\bar{x}(k-1))-\eta s(k-1)\| \\ &\leq 2\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| + \eta\|s(k-1)\|. \end{aligned} \tag{22}$$

Combining (19), (21) and (22) yields

$$\begin{aligned} &\|s(k)-\mathbf{1}g(k)\| \\ &\leq (\sigma+\eta\beta)\|s(k-1)-\mathbf{1}g(k-1)\| \\ &\quad + 2\beta\|x(k-1)-\mathbf{1}\bar{x}(k-1)\| + \eta\beta\sqrt{n}\|g(k-1)\|. \end{aligned}$$

Combining the above and (20) yields (18).

Step 2: Consensus error. We prove that

$$\|x(k)-\mathbf{1}\bar{x}(k)\| \leq A_1\theta^k + A_2\sum_{\ell=0}^{k-1}\theta^{k-1-\ell}\|g(\ell)\| \tag{23}$$

where A_1, A_2 and θ are defined as follows.

$$A_1 = \frac{1}{\beta}\|s(0)-\mathbf{1}g(0)\| + 2\|x(0)-\mathbf{1}\bar{x}(0)\|$$

$$A_2 = \eta\sqrt{n},\ \theta = \frac{1+\sigma}{2}$$

To prove (23), we first notice that by (18), we have

$$\tilde{z}(k) \leq \tilde{G}(\eta)^k\tilde{z}(0) + \sum_{\ell=0}^{k-1}\tilde{G}(\eta)^{k-1-\ell}\tilde{b}(\ell). \tag{24}$$

The two eigenvalues of $\tilde{G}(\eta)$ are

$$\frac{2\sigma+\eta\beta\pm\sqrt{\eta^2\beta^2+8\eta\beta}}{2}.$$

Then since $\eta^2\beta^2 < \eta\beta < \sqrt{\eta\beta}$, we have $\sigma < \rho(\tilde{G}(\eta)) < \sigma + 2\sqrt{\eta\beta} \le \sigma + 2\sqrt{\frac{(1-\sigma)^2}{160}} < \sigma + \frac{1-\sigma}{2} = \theta$. Therefore the entries of $\tilde{G}(\eta)^k$ decay with rate $O(\theta^k)$, and one can expect an inequality like (23) to hold. To get the exact form of (23) we need to do careful calculations, which are postponed to Appendix-D .

Step 3: $g(t)$ is an inexact gradient of f at $\bar{x}(t)$. We show that, $\forall t$, $\exists \hat{f}_t \in \mathbb{R}$ s.t. $\forall \omega \in \mathbb{R}^N$, we have

$$f(\omega) \ge \hat{f}_t + \langle g(t), \omega - \bar{x}(t)\rangle \tag{25}$$

$$\begin{aligned} f(\omega) \le \hat{f}_t + \langle g(t), \omega - \bar{x}(t)\rangle + \beta\|\omega - \bar{x}(t)\|^2 \\ + \frac{\beta}{n}\|x(t) - \mathbf{1}\bar{x}(t)\|^2. \end{aligned} \tag{26}$$

To prove (25) and (26), we define

$$\hat{f}_t = \frac{1}{n}\sum_{i=1}^{n}\big[f_i(x_i(t)) + \langle \nabla f_i(x_i(t)), \bar{x}(t) - x_i(t)\rangle\big].$$

Then, for any $\omega \in \mathbb{R}^N$, we have

$$\begin{aligned} f(\omega) &= \frac{1}{n}\sum_{i=1}^{n} f_i(\omega) \\ &\ge \frac{1}{n}\sum_{i=1}^{n}\big[f_i(x_i(t)) + \langle \nabla f_i(x_i(t)), \omega - x_i(t)\big] \\ &= \frac{1}{n}\sum_{i=1}^{n}\big[f_i(x_i(t)) + \langle \nabla f_i(x_i(t)), \bar{x}(t) - x_i(t)\big] \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}\langle \nabla f_i(x_i(t)), \omega - \bar{x}(t)\rangle \\ &= \hat{f}_t + \langle g(t), \omega - \bar{x}(t)\rangle \end{aligned}$$

which shows (25). For (26), similarly,

$$\begin{aligned} f(\omega) &\le \frac{1}{n}\sum_{i=1}^{n}\big[f_i(x_i(t)) + \langle \nabla f_i(x_i(t)), \omega - x_i(t) \\ &\quad + \frac{\beta}{2}\|\omega - x_i(t)\|^2\big] \\ &= \frac{1}{n}\sum_{i=1}^{n}\big[f_i(x_i(t)) + \langle \nabla f_i(x_i(t)), \bar{x}(t) - x_i(t)\big] \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}\langle \nabla f_i(x_i(t)), \omega - \bar{x}(t)\rangle \\ &\quad + \frac{\beta}{2}\frac{1}{n}\sum_{i=1}^{n}\|\omega - \bar{x}(t)\|^2 \\ &= \hat{f}_t + \langle g(t), \omega - \bar{x}(t)\rangle \\ &\quad + \frac{\beta}{2}\frac{1}{n}\sum_{i=1}^{n}\|(\omega - \bar{x}(t)) + (\bar{x}(t) - x_i(t))\|^2 \\ &\le \hat{f}_t + \langle g(t), \omega - \bar{x}(t)\rangle + \beta\|\omega - \bar{x}(t)\|^2 \\ &\quad + \beta\frac{1}{n}\sum_{i=1}^{n}\|\bar{x}(t) - x_i(t)\|^2 \\ &= \hat{f}_t + \langle g(t), \omega - \bar{x}(t)\rangle + \beta\|\omega - \bar{x}(t)\|^2 \\ &\quad + \frac{\beta}{n}\|x(t) - \mathbf{1}\bar{x}(t)\|^2 \end{aligned}$$

where in the second inequality we have used the elementary fact that $\|u+v\|^2 \le 2\|u\|^2 + 2\|v\|^2$ for all $u, v \in \mathbb{R}^N$.

Step 4: Follow the proof of CGD. Define $r_k = \|\bar{x}(k) - x^*\|^2$. Then

$$\begin{aligned} r_k &= \|\bar{x}(k+1) - x^* - (\bar{x}(k+1) - \bar{x}(k))\|^2 \\ &= r_{k+1} - 2\langle \bar{x}(k+1) - \bar{x}(k), \bar{x}(k+1) - x^*\rangle \\ &\qquad + \|\bar{x}(k+1) - \bar{x}(k)\|^2 \\ &\overset{(a)}{=} r_{k+1} + 2\eta\langle g(k), \bar{x}(k+1) - x^*\rangle + \eta^2\|g(k)\|^2 \\ &= r_{k+1} + 2\eta\langle g(k), \bar{x}(k) - x^*\rangle \\ &\qquad + 2\eta\big[\langle g(k), \bar{x}(k+1) - \bar{x}(k)\rangle + \frac{\eta}{2}\|g(k)\|^2\big] \\ &\overset{(b)}{\ge} r_{k+1} + 2\eta(\hat{f}_k - f^*) + 2\eta\Big[f(\bar{x}(k+1)) - \hat{f}_k \\ &\qquad + (\frac{\eta}{2} - \eta^2\beta)\|g(k)\|^2 - \frac{\beta}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2\Big] \\ &= r_{k+1} + 2\eta(f(\bar{x}(k+1)) - f^*) \\ &\qquad + 2\eta\big[(\frac{\eta}{2} - \eta^2\beta)\|g(k)\|^2 - \frac{\beta}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2\big] \end{aligned} \tag{27}$$

where in (a) we have used $\bar{x}(k+1) - \bar{x}(k) = -\eta g(k)$. In (b) we have used two inequalities. The first one is $\langle g(k), \bar{x}(k) - x^*\rangle \ge \hat{f}_k - f^*$ (by (25) with $\omega = x^*$), and the second one is

$$\begin{aligned} &\langle g(k), \bar{x}(k+1) - \bar{x}(k)\rangle \\ &\ge f(\bar{x}(k+1)) - \hat{f}_k - \beta\|\bar{x}(k+1) - \bar{x}(k)\|^2 \\ &\quad - \frac{\beta}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \\ &= f(\bar{x}(k+1)) - \hat{f}_k - \beta\eta^2\|g(k)\|^2 - \frac{\beta}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \end{aligned}$$

which follows from (26) (with $\omega = \bar{x}(k+1)$) and the fact $\bar{x}(k+1) - \bar{x}(k) = -\eta g(k)$. Summing up (27) for $k = 0, \ldots, t$, we get

$$\begin{aligned} \sum_{k=0}^{t}[f(\bar{x}(k+1)) - f^*] \le \frac{r_0}{2\eta} + \sum_{k=0}^{t}\Big[\frac{\beta}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \\ - (\frac{\eta}{2} - \eta^2\beta)\|g(k)\|^2\Big]. \end{aligned} \tag{28}$$

Now, by (26),

$$\begin{aligned} f(x_i(k)) &\le \hat{f}_k + \langle g(k), x_i(k) - \bar{x}(k)\rangle + \beta\|x_i(k) - \bar{x}(k)\|^2 \\ &\quad + \frac{\beta}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \\ &\le f(\bar{x}(k)) + \langle g(k), x_i(k) - \bar{x}(k)\rangle \\ &\quad + \beta\|x_i(k) - \bar{x}(k)\|^2 + \frac{\beta}{n}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \end{aligned}$$

where in the second inequality we have used $\hat{f}_k \le f(\bar{x}(k))$, which can be derived from (25) by letting $\omega = \bar{x}(k)$. Hence,

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{t}[f(x_i(k+1)) - f^*] \\ &\le \sum_{k=0}^{t}[f(\bar{x}(k+1)) - f^*] + \frac{2\beta}{n}\sum_{k=0}^{t+1}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \end{aligned}$$

$$\leq \frac{r_0}{2\eta} + \frac{3\beta}{n}\sum_{k=0}^{t+1}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 + (\eta^2\beta - \frac{\eta}{2})\sum_{k=0}^{t}\|g(k)\|^2 \tag{29}$$

where in the last inequality we have used (28). Now we try to bound $\sum_{k=0}^{t}\|x(k) - \mathbf{1}\bar{x}(k)\|^2$. Fix t, define vector $\mu = [A_1, A_2\|g(0)\|, \ldots, A_2\|g(t-1)\|]^T \in \mathbb{R}^{t+1}$, $\chi_k = [\theta^k, \theta^{k-1}, \theta^{k-2}, \ldots, \theta, 1, 0, \ldots, 0]^T \in \mathbb{R}^{t+1}$, then (23) can be rewritten as

$$\|x(k) - \mathbf{1}\bar{x}(k)\| \leq \chi_k^T\mu.$$

And hence

$$\sum_{k=0}^{t}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \leq \mu^T X \mu$$

where $X \triangleq \sum_{k=0}^{t}\chi_k\chi_k^T \in \mathbb{R}^{(t+1)\times(t+1)}$. It can be easily seen that X is a symmetric and positive semi-definite matrix. Let X's (p,q)th element be X_{pq}, then for $1 \leq p \leq q \leq t+1$, $X_{pq} = \sum_{k=q-1}^{t}\theta^{k+1-p}\theta^{k+1-q} = \theta^{q-p}\frac{1-\theta^{2(t+2-q)}}{1-\theta^2}$. Now we calculate the absolute row sum of the p'th row of X, getting

$$\begin{aligned}\sum_{q=1}^{t+1}|X_{pq}| &= \sum_{q=p}^{t+1}|X_{pq}| + \sum_{q=1}^{p-1}|X_{pq}| \\ &\leq \frac{1}{1-\theta^2}\sum_{q=p}^{t+1}\theta^{q-p} + \frac{1}{1-\theta^2}\sum_{q=1}^{p-1}\theta^{p-q} \\ &= \frac{1}{1-\theta^2}\frac{1-\theta^{t+2-p}}{1-\theta} + \frac{1}{1-\theta^2}\frac{\theta(1-\theta^{p-1})}{1-\theta} \\ &< \frac{3}{(1-\theta)^2}.\end{aligned}$$

By Gershgorin Circle Theorem [42], this shows that $\rho(X) \leq \frac{3}{(1-\theta)^2}$. Since $\mu^T X\mu \leq \rho(X)\|\mu\|^2$, we get

$$\begin{aligned}\sum_{k=0}^{t}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 &\leq \frac{3}{(1-\theta)^2}\|\mu\|^2 \\ &\leq \frac{3}{(1-\theta)^2}\left[A_1^2 + A_2^2\sum_{k=0}^{t-1}\|g(k)\|^2\right].\end{aligned} \tag{30}$$

Combining this with (29), and plugging in the value of A_2, we get

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{t}[f(x_i(k+1)) - f^*] \\ &\leq \frac{r_0}{2\eta} + \frac{9\beta A_1^2}{n(1-\theta)^2} + (\frac{10\beta\eta^2}{(1-\theta)^2} - \frac{\eta}{2})\sum_{k=0}^{t}\|g(k)\|^2 \\ &\leq \frac{r_0}{2\eta} + \frac{9\beta A_1^2}{n(1-\theta)^2}\end{aligned} \tag{31}$$

where in the last inequality, we have used

$$\frac{10\beta\eta^2}{(1-\theta)^2} - \frac{\eta}{2} = \eta(\frac{40\eta\beta}{(1-\sigma)^2} - \frac{1}{2}) \leq -\frac{1}{4}\eta < 0 \tag{32}$$

which follows from $\theta = (1+\sigma)/2$, and the step size rule $0 < \eta \leq \frac{(1-\sigma)^2}{160\beta}$. Recall that $\hat{x}_i(t+1) = \frac{1}{t+1}\sum_{k=1}^{t+1}x_i(k)$, then by convexity of f we have $f(\hat{x}_i(t+1)) \leq \frac{1}{t+1}\sum_{k=0}^{t}f(x_i(k+1))$. Combining this with (31) leads to

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}[f(\hat{x}_i(t+1)) - f^*] \\ &\leq \frac{1}{t+1}\left\{\frac{r_0}{2\eta} + \frac{9\beta A_1^2}{n(1-\theta)^2}\right\} \\ &= \frac{1}{t+1}\left\{\frac{\|\bar{x}(0) - x^*\|^2}{2\eta}\right. \\ &\left.+ \frac{36\beta}{(1-\sigma)^2}\left[\frac{1}{\beta\sqrt{n}}\|s(0) - \mathbf{1}g(0)\| + \frac{2}{\sqrt{n}}\|x(0) - \mathbf{1}\bar{x}(0)\|\right]^2\right\}\end{aligned}$$

which gives part (a) of the Theorem. For part (b), we consider the second inequality in (31). Notice the left hand side of (31) is nonnegative, and $\frac{10\beta\eta^2}{(1-\theta)^2} - \frac{\eta}{2} \leq -\frac{1}{4}\eta$ (by (32)). We have,

$$\sum_{k=0}^{t}\|g(k)\|^2 \leq (\frac{r_0}{2\eta} + \frac{9\beta A_1^2}{n(1-\theta)^2})\frac{4}{\eta}.$$

Combining the above with (30), we get

$$\begin{aligned}&\sum_{k=0}^{t}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \\ &\leq \frac{3}{(1-\theta)^2}\left[A_1^2 + A_2^2(\frac{r_0}{2\eta} + \frac{9\beta A_1^2}{n(1-\theta)^2})\frac{4}{\eta}\right] \\ &\leq \frac{1740}{(1-\sigma)^4}\left[\frac{1}{\beta}\|s(0) - \mathbf{1}g(0)\| + 2\|x(0) - \mathbf{1}\bar{x}(0)\|\right]^2 \\ &\qquad + \frac{24}{(1-\sigma)^2}\|\mathbf{1}\bar{x}(0) - \mathbf{1}x^*\|^2.\end{aligned} \tag{33}$$

Also notice that $\sum_{k=0}^{t}\|x(k) - \mathbf{1}\bar{x}(k)\|^2 \geq t\min_{0\leq k\leq t}\|x(k) - \mathbf{1}\bar{x}(k)\|^2$. Combining this with (33) leads to part (b). □

V. Numerical Experiments

A. Experiments with different objective functions

We simulate our algorithm on different objective functions and compare it with other algorithms. We choose $n = 100$ agents and the graph is generated using the Erdos-Renyi model [43] with connectivity probability 0.3.[7] The weight matrix W is chosen using the Laplacian method [22]. In details, $W = I - \frac{1}{\max_{i=1}^n d_i + 1}L$, where d_i is degree of node i in the graph $\mathcal{G}$, and $L = [L_{ij}]$ is the Laplacian of the graph defined to be $L_{ij} = -1$ for $(i,j) \in E$, and $L_{ii} = d_i$ and $L_{ij} = 0$ for i,j not connected. The algorithms we compare include DGD (5) with a vanishing step size and with a fixed step size, the algorithm proposed in [22] (with $\tilde{W} = \frac{W+I}{2}$), and CGD with a fixed step size. Each element of the initial point $x_i(0)$ is drawn from i.i.d. Gaussian with mean 0 and variance 25. For the functions f_i, we consider three cases.

Case I: The functions f_i are square losses for linear regression, i.e. $f_i(x) = \sum_{m=1}^{M_i}(\langle u_{im}, x\rangle - v_{im})^2$ where $u_{im} \in \mathbb{R}^N$ are the features and $v_{im} \in \mathbb{R}$ are the observed outputs, and $\{(u_{im}, v_{im})\}_{m=1}^{M_i}$ are $M_i = 20$ data samples for agent i. We generate each data sample independently. We first fix a predefined parameter $\tilde{x} \in \mathbb{R}^N$ with each element drown uniformly from $[0,1]$. For each sample (u_{im}, v_{im}), the last

[7] We discard the graphs that are not connected.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2017.2698261, IEEE Transactions on Control of Network Systems

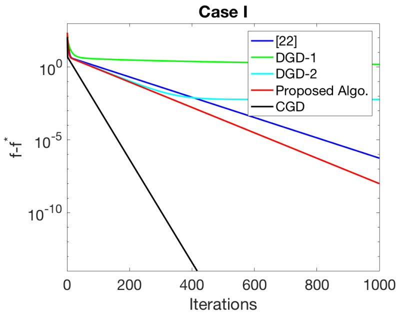


Fig. 1: Simulation results for case I. Green (shown as 'DGD-1') is DGD (5) with vanishing step size; cyan ('DGD-2') is DGD (5) with fixed step size; blue ('[22]') is the algorithm in [22]; red ('Proposed Algo.') is our algorithm; black ('CGD') is CGD.

element of u_{im} is fixed to be 1, and the rest elements are drawn from i.i.d. Gaussian with mean 0 and variance 25. Then we generate $v_{im} = \langle \tilde{x}, u_{im} \rangle + \epsilon_{im}$ where ϵ_{im} are independent Gaussian noises with mean 0 and variance 1.

Case II: The functions f_i are the loss functions for logistic regression [44], i.e. $f_i(x) = \sum_{m=1}^{M_i} \left[\ln(1 + e^{\langle u_{im}, x \rangle}) - v_{im} \langle u_{im}, x \rangle \right]$ where $u_{im} \in \mathbb{R}^N$ are the features and $v_{im} \in \{0, 1\}$ are the observed labels, and $\{(u_{im}, v_{im})\}_{m=1}^{M_i}$ are $M_i = 20$ data samples for agent i. The data samples are generated independently. We first fix a predefined parameter $\tilde{x} \in \mathbb{R}^N$ with each element drown uniformly from $[0, 1]$. For each sample (u_{im}, v_{im}), the last element of u_{im} is fixed to be 1, and the rest elements are drawn from i.i.d. Gaussian with mean 0 and variance 25. We then generate v_{im} to be 1 from a Bernoulli distribution, with probability of $v_{im} = 1$ being $\frac{1}{1+e^{-\langle \tilde{x}, u_{im} \rangle}}$.

Case III: The functions f_i are smooth and convex but $\nabla^2 f$ is zero at the optimum x^*. In details, we choose $N = 1$ and $\forall x \in \mathbb{R}$, $f_i(x) = u(x) + b_i x$, where b_i is randomly chosen that satisfies $\sum_i b_i = 0$, and $u(x) = \frac{1}{4} x^4$ for $|x| \leq 1$, and $u(x) = |x| - \frac{3}{4}$ for $|x| > 1$.

Case I and case II satisfy Assumption 1 and 2, while case III only satisfies Assumption 1. In case I and II, we plot the average objective error, i.e. $\frac{1}{n} \sum_i f(x_i(t)) - f^*$.[8] Case III is intended to test the sublinear convergence rate $\frac{1}{t}$ of the algorithm (Theorem 3), therefore in addition to the average objective error, we also plot $t \times (\frac{1}{n} \sum_i f(x_i(t)) - f^*)$ to check if the objective error decays as $O(\frac{1}{t})$. The results are shown in Figure 1, 2 and 3.

B. Experiments with different graph sizes

As pointed out by Remark 3, the convergence rates of our algorithm depend on σ (not directly on n). Therefore for graphs with different sizes n but similar σ, our algorithm should have roughly the same convergence rate, and therefore is 'scale-free'. This section will test this property through simulation. We choose random 3-regular graphs with sizes $n = 50, 100, 150, \ldots, 500$.[9] We obtain W by the Laplacian method. It is known that with a high probability, a random regular graph is a regular expander graph (see Section 7.3.2 of [47]), and thus with a high probability, σ is free of the size n of the graph (see Corollary 1(d) and Lemma 4 of [9]). Therefore, our algorithm should be 'scale-free' for those graphs. We choose the objective functions using the same method as Case I in the previous subsection. For each graph size, we list the parameter σ along with the strongly-convex parameter α, the smooth parameter β in Table I. Each element of $x_i(0)$ is drawn from i.i.d. Gaussion distribution with mean 0 and variance 25. We plot the number of iterations it take for the average objective error $(\frac{1}{n} \sum_i f(x_i(t)) - f^*)$ to reach a predefined error level 1×10^{-10} versus the graph size n. The results are shown in Figure 4.

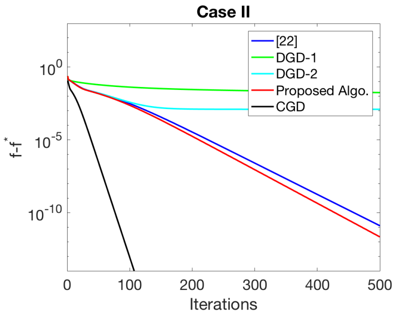


Fig. 2: Simulation results for case II. The meanings of the legends are the same as in Figure 1.

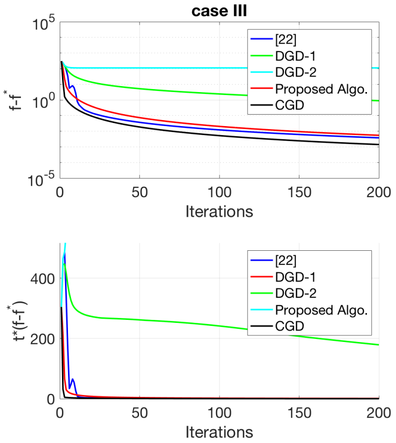


Fig. 3: Simulation results for case III. Upper: objective error. Lower: t times objective error. The meanings of the legends are the same as in Figure 1.

VI. Conclusion

In this paper, we have proposed a method that can effectively harness smoothness to speed up distributed optimization. The method features a gradient estimation scheme. It achieves

[8] For case I and III, there are closed form expressions for f^* and we compute f^* using the closed form expressions. For case II, we compute f^* using centralized gradient descent until the gradient reaches the smallest value (almost 0) that MATLAB can handle.

[9] A 3-regular graph is a graph in which each node is adjacent to 3 other nodes. We generate the random 3-regular graphs using the method in [45]. To ensure the connectivity of the generated graphs, we discard the graphs that are not connected, which happen very rarely (see Theorem 2.10 of [46]).

TABLE I: Simulation parameters.

Graph size	α	β	σ
$n = 50$	1.0000	26.23	0.9150
$n = 100$	0.9989	25.52	0.9488
$n = 150$	0.9995	25.25	0.9503
$n = 200$	0.9995	25.22	0.9544
$n = 250$	0.9997	24.34	0.9429
$n = 300$	0.9996	24.56	0.9492
$n = 350$	0.9997	25.25	0.9566
$n = 400$	0.9999	25.23	0.9512
$n = 450$	0.9999	24.70	0.9530
$n = 500$	0.9998	24.97	0.9526

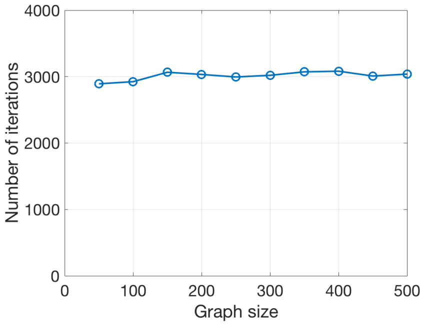


Fig. 4: Simulation results for different graph sizes. The x-axis is the graph size, the y-axis is the number of iterations needed to bring down the average objective error to 1×10^{-10}.

a $O(\frac{1}{t})$ convergence rate when the objective function is convex and smooth, and achieves a linear convergence rate when the function is strongly convex and smooth. Both rates are comparable to the centralized gradient methods except for some constants. Future work includes applying the gradient estimation scheme to other first order optimization algorithms, like Nesterov gradient descent.

References

[1] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," in *2016 55th IEEE Conference on Decision and Control*. IEEE, 2016.

[2] B. Johansson, "On distributed optimization in networked systems," 2008.

[3] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1847–1862, 2010.

[4] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, no. May, pp. 1663–1707, 2010.

[5] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," in *1984 American Control Conference*, 1984, pp. 484–489.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

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

[8] I. Lobel and A. Ozdaglar, "Convergence analysis of distributed subgradient methods over random networks," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 353–360.

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

[10] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[11] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *arXiv preprint arXiv:1406.2075*, 2014.

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

[13] I. Matei and J. S. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 754–771, 2011.

[14] A. Olshevsky, "Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control," *arXiv preprint arXiv:1411.4186*, 2014.

[15] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 151–164, 2012.

[16] I. Lobel, A. Ozdaglar, and D. Feijer, "Distributed multi-agent optimization with state-dependent communication," *Mathematical Programming*, vol. 129, no. 2, pp. 255–284, 2011.

[17] I.-A. Chen *et al.*, "Fast distributed first-order methods," Master's thesis, Massachusetts Institute of Technology, 2012.

[18] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *arXiv preprint arXiv:1310.7063*, 2013.

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

[20] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[21] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 2012, pp. 601–608.

[22] 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, 2015.

[23] G. Qu and N. Li, "Accelerated distributed nesterov gradient descent for smooth and strongly convex functions," in *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*. IEEE, 2016.

[24] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 2055–2060.

[25] P. Di Lorenzo and G. Scutari, "Distributed nonconvex optimization over networks," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*. IEEE, 2015, pp. 229–232.

[26] 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.

[27] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *arXiv preprint arXiv:1607.03218*, 2016.

[28] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe, "Geometrically convergent distributed optimization with uncoordinated step-sizes," *arXiv preprint arXiv:1609.05877*, 2016.

[29] C. Xi and U. A. Khan, "Add-opt: Accelerated distributed directed optimization," *arXiv preprint arXiv:1607.04757*, 2016.

[30] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "A decentralized second-order method with exact linear convergence rate for consensus optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 507–522, 2016.

[31] D. Bajovic, D. Jakovetic, N. Krejic, and N. K. Jerinkic, "Newton-like method with diagonal correction for distributed optimization," *arXiv preprint arXiv:1509.01703*, 2015.

[32] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-newton methods," *arXiv preprint arXiv:1605.00933*, 2016.

[33] 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.

[34] E. Wei and A. Ozdaglar, "On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 551–554.

[35] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[36] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[37] S. Bubeck, "Convex optimization: Algorithms and complexity," *arXiv preprint arXiv:1405.4980*, 2014.
[38] D. P. Bertsekas, "Nonlinear programming," 1999.
[39] J. N. Tsitsiklis and Z.-Q. Luo, "Communication complexity of convex optimization," *Journal of Complexity*, vol. 3, no. 3, pp. 231–243, 1987.
[40] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.
[41] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Mathematical Programming*, vol. 146, no. 1-2, pp. 37–75, 2014.
[42] S. A. Gershgorin, "Uber die abgrenzung der eigenwerte einer matrix," *Bulletin de l'Academie des Sciences de l'URSS. Classe des sciences mathematiques et na*, no. 6, pp. 749–754, 1931.
[43] P. Erdos and A. Renyi, "On random graphs i," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.
[44] (2012) Logistic regression. [Online]. Available: http://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf
[45] J. H. Kim and V. H. Vu, "Generating random regular graphs," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. ACM, 2003, pp. 213–222.
[46] N. C. Wormald, "Models of random regular graphs," in *London Mathematical Society Lecture Note Series*, 1999, pp. 239–298.
[47] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.

Appendix

A. Proof of Lemma 7 and Lemma 8

Proof of Lemma 7: Since W is doubly stochastic, we have $\mathbf{1}^T W = \mathbf{1}^T$. Therefore,

$$\begin{aligned}\bar{x}(t+1) &= \frac{1}{n}\mathbf{1}^T(Wx(t) - \eta s(t)) \\ &= \bar{x}(t) - \eta\bar{s}(t).\end{aligned}$$

Similarly,

$$\begin{aligned}\bar{s}(t+1) &= \frac{1}{n}\mathbf{1}^T[Ws(t) + \nabla(t+1) - \nabla(t)] \\ &= \bar{s}(t) + g(t+1) - g(t).\end{aligned}$$

Do this recursively, we can get $\bar{s}(t+1) = \bar{s}(0) + g(t+1) - g(0)$. Since $s(0) = \nabla(0)$, we have $\bar{s}(0) = g(0)$. This finishes the proof. □

Proof of Lemma 8: (a)

$$\begin{aligned}\|\nabla(t) - \nabla(t-1)\| &= \sqrt{\sum_{i=1}^n \|\nabla f_i(x_i(t)) - \nabla f_i(x_i(t-1))\|^2} \\ &\le \sqrt{\sum_{i=1}^n \beta^2 \|x_i(t) - x_i(t-1)\|^2} \\ &= \beta\|x(t) - x(t-1)\|.\end{aligned}$$

(b)

$$\begin{aligned}\|g(t) - g(t-1)\| &= \|\sum_{i=1}^n \frac{\nabla f_i(x_i(t)) - \nabla f_i(x_i(t-1))}{n}\| \\ &\le \beta\sum_{i=1}^n \frac{\|x_i(t) - x_i(t-1)\|}{n} \\ &\le \beta\sqrt{\sum_{i=1}^n \frac{\|x_i(t) - x_i(t-1)\|^2}{n}} \\ &= \beta\frac{1}{\sqrt{n}}\|x(t) - x(t-1)\|.\end{aligned}$$

(c)

$$\begin{aligned}\|g(t) - h(t)\| &= \|\sum_{i=1}^n \frac{\nabla f_i(x_i(t)) - \nabla f_i(\bar{x}(t))}{n}\| \\ &\le \beta\sum_{i=1}^n \frac{\|x_i(t) - \bar{x}(t)\|}{n} \\ &\le \beta\sqrt{\sum_{i=1}^n \frac{\|x_i(t) - \bar{x}(t)\|^2}{n}} \\ &= \beta\frac{1}{\sqrt{n}}\|x(t) - \mathbf{1}\bar{x}(t)\|.\end{aligned}$$

□

B. Proof of Proposition 9

Proof of Proposition 9: On one hand, assume the gradient estimation error $\|s(t) - \mathbf{1}g(t)\|$ decays at a linear rate, i.e. $\|s(t) - \mathbf{1}g(t)\| \le C_1\kappa_1^t$ for some constant $C_1 > 0$ and $\kappa_1 \in (0,1)$. Then, the consensus error satisfy

$$\begin{aligned}\|x(t) - \mathbf{1}\bar{x}(t)\| &\le \|Wx(t-1) - \mathbf{1}\bar{x}(t-1)\| \\ &\quad + \eta\|s(t-1) - \mathbf{1}g(t-1)\| \\ &\le \sigma\|x(t-1) - \mathbf{1}\bar{x}(t-1)\| + \eta C_1\kappa_1^{t-1} \\ &\le \sigma^t\|x(0) - \mathbf{1}\bar{x}(0)\| + \eta C_1\sum_{k=0}^{t-1}\sigma^k\kappa_1^{t-1-k} \\ &= \sigma^t\|x(0) - \mathbf{1}\bar{x}(0)\| + \eta C_1\frac{\sigma^t - \kappa_1^t}{\sigma - \kappa_1}\end{aligned}$$

where in the first inequality we have used Lemma 7(b), and in the second inequality we have used the averaging property of W. Therefore, the consensus error also decays at a linear rate. Then, by Lemma 8 (c), $\|g(t) - h(t)\|$ also decays at a linear rate, i.e. we have $\|g(t) - h(t)\| \le C_2\kappa_2^t$ for some $C_2 > 0$ and $\kappa_2 \in (0,1)$. By Lemma 7(b), $\bar{x}(t) = \bar{x}(t-1) - \eta h(t-1) - \eta(g(t-1) - h(t-1))$. Since $h(t-1) = \nabla f(\bar{x}(t-1))$, $\bar{x}(t-1) - \eta h(t-1)$ is a standard gradient step for function f. Since f is strongly convex and smooth, a standard gradient descent step shrinks the distance to the minimizer by a least a fixed ratio (see Lemma 10), hence we have

$$\|\bar{x}(t-1) - \eta h(t-1) - x^*\| \le \lambda\|\bar{x}(t-1) - x^*\|$$

for some $\lambda \in (0,1)$. Hence,

$$\begin{aligned}\|\bar{x}(t) - x^*\| &\le \lambda\|\bar{x}(t-1) - x^*\| + \eta\|h(t-1) - g(t-1)\| \\ &\le \lambda\|\bar{x}(t-1) - x^*\| + \eta C_2\kappa_2^{t-1} \\ &\le \lambda^t\|\bar{x}(0) - \mathbf{1}x^*\| + \eta C_2\sum_{k=0}^{t-1}\lambda^{t-1-k}\kappa_2^k \\ &= \lambda^t\|\bar{x}(0) - x^*\| + \eta C_2\frac{\lambda^t - \kappa_2^t}{\lambda - \kappa_2}.\end{aligned}$$

Therefore $\|\bar{x}(t) - x^*\|$, the distance of the average $\bar{x}(t)$ to the minimizer decays at a linear rate. Combining this with the fact that the consensus error $\|x(t) - \mathbf{1}\bar{x}(t)\|$ decays at a linear rate and using the triangle inequality, we have $\|x(t) - \mathbf{1}x^*\|$, the distance to the optimizer, decays at a linear rate.

On the other hand, assume $\|x(t) - \mathbf{1}x^*\|$ decays at a linear rate, then $\|x(t) - x(t-1)\|$ also decays at a linear rate, i.e. $\|x(t) - x(t-1)\| \le C_3\kappa_3^{t-1}$ for some $C_3 > 0$ and $\kappa_3 \in (0,1)$. Then,

$$\begin{aligned}&\|s(t) - \mathbf{1}g(t)\| \\ &\le \|Ws(t-1) - \mathbf{1}g(t-1)\| + \|\nabla(t) - \nabla(t-1)\| \\ &\quad + \|\mathbf{1}g(t) - \mathbf{1}g(t-1)\| \\ &\le \sigma\|s(t-1) - \mathbf{1}g(t-1)\| + 2\beta C_3\kappa_3^{t-1} \\ &\le \sigma^t\|s(0) - \mathbf{1}g(0)\| + 2\beta C_3\frac{\kappa_3^t - \sigma^t}{\kappa_3 - \sigma}\end{aligned}$$

where in the second inequality we have used Lemma 8(a)(b). Hence the gradient estimation error $\|s(t)-\mathbf{1}g(t)\|$ decays at a linear rate. □

C. Proof of Lemma 10

In the proof of Lemma 10 we will use the following result, which is the same as Lemma 3.11 of [37] in which the proof can be found.

Lemma 11. *Let $f:\mathbb{R}^N\to\mathbb{R}$ be α-strongly convex and β-smooth, $\forall x,y\in\mathbb{R}^N$, we have*

$$\begin{aligned}&\langle\nabla f(x)-\nabla f(y),x-y\rangle\\ &\geq\frac{\alpha\beta}{\alpha+\beta}\|x-y\|^2+\frac{1}{\alpha+\beta}\|\nabla f(x)-\nabla f(y)\|^2.\end{aligned}$$

As a special case, let $y=x^*$ be the unique optimizer of f. Since $\nabla f(x^*)=0$, we have

$$\langle\nabla f(x),x-x^*\rangle\geq\frac{\alpha\beta}{\alpha+\beta}\|x-x^*\|^2+\frac{1}{\alpha+\beta}\|\nabla f(x)\|^2.$$

Now we proceed to prove Lemma 10.

Proof of Lemma 10: If $0<\eta\leq\frac{2}{\alpha+\beta}$, then $\frac{2}{\eta}-\alpha\geq\beta$. Let $\alpha'=\alpha$, $\beta'=\frac{2}{\eta}-\alpha\geq\beta$, then f is also α'-strongly convex and β'-smooth. Then, we have

$$\begin{aligned}&\|x-x^*-\eta\nabla f(x)\|^2\\ &=\|x-x^*\|^2-2\eta\langle\nabla f(x),x-x^*\rangle+\eta^2\|\nabla f(x)\|^2\\ &\leq(1-2\eta\frac{\alpha'\beta'}{\alpha'+\beta'})\|x-x^*\|^2+(\eta^2-2\eta\frac{1}{\alpha'+\beta'})\|\nabla f(x(t))\|^2\\ &=(1-\alpha\eta)^2\|x-x^*\|^2\\ &=\lambda^2\|x-x^*\|^2\end{aligned}$$

where the first inequality is due to Lemma 11 and the last equality is due to $|1-\alpha\eta|\geq|1-\beta\eta|$, which follows from $\alpha<\beta$ and $0<\eta\leq\frac{2}{\alpha+\beta}$. The case $\frac{2}{\beta}>\eta>\frac{2}{\alpha+\beta}$ follows from a similar argument (but with $\alpha'=\frac{2}{\eta}-\beta$ and $\beta'=\beta$) and the details are omitted. □

D. Derivation of (23)

We first diagonalize $\tilde{G}(\eta)$ as $\tilde{G}(\eta)=V\Lambda V^{-1}$, where

$$\Lambda=\begin{bmatrix}\theta_1 & 0\\ 0 & \theta_2\end{bmatrix}$$

with $\theta_1=\frac{2\sigma+\eta\beta-\sqrt{\eta^2\beta^2+8\eta\beta}}{2}$ and $\theta_2=\frac{2\sigma+\eta\beta+\sqrt{\eta^2\beta^2+8\eta\beta}}{2}$. Matrix V and V^{-1} are given by

$$V=\begin{bmatrix}\frac{\beta\sqrt{\eta}-\sqrt{\beta}\sqrt{8+\eta\beta}}{2\sqrt{\eta}} & \frac{\beta\sqrt{\eta}+\sqrt{\beta}\sqrt{8+\eta\beta}}{2\sqrt{\eta}}\\ 1 & 1\end{bmatrix}$$

$$V^{-1}=\begin{bmatrix}-\frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}} & \frac{1}{2}+\frac{1}{2}\frac{\sqrt{\eta\beta}}{\sqrt{8+\eta\beta}}\\ \frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}} & \frac{1}{2}-\frac{1}{2}\frac{\sqrt{\eta\beta}}{\sqrt{8+\eta\beta}}\end{bmatrix}.$$

Therefore, $\forall p,\ell\in\mathbb{N}$,

$$\begin{aligned}\tilde{G}(\eta)^p\tilde{b}(\ell)&=V\Lambda^p\begin{bmatrix}-\frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}}\\ \frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}}\end{bmatrix}\eta\beta\sqrt{n}\|g(\ell)\|\\ &=V\begin{bmatrix}-\frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}}\theta_1^p\\ \frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}}\theta_2^p\end{bmatrix}\eta\beta\sqrt{n}\|g(\ell)\|.\end{aligned}$$

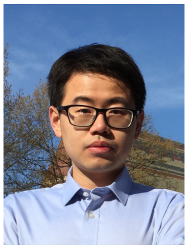
Therefore, the second row of $\tilde{G}(\eta)^p\tilde{b}(\ell)$ is given by

$$\frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}}\eta\beta\sqrt{n}\|g(\ell)\|(\theta_2^p-\theta_1^p)\leq\eta\sqrt{n}\|g(\ell)\|\theta^p \quad (34)$$

where we have used the fact that both $|\theta_1|$, $|\theta_2|$ are upper bounded by θ, and the fact $\eta<\frac{1}{\beta}$. Similarly, we compute the second row of $\tilde{G}(\eta)^k\tilde{z}(0)$, and get

$$\begin{aligned}&-\theta_1^k\frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}}\|s(0)-\mathbf{1}g(0)\|\\ &\quad+\theta_1^k(\frac{1}{2}+\frac{1}{2}\frac{\sqrt{\eta\beta}}{\sqrt{8+\eta\beta}})\|x(0)-\mathbf{1}\bar{x}(0)\|\\ &\quad+\theta_2^k\frac{\sqrt{\eta}}{\sqrt{\beta}\sqrt{8+\eta\beta}}\|s(0)-\mathbf{1}g(0)\|\\ &\quad+\theta_2^k(\frac{1}{2}-\frac{1}{2}\frac{\sqrt{\eta\beta}}{\sqrt{8+\eta\beta}})\|x(0)-\mathbf{1}\bar{x}(0)\|\\ &\leq\theta^k\left\{\frac{1}{\beta}\|s(0)-\mathbf{1}g(0)\|+2\|x(0)-\bar{x}(0)\|\right\} \quad (35)\end{aligned}$$

where we have used $\sqrt{\frac{\eta}{\beta}}<\frac{1}{\beta}$, and $\max(|\theta_1|,|\theta_2|)\leq\theta$. Notice that $\|x(k)-\mathbf{1}\bar{x}(k)\|$ is the second row of $\tilde{z}(k)$. Then combining (34) and (35) with (24) yields (23).



Guannan Qu (S'14) received his B.S. degree in Electrical Engineering from Tsinghua University in Beijing, China in 2014. Since 2014 he has been a graduate student in the School of Engineering and Applied Sciences at Harvard University. His research interest lies in network control and optimization.

Na Li (M'09) received her B.S. degree in Mathematics and Applied Mathematics from Zhejiang University in China and her PhD degree in Control and Dynamical systems from the California Institute of Technology in 2013.

She is an Assistant Professor in the School of Engineering and Applied Sciences in Harvard University. She was a postdoctoral associate of the Laboratory for Information and Decision Systems at Massachusetts Institute of Technology. She was a Best Student Paper Award finalist in the 2011 IEEE Conference on Decision and Control. Her research lies in the design, analysis, optimization and control of distributed network systems, with particular applications to power networks and systems biology/physiology.