Convergence Rates of Distributed Nesterov-Like Gradient Methods on Random Networks

Dušan Jakovetić, *Member, IEEE*, João Manuel Freitas Xavier, *Member, IEEE*, and José M. F. Moura, *Fellow, IEEE*

***Abstract*—We consider distributed optimization in random networks where N nodes cooperatively minimize the sum $\sum_{i=1}^{N} f_i(x)$ of their individual convex costs. Existing literature proposes distributed gradient-like methods that are computationally cheap and resilient to link failures, but have slow convergence rates. In this paper, we propose accelerated distributed gradient methods that 1) are resilient to link failures; 2) computationally cheap; and 3) improve convergence rates over other gradient methods. We model the network by a sequence of independent, identically distributed random matrices $\{W(k)\}$ drawn from the set of symmetric, stochastic matrices with positive diagonals. The network is connected on average and the cost functions are convex, differentiable, with Lipschitz continuous and bounded gradients. We design two distributed Nesterov-like gradient methods that modify the D–NG and D–NC methods that we proposed for static networks. We prove their convergence rates in terms of the expected optimality gap at the cost function. Let k and $\mathcal{K}$ be the number of per-node gradient evaluations and per-node communications, respectively. Then the modified D–NG achieves rates $O(\log k/k)$ and $O(\log \mathcal{K}/\mathcal{K})$, and the modified D–NC rates $O(1/k^2)$ and $O(1/\mathcal{K}^{2-\xi})$, where $\xi > 0$ is arbitrarily small. For comparison, the standard distributed gradient method cannot do better than $\Omega(1/k^{2/3})$ and $\Omega(1/\mathcal{K}^{2/3})$, on the same class of cost functions (even for static networks). Simulation examples illustrate our analytical findings.**

***Index Terms*—Consensus, convergence rate, distributed optimization, Nesterov gradient, random networks.**

I. Introduction

A. Motivation

With many distributed signal processing applications, a common research challenge is to develop distributed algorithms whereby all nodes in a generic, connected network recover a *global* vector parameter $x^\star$ of common interest, while each node possesses only a partial, *local* knowledge on the unknown vector $x^\star$ and interacts only *locally*, with immediate neighbors in the network. For example, such situation arises with (distance-measurement based) acoustic source localization in wireless sensor networks (WSNs). Node i measures the received signal energy that contains information only about its distance to the acoustic source, and hence node i in isolation cannot recover the unknown source location; but, it can recover the source location by collaborating with other nodes in the network (see Example 3 below for details). In many applications, like with WSNs, the inter-node communications are prone to random communication failures (e.g., random packet dropouts in WSNs); an important challenge in developing distributed algorithms to recover $x^\star$ is to make them provably resilient to random communication failures.

Similarly to, e.g., [1]–[3], we address the above problem in the framework of distributed (smooth) optimization. Each node i in a generic, connected network has a differentiable, convex cost function $f_i(x; d_i)$ known only by node i, parameterized by node i's local data d_i, with $x \in \mathbb{R}^d$ the global optimization variable common to all nodes. Each node in the network wants to find a parameter $x^\star \in \mathbb{R}^d$ that minimizes the sum of the nodes' local costs $f_i(x) := f_i(x; d_i)$:

$$\text{minimize} \sum_{i=1}^{N} f_i(x) =: f(x), \tag{1}$$

where we assume that each f_i has Lipschitz continuous and bounded gradients. In this paper, our goals are: 1) to develop distributed, iterative, gradient-based methods that solve (1), whereby nodes over iterations exchange messages only with their immediate neighbors; and 2) to provide convergence rate guarantees of the methods (on the assumed functions class) in the presence of random communication failures.

We now motivate setup (1) with three application examples from the literature, namely 1) distributed learning of a linear classifier, 2) distributed robust estimation, and 3) distributed source localization. Each of the three example problems obeys the Assumptions that we make on the f_i's (see ahead Assumptions 3, 4, and 5 for details.) Besides those, existing literature provides many other application examples, including distributed estimation, e.g., [4]–[6], distributed detection, e.g., [7], [8], target localization and intruder detection in biological networks, e.g., [9], and spectrum sensing for cognitive radio networks, e.g., [10].

1) Example 1: Distributed Learning of a Linear Classifier: Consider a distributed learning scenario where training data is distributed across nodes in the network; each node has N_s data samples, $\{a_{ij}, b_{ij}\}_{j=1}^{N_s}$, where $a_{ij} \in \mathbb{R}^m$ is a feature vector and

Manuscript received May 11, 2013; revised August 14, 2013; accepted October 03, 2013. Date of publication November 14, 2013; date of current version January 20, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shuguang (Robert) Cui. The work of D. Jakovetić and J. M. F. Xavier was supported by Carnegie Mellon/Portugal Program under a grant from the Fundação de Ciência e Tecnologia (FCT) from Portugal; this work was supported by FCT grants CMU-PT/SIA/0026/2009 and SFRH/BD/33518/2008 (through the Carnegie Mellon\Portugal Program managed by ICTI); by ISR/IST pluriannual funding (POSC program, FEDER); the work of D. Jakovetić and J. M. F. Moura was supported by an AFOSR grant FA95501010291 and by an NSF grant CCF1011903, while D. Jakovetić was a doctoral and postdoctoral student within the Carnegie Mellon/Portugal Program.

D. Jakovetić is with the University of Novi Sad, BioSense Center, 21000 Novi Sad, Serbia (e-mail: djakovet@uns.ac.rs).

J. M. F. Xavier is with the SIPG—Signal and Image Processing Group, Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST), University of Lisboa, 1049-001 Lisbon, Portugal (e-mail: jxavier@isr.ist.utl.pt).

J. M. F. Moura is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: moura@ece.cmu.edu).

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

$b_{ij} \in \{-1,+1\}$ is the class label of the vector a_{ij}, e.g., [11]. For the purpose of future feature vector classifications, each node wants to learn the linear classifier $a \mapsto \text{sign}\left(a^\top x' + x''\right)$, i.e., to determine a vector $x' \in \mathbb{R}^m$ and a scalar $x'' \in \mathbb{R}$, based on all nodes' data samples, that yields the best classification in a certain sense. Specifically, we seek $x' \in \mathbb{R}^m$ and $x'' \in \mathbb{R}$ that solve:

$$\text{minimize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N_s} \phi\left(-b_{ij}(a_{ij}^\top x' + x'')\right), \tag{2}$$

where $\phi(z) = \log(1+e^z)$ is the logistic loss. Problem (2) fits (1), with $x = ((x')^\top, x'')^\top$, and $f_i(x) := \sum_{j=1}^{N_s} \phi\left(-b_{ij}(a_{ij}^\top x' + x'')\right)$.

2) Example 2: Distributed Robust Estimation in Sensor Networks: Consider a sensor network deployed, e.g., to measure a pollution level in a certain area [12]. Each sensor i makes N_s scalar measurements (of the level of pollution), $\{a_{ij}\}_{j=1}^{N_s}$. Assume a signal+noise measurement model, where $a_{ij} = \theta + \nu_{ij}$, with θ the "signal" (real pollution level), and ν_{ij} a zero-mean noise, independent across all indices i, j. Further, suppose that there are two groups of sensors, $\mathcal{J}_1$ and $\mathcal{J}_2$. Sensors $i \in \mathcal{J}_1$ operate correctly, and their measurements have a small variance $\sigma_1^2 = \text{Var}(\nu_{ij}), \forall i \in \mathcal{J}_1, \forall j = 1,\ldots,N_s$; sensors $i = \mathcal{J}_2$ are damaged, and their measurements have a large variance $\sigma_2^2 = \text{Var}(\nu_{ij}), \forall i \in \mathcal{J}_2, \forall j = 1,\ldots,N_s$. To combat the outlier measurements from damaged sensors in $\mathcal{J}_2$, [12] estimates the parameter θ through the Huber loss, i.e., it obtains an estimate $\widehat{\theta}$ as a solution to the following problem:

$$\text{minimize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N_s} h(x; a_{ij}), \tag{3}$$

where $h(\cdot; a)$ is the Huber loss: $h(z;a) = \frac{1}{2}\|z-a\|^2$, if $\|z-a\| \le 1$, and $h(z;a) = \|z-a\| - 1/2$, otherwise.

3) Example 3: Acoustic Source Localization in Sensor Networks: Suppose that an acoustic source is positioned at an unknown location $\theta \in \mathbb{R}^2$ in the field [12], [13]. Each node (sensor) i measures the received signal energy from the source:

$$y_i = \frac{c_1}{\|\theta - r_i\|^{c_2}} + \zeta_i.$$

Here $r_i \in \mathbb{R}^2$ is node i's location (known to node i), $c_1 > 0$ and $c_2 > 0$ are constants known to all nodes, and ζ_i is zero-mean additive noise. The goal is for each node to estimate the source's position θ. Reference [13] proposes to obtain an estimate of θ by solving:

$$\text{minimize} \quad \sum_{i=1}^{N} \text{dist}\left(x, C_i\right), \tag{4}$$

where C_i is the disk

$$C_i = \left\{ x \in \mathbb{R}^2 : \|x - r_i\| \le \delta_i := \left(\frac{c_1}{y_i}\right)^{1/c_2} \right\},$$

and $\text{dist}(x, C) = \inf_{y \in C} \|x - y\|$ is the distance from x to the set C. It can be shown that $\text{dist}\left(x, C_i\right) = \max\left\{\|x - r_i\| - \delta_i,\, 0\right\}$, which is convex but nonsmooth. To adapt the latter function to our setting, we take standard smooth approximations of the involved nonsmooth functions. Namely, we approximate $z \in \mathbb{R} \mapsto \max\{z, 0\}$ with $z \mapsto g_m(z) = \frac{1}{\gamma}\log(1 + e^{\gamma z})$ and $z \in \mathbb{R}^d \mapsto \|z\|$ with $z \mapsto g_n(z) = \sqrt{z^\top z + \epsilon^2}$, where $\gamma > 0$ is a large scalar and $\epsilon > 0$ is a small scalar. The resulting optimization problem takes the form of (1) with $f_i(x) = g_m\left(g_n\left(x - r_i\right) - \delta_i\right)$.

B. Contributions

We now state our main contributions by placing them in the context of existing work. For problem (1), [3], see also [14], [15], presents two distributed Nesterov-like gradient algorithms for static (non-random) networks, referred to as D–NG (Distributed Nesterov Gradient algorithm) and D–NC (Distributed Nesterov gradient with Consensus iterations).

In this paper, we propose the mD–NG and mD–NC algorithms, which modify the D–NG and D–NC algorithms, and, beyond proving their convergence, we solve the much harder problem of establishing their convergence rate guarantees on *random networks*. We model the network by a sequence of random independent, identically distributed (i.i.d.) weight matrices $W(k)$ drawn from a set of symmetric, stochastic matrices with positive diagonals, and we assume that the network is connected on average (the graph supporting $\mathbb{E}\left[W(k)\right]$ is connected). We establish the convergence rates of the expected optimality gap in the cost function (at any node i) of mD-NG and mD-NC, in terms of the number of per node gradient evaluations k and the number of per-node communications $\mathcal{K}$, when the functions f_i are convex and differentiable, with Lipschitz continuous and bounded gradients. We show that the modified methods achieve *in expectation* the same rates that the methods in [3] achieve on static networks, namely: mD–NG converges at rates $O(\log k/k)$ and $O(\log \mathcal{K}/\mathcal{K})$, while mD–NC has rates $O(1/k^2)$ and $O(1/\mathcal{K}^{2-\xi})$, where ξ is an arbitrarily small positive number. We explicitly give the convergence rate constants in terms of the number of nodes N and the network statistics, more precisely, in terms of the quantity $\overline{\mu} := \left(\|\mathbb{E}[W(k)^2] - J\|\right)^{1/2}$ (See ahead paragraph with heading Notation.)

We contrast D–NG and D–NC in [3] with their modified variants, mD–NG and mD–NC, respectively. Simulations in Section VI show that D–NG may diverge when links fail, while mD–NG converges, possibly at a slightly lower rate on static networks and requires an additional (d-dimensional) vector communication per iteration k. Hence, mD–NG compromises slightly speed of convergence for robustness to link failures.

Algorithm mD–NC has one inner consensus with $2d$-dimensional variables per outer iteration k, while D–NC has two consensus algorithms with d-dimensional variables. Both D–NC variants converge in our simulations when links fail, showing similar performance.

The analysis here differs from [3], since the dynamics of disagreements are different from the dynamics in [3]. This requires novel bounds on certain products of time-varying matrices. By disagreement, we mean how different the solution estimates of distinct nodes are, say $x_i(k)$ and $x_j(k)$ for nodes i and j.

C. Brief Comment on the Literature

There is increased interest in distributed optimization and learning. Broadly, the literature considers two types of methods, namely, batch processing, e.g., [1], [10], [16]–[19], and online adaptive processing, e.g., [4], [9], [20], [21]. With batch processing, data is acquired beforehand, and hence the f_i's are known before the algorithm runs. In contrast, with adaptive online processing, nodes acquire new data at each iteration k of the distributed algorithm. In general, adaptive and batch processing show inherent tradeoffs. We comment on certain advantages and disadvantages of the two methods. When the unknown parameter $x^\star = x^\star(k)$ is time-varying, and the time constant of the dynamics of $x^\star(k)$ is comparable with the time needed to perform one distributed algorithm's iteration, adaptive processing is the right choice. When the dynamics of $x^\star(k)$ are slow compared to the time needed to iterate the distributed algorithm until convergence, or when $x^\star(k)$ does not vary with time (the case we consider here), both batch and adaptive processing can be applied. To our best knowledge, existing literature does not address comparisons of distributed adaptive and distributed batch optimization algorithms. A systematic comparison among the two types of distributed methods is a nontrivial task and is outside of our paper's scope. In the centralized setting, [22] addresses a similar problem in machine learning, by comparing the standard gradient method (batch processing) and the stochastic gradient method (adaptive/online processing). Their results roughly show that, when the number of data samples is small enough and the allowed computational cost to perform optimization is large enough, it is advantageous to use batch processing (standard gradient method); on the other hand, if the number of data samples is large enough and the allowed computational cost is small enough, it is advantageous to use adaptive processing (online gradient method). We consider in this paper batch processing.

Distributed *gradient methods* are, e.g., in [1], [2], [12], [16]–[18], [23]–[27]. References [1], [16], [24] proved convergence of their algorithms under deterministically time varying or random networks. Typically, f_i's are convex, non-differentiable, and with bounded gradients over the constraint set. Reference [2] establishes $O\left(\log k/\sqrt{k}\right)$ convergence rate (with high probability) of a version of the distributed dual averaging method. We assume a more restricted class $\mathcal{F}$ of cost functions–f_i's that are convex and have Lipschitz continuous and bounded gradients, but, in contradistinction, we establish strictly faster convergence rates–at least $O(\log k/k)$ that are not achievable by standard distributed gradient methods [1] on the same class $\mathcal{F}$. Indeed, [3] shows that the method in [1] cannot achieve a worst-case rate better than $\Omega\left(1/k^{2/3}\right)$ on the same class $\mathcal{F}$, even for static networks. Reference [28] proposes an accelerated distributed proximal gradient method, which resembles our D–NC method for *deterministically* time varying networks; in contrast, we deal here with *randomly varying networks*. For a detailed comparison of D–NC with [28], we refer to [3].

In addition to distributed gradient-like methods, a different type of methods – distributed (augmented) Lagrangian and distributed alternating direction of multipliers methods (ADMM) have been studied, e.g., in [10], [29]–[35]. They have in general more complex iterations than gradient methods, but may have a lower total communication cost, e.g., [30]. [34] shows convergence of an asynchronous ADMM algorithm while [35] shows an $O(1/\mathcal{K})$ rate (in expectation) of an asynchronous ADMM method studied therein.

In summary, our paper differs from the existing literature by simultaneously considering the following three problem dimensions. Namely, we consider (1) *distributed*, (2) *Nesterov-like (accelerated) gradient methods* that operate on (3) *random networks*. To our best knowledge, neither of the exiting works considers these three dimensions simultaneously.

D. Paper Organization

The next paragraph sets notation. Section II introduces the network and optimization models, reviews D–NG and D–NC in [3], and gives certain preliminary results. Section III presents mD–NG, states its convergence rate, and proves the rate. Section IV presents mD–NC and its convergence rate with proofs. Section V discusses extensions to our results. Section VI illustrates mD–NG and mD–NC on a Huber loss example. We conclude in Section VII. The remaining proofs are in the Appendix.

Notation

Denote by: $\mathbb{R}^d$ the d-dimensional real space: A_{lm} or $[A]_{lm}$ the (l,m) entry of A; $A^\top$ the transpose of A; $[a]_{l:m}$ the selection of the l-th, $(l+1)$-th, $\cdots$, m-th entries of vector a; I, 0, $\mathbf{1}$, and e_i, respectively, the identity matrix, the zero matrix, the column vector with unit entries, and the i-th column of I; $J := (1/N)\mathbf{1}\mathbf{1}^\top$ denotes the $N \times N$ ideal consensus matrix; $\otimes$ the Kronecker product of matrices; $\|\cdot\|_l$ the vector (matrix) l-norm of its argument; $\|\cdot\| = \|\cdot\|_2$ the Euclidean (spectral) norm of its vector (matrix) argument ($\|\cdot\|$ also denotes the modulus of a scalar); $\lambda_i(\cdot)$ the i-th smallest *in modulus* eigenvalue; $A \succ 0$ a positive definite Hermitian matrix A; $\lceil a \rceil$ the smallest integer greater than or equal to a real scalar a; $\nabla\phi(x)$ the gradient at x of a differentiable function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$, $d \geq 1$; $\mathbb{P}(\cdot)$ and $\mathbb{E}[\cdot]$ the probability and expectation, respectively. For two positive sequences η_n and χ_n, we have: $\eta_n = O(\chi_n)$ if $\limsup_{n\rightarrow\infty} \frac{\eta_n}{\chi_n} < \infty$; $\eta_n = \Omega(\chi_n)$ if $\liminf_{n\rightarrow\infty} \frac{\eta_n}{\chi_n} > 0$; and $\eta_n = \Theta(\chi_n)$ if $\eta_n = O(\chi_n)$, and $\eta_n = \Omega(\chi_n)$.

II. Model and Preliminaries

Section II-A introduces the network and optimization models, Section II-B reviews the D–NG and D–NC distributed methods proposed in [3], and Section II-C gives preliminary results of certain products of time varying matrices.

A. Problem Model

1) Random Network Model: The network is random, due to link failures or communication protocol used (e.g., gossip, [36], [37].) It is defined by a sequence $\{W(k)\}_{k=1}^\infty$ of $N \times N$ random weight matrices.

Assumption 1 (Random Network): We have:

(a) The sequence $\{W(k)\}_{k=1}^\infty$ is i.i.d.

(b) Almost surely (a.s.), $W(k)$ are symmetric, stochastic, with strictly positive diagonal entries.

(c) There exists $\underline{w} > 0$ such that, for all $i, j = 1, \ldots, N$, a.s. $W_{ij}(k) \notin (0, \underline{w})$.

By Assumptions 1 (b) and (c), $W_{ii}(k) \geq \underline{w}$ a.s., $\forall i$; also, $W_{ij}(k)$, $i \neq j$, may be zero, but if $W_{ij}(k) > 0$ (nodes i and j communicate) it is non-negligible (at least $\underline{w}$). Assumption 1 (c) is mild and standard in the analysis of consensus and distributed gradient methods, e.g., [1], [23]. It says that a node i always gives a non-negligible weight to itself.

Let $\overline{W} := \mathbb{E}\left[W(k)\right]$, the supergraph $\mathcal{G} = (\mathcal{N}, E)$, $\mathcal{N}$ the set of N nodes, and $E = \left\{\{i,j\} : i < j, \overline{W}_{ij} > 0\right\}$–$\mathcal{G}$ collects all realizable links, all pairs $\{i,j\}$ for which $W_{ij}(k) > 0$ with positive probability.

2) Link Failure Model is covered by Assumption 1. This model is suitable, e.g., for a practical WSN, where random packet dropouts are adequately modeled by random link failures. Here, each link $\{i,j\} \in E$ at time k is Bernoulli: when it is one, $\{i,j\}$ is online (communication), and when it is zero, the link fails (no communication). The Bernoulli links are independent over time, but may be correlated in space. Possible weights are: 1) $i \neq j$, $\{i,j\} \in E$: $W_{ij}(k) = w_{ij} = 1/N$, when $\{i,j\}$ is online, and $W_{ij}(k) = 0$, else; 2) $i \neq j$, $\{i,j\} \notin E$: $W_{ij}(k) \equiv 0$; and 3) $W_{ii}(k) = 1 - \sum_{j \neq i} W_{ij}(k)$. While the weights $W_{ij}(k)$, $\{i,j\} \in E$ here are binary random variables (taking values either $1/N$ or 0), the *diagonal weights* $W_{ii}(k)$ have a more complex probability distribution.

As noted, our network model covers intermittent link failures but does not cover node failures, which may occur due to, e.g., energy depletion of a node. Modeling node failures is an interesting topic for future work.

We further make the following Assumption.

Assumption 2 (Network Connectedness): $\mathcal{G}$ is connected.

Denote by $\widetilde{W}(k) = W(k) - J = W(k) - (1/N)\mathbf{1}\mathbf{1}^\top$, by

$$\widetilde{\Phi}(k,t) = \widetilde{W}(k) \cdots \widetilde{W}(t+2), \; t = 0, 1, \ldots, k-2, \tag{5}$$

and by $\widetilde{\Phi}(k, k-1) = I$. One can show that $\overline{\mu} := \left(\left\|\mathbb{E}\left[W^2(k)\right] - J\right\|\right)^{1/2}$ is the square root of the second largest eigenvalue of $\mathbb{E}\left[W^2(k)\right]$ and that, under Assumptions 1 and 2, $\overline{\mu} < 1$. Lemma 1 (proof in Appendix A) shows that $\overline{\mu}$ characterizes the geometric decay of the first and second moments of $\widetilde{\Phi}(k,t)$.

Lemma 1: Let Assumptions 1 and 2 hold. Then:

$$\mathbb{E}\left[\left\|\widetilde{\Phi}(k,t)\right\|\right] \leq N^{1/2}\,\overline{\mu}^{k-t-1} \tag{6}$$

$$\mathbb{E}\left[\left\|\widetilde{\Phi}(k,t)^\top \widetilde{\Phi}(k,t)\right\|\right] \leq N\,\overline{\mu}^{2(k-t-1)} \tag{7}$$

$$\mathbb{E}\left[\left\|\widetilde{\Phi}(k,s)^\top \widetilde{\Phi}(k,t)\right\|\right] \leq N^{3/2}\,\overline{\mu}^{(k-t-1)+(k-s-1)}, \tag{8}$$

for all $t, s = 0, \ldots, k-1$, for all $k = 1, 2, \ldots$

The bounds in (6)–(8) may be loose, but are enough to prove the results below and simplify the presentation.

3) Optimization Model: We now introduce the optimization model. The nodes solve the unconstrained problem (1). The function $f_i : \mathbb{R}^d \to \mathbb{R}$ is known only to node i. We impose the following three Assumptions.

Assumption 3 (Solvability): There exists a solution $x^\star \in \mathbb{R}^d$ such that $f(x^\star) = f^\star := \inf_{x \in \mathbb{R}^d} f(x)$.

Assumption 4 (Lipschitz Continuous Gradient): For all i, f_i is convex, differentiable, and has Lipschitz continuous gradient with constant $L \in [0, \infty)$:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d.$$

Assumption 5 (Bounded Gradients): There exists a constant $G \in [0, \infty)$ such that, $\forall i$, $\|\nabla f_i(x)\| \leq G$, $\forall x \in \mathbb{R}^d$.

Assumptions 3 and 4 are standard in gradient methods; in particular, Assumption 4 is precisely the Assumption required by the centralized Nesterov gradient method [38]. Assumption 5 is not required in centralized Nesterov. Reference [3] demonstrates that (even on) static networks and a constant $W(k) \equiv W$, the convergence rates of D–NG or of the standard distributed gradient method in [1] become arbitrarily slow if Assumption 5 is relaxed. See examples 1–3 in the introduction for the f_i's that obey Assumptions 3–5.

Algorithms D–NG and D–NC for Static Networks

We briefly review the D–NG and D–NC methods proposed in [3] for static networks. For this purpose, current subsection assumes a static, deterministic, connected network, with an associated symmetric, stochastic, deterministic weight matrix $W(k) \equiv W$, with $\overline{\mu} = \|W - J\| < 1$. With D–NG, the matrix W is positive definite, while with D–NC this requirement is not needed.

Algorithm D–NG: Node i maintains its solution estimate $x_i(k)$ and an auxiliary variable $y_i(k)$, $k = 0, 1, \ldots$ It uses arbitrary initialization $x_i(0) = y_i(0) \in \mathbb{R}^d$ and, for $k = 1, 2, \ldots$, performs the updates

$$x_i(k) = \sum_{j \in O_i} W_{ij}\, y_j(k-1) - \alpha_{k-1} \nabla f_i(y_i(k-1)) \tag{9}$$

$$y_i(k) = (1 + \beta_{k-1})\, x_i(k) - \beta_{k-1}\, x_i(k-1). \tag{10}$$

In (9)–(10), $O_i = \{j \in \{1, \ldots, N\} : W_{ij} > 0\}$ is the neighborhood of node i (including node i). For $k = 0, 1, 2, \ldots$, the step-size α_k is:

$$\alpha_k = c/(k+1), \; c \leq 1/(2L), \tag{11}$$

and β_k is the sequence from the centralized Nesterov gradient method, [38]:

$$\beta_k = \frac{k}{k+3}. \tag{12}$$

The D–NG algorithm works as follows. At iteration k, node i receives the variables $y_j(k-1)$ from its neighbors $j \in O_i - \{i\}$, and updates $x_i(k)$ and $y_i(k)$ via (9) and (10).

Algorithm D–NC operates in two time scales. In the outer (slow time scale) iterations k, each node i updates its solution estimate $x_i(k)$, and updates an auxiliary variable $y_i(k)$ (as with the D–NG). In the inner iterations s, nodes perform two rounds of consensus with the number of inner iterations $\tau_x(k)$ and $\tau_y(k)$, given by:

$$\tau_x(k) = \left\lceil \frac{2\log k}{-\log \overline{\mu}} \right\rceil, \; \tau_y(k) = \left\lceil \frac{\log 3}{-\log \overline{\mu}} + \frac{2\log k}{-\log \overline{\mu}} \right\rceil.$$

The D–NC algorithm is presented in Algorithm 1.

Algorithm 1: Algorithm D–NC

```
1: Initialization: Node $i$ sets: $x_i(0) = y_i(0) \in \mathbb{R}^d$, $k = 1$.
2: Node $i$ calculates: $x_i^{(a)}(k) = y_i(k-1) - \alpha \nabla f_i(y_i(k-1))$.
3: (First consensus) Nodes run average consensus initialized
   by $x_i^{(c)}(s = 0, k) = x_i^{(a)}(k)$:
       $x_i^{(c)}(s,k) = \sum_{j \in O_i} W_{ij} x_j^{(c)}(s-1,k), \; s = 1, \ldots, \tau_x(k)$
   and set $x_i(k) := x_i^{(c)}(s = \tau_x(k), k)$.
4: Node $i$ calculates $y_i^{(a)}(k) = x_i(k) + \beta_{k-1}(x_i(k) - x_i(k-1))$.
5: (Second consensus) Nodes run average consensus
   initialized by $y_i^{(c)}(s = 0, k) = y_i^{(a)}(k)$:
       $y_i^{(c)}(s,k) = \sum_{j \in O_i} W_{ij} y_j^{(c)}(s-1,k), \; s = 1, \ldots, \tau_y(k)$
   and set $y_i(k) := y_i^{(c)}(s = \tau_y(k), k)$.
6: Set $k \mapsto k+1$ and go to step 2.
```

B. Scalar Sums and Products of Time-Varying Matrices

In the convergence rate analysis of mD–NG and mD–NC, we make use of certain scalar sum bounds and bounds on the products of 2×2 time varying matrices. We state these preliminary results here and prove them in Appendix B.

Lemma 2 (Scalar Sums): Let $0 < r < 1$. Then, for all $k = 1, 2, \ldots$

$$\sum_{t=1}^{k} r^t \, t \leq \frac{r}{(1-r)^2} \leq \frac{1}{(1-r)^2} \tag{13}$$

$$\sum_{t=0}^{k-1} r^{k-t-1} \frac{1}{t+1} \leq \frac{1}{(1-r)^2 \, k}. \tag{14}$$

For $k = 1, 2, \ldots$, let $B(k)$ be:

$$B(k) := \begin{bmatrix} (1+\beta_{k-1}) & -\beta_{k-1} \\ 1 & 0 \end{bmatrix}, \tag{15}$$

with β_{k-1} in (12). Further, let $\mathcal{B}(k,-1) := I$ and:

$$\mathcal{B}(k,t) := B(k) \cdots B(k-t), \; t = 0, 1, \ldots, k-2. \tag{16}$$

We have the following important result, proved in Appendix B.

Lemma 3: Consider $\mathcal{B}(k,t)$ in (16). Then, for all $t = 0, \ldots, k-1$, for all $k = 1, 2, \ldots$

$$\|\mathcal{B}(k, k-t-2)\| \leq 8 \frac{(k-t-1)(t+1)}{k} + 5. \tag{17}$$

Algorithm mD–NG

We now present our mD–NG algorithm for random networks. Section III-A describes the algorithm, Section III-B sates our results on its convergence rate, and Section III-C proves these results.

C. The Algorithm

We modify D–NG in (9)–(10) to handle random networks. Node i maintains its solution estimate $x_i(k)$ and auxiliary variable $y_i(k)$, $k = 0, 1, \ldots$ It uses arbitrary initialization $x_i(0) = y_i(0) \in \mathbb{R}^d$ and, for $k = 1, 2, \ldots$, performs the updates

$$x_i(k) = \sum_{j \in O_i(k)} W_{ij}(k) \, y_j(k-1) - \alpha_{k-1} \nabla f_i(y_i(k-1)) \tag{18}$$

$$y_i(k) = (1 + \beta_{k-1}) \, x_i(k) - \beta_{k-1} \sum_{j \in O_i(k)} W_{ij}(k) \, x_j(k-1). \tag{19}$$

In (18)–(19), $O_i(k) = \{j \in \{1, \ldots, N\} : W_{ij}(k) > 0\}$ is the (random) neighborhood of node i (including node i) at time k. For $k = 0, 1, 2, \ldots$, the step-size α_k is in (11), and the sequence β_k is in (12). We assume nodes know L (Section V relaxes this.)

The mD–NG algorithm (18)–(19) differs from D–NG in (9)–(10) in step (19). With D–NG, nodes communicate only the $y_j(k-1)$'s; with mD–NG, they also communicate the $x_j(k-1)$'s (see the sum term in (19)). This modification allows for the robustness to link failures. (See Theorems 4 and 5 and the simulations in Section VI.) Further, mD–NG does not require the weight matrix to be positive definite, while D–NG does.

1) Vector Form: Let $x(k) := (x_1(k)^\top, \ldots, x_N(k)^\top)^\top$, $y(k) := (y_1(k)^\top, \ldots, y_N(k)^\top)^\top$, and $F : \mathbb{R}^{Nd} \to \mathbb{R}$, $F(x_1, \ldots, x_N) := f_1(x_1) + \cdots + f_N(x_N)$. Then, for $k = 1, 2, \ldots$, with $x(0) = y(0) \in \mathbb{R}^{Nd}$, $W(k) \otimes I$ the Kronecker product of $W(k)$ with the $d \times d$ identity I, mD–NG in vector form is:

$$x(k) = (W(k) \otimes I) \; y(k-1) - \alpha_{k-1} \, \nabla F(y(k-1)) \tag{20}$$

$$y(k) = (1 + \beta_{k-1}) \, x(k) - \beta_{k-1} \; (W(k) \otimes I) \; x(k-1). \tag{21}$$

2) Initialization: For notation simplicity, without loss of generality (wlog), we assume, with all proposed methods, that nodes initialize their estimates to the same values, i.e., $x_i(0) = y_i(0) = x_j(0) = y_j(0)$, for all i, j; for example, $x_i(0) = y_i(0) = x_j(0) = y_j(0) = 0$.

3) Communication and Computational Costs, Per Node and Iteration k: We show the communication and computational costs of mD–NG per iteration k and compare it with the costs of the standard distributed gradient method in [1]. Consider, for simplicity, a regular, static network, with degree (number of neighbors of each node) δ. Further, suppose that the computational cost of computing the gradient of each f_i is similar, $i = 1, \ldots, N$. mD–NG requires, per k, $(2\delta + 5)d$ multiplications, $(2\delta + 3)d$ additions, and one gradient evaluation ($\nabla f_i(y_i(k-1))$), while [1] requires $(\delta + 2)d$ multiplications, $(\delta + 1)d$ additions, and one gradient evaluation. The cost of the gradient evaluation depends on the f_i's and is usually dominant over other terms. Regarding communications per k, mD–NG requires $2d$ communicated scalars per node, while [1] requires d communicated scalars per node. Hence, compared with [1], mD–NG has about twice larger communication cost and (less than) twice larger computational cost per k. As we show further ahead in Theorem 5, mD–NG has rate $O(\log k / k)$, while [1] is (in a worst-case sense) no faster than $\Omega(k^{-2/3})$, [3]. Given these rates and the fact that one iteration of mD–NG costs (roughly) as two iterations of [1], it is clear that mD–NG has a faster rate than [1] in terms of the overall cost.

D. Convergence Rate of mD–NG

We state our convergence rate result for mD–NG. Proofs are in Section III-C. We estimate the expected optimality gap in the cost at each node i normalized by N, e.g., [2], [27]: $\frac{1}{N}\mathbb{E}\left[f(x_i) - f^\star\right]$, where x_i is node i's solution at a certain stage of the algorithm. We study how node i's optimality gap decreases with: 1) the number k of iterations, or of per-node gradient evaluations; and 2) the total number $\mathcal{K}$ of $2\,d$-dimensional vector communications per node. With mD–NG, $k = \mathcal{K}$–at each k, there is one and only one per-node $2\,d$-dimensional communication and one per-node gradient evaluation. Not so with mD–NC, as we will see. We establish for both methods convergence rates on the mean square disagreements of different node estimates in terms of k and $\mathcal{K}$, showing that it converges to zero.

Let: the global averages of the nodes' estimates be $\overline{x}(k) := \frac{1}{N}\sum_{i=1}^N x_i(k)$ and $\overline{y}(k) := \frac{1}{N}\sum_{i=1}^N y_i(k)$; the disagreements: $\widetilde{x}_i(k) = x_i(k) - \overline{x}(k)$ and $\widetilde{x}(k) = \left(\widetilde{x}_1(k)^\top, \ldots, \widetilde{x}_N(k)^\top\right)^\top$, and analogously for $\widetilde{y}_i(k)$ and $\widetilde{y}(k)$; and $\widetilde{z}(k) := \left(\widetilde{y}(k)^\top, \widetilde{x}(k)^\top\right)^\top$. We have the following Theorem on $\mathbb{E}\left[\|\widetilde{z}(k)\|\right]$ and $\mathbb{E}\left[\|\widetilde{z}(k)\|^2\right]$. Note $\|\widetilde{x}(k)\| \leq \|\widetilde{z}(k)\|$, and so $\mathbb{E}\left[\|\widetilde{x}(k)\|\right] \leq \mathbb{E}\left[\|\widetilde{z}(k)\|\right]$ and $\mathbb{E}\left[\|\widetilde{x}(k)\|^2\right] \leq \mathbb{E}\left[\|\widetilde{z}(k)\|^2\right]$. (Equivalent inequalities hold for $\widetilde{y}(k)$.) Recall also $\overline{\mu}$ in Lemma 1.

Theorem 4: Consider mD–NG (18)–(19) under Assumptions 1–5. Then, for all $k = 1, 2, \ldots$

$$\mathbb{E}\left[\|\widetilde{z}(k)\|\right] \leq \frac{50\, c\, N\, G}{(1-\overline{\mu})^2}\,\frac{1}{k} \tag{22}$$

$$\mathbb{E}\left[\|\widetilde{z}(k)\|^2\right] \leq \frac{50^2\, c^2\, N^{5/2} G^2}{(1-\overline{\mu})^4\, k^2}. \tag{23}$$

Theorem 5 establishes the convergence rate of mD–NG as $O(\log k/k)$ (and $O(\log \mathcal{K}/\mathcal{K})$).

Theorem 5: Consider mD–NG (18)–(19) under Assumptions 1–5. Let $\|\overline{x}(0) - x^\star\| \leq R$, $R \geq 0$. Then, at any node i, the expected normalized optimality gap $\frac{1}{N}\mathbb{E}\left[f(x_i(k)) - f^\star\right]$ is $O(\log k/k)$; more precisely:

$$\frac{\mathbb{E}\left[f(x_i(k)) - f^\star\right]}{N} \leq \frac{2\,R^2}{c}\,\frac{1}{k} + \frac{50^2\, c^2\, N^{3/2}\, L\, G^2}{(1-\overline{\mu})^4}\,\frac{1}{k}\sum_{t=1}^{k-1}\frac{(t+2)^2}{(t+1)t^2} + \frac{50\, N\, c\, G^2}{(1-\overline{\mu})^2}\,\frac{1}{k}. \tag{24}$$

We now examine how the expected optimality gap with mD–NG depends on the number of nodes N and the network connectivity, measured by $\overline{\mu}$. (The smaller $\overline{\mu}$ is, the better the connectivity.) We also compare mD–NG with D–NG. We consider separately random and static networks.

1) Network Dependence – Random Networks: First, note that the right hand side in (24) is upper bounded by $\mathcal{C}\left(\frac{1}{k}\sum_{t=1}^k \frac{(t+2)^2}{(t+1)t^2}\right)$, where the constant $\mathcal{C} = \frac{2\,R^2}{c} + \frac{50^2\, c^2\, N^{3/2}\, L\, G^2}{(1-\overline{\mu})^4} + \frac{50\, N\, c\, G^2}{(1-\overline{\mu})^2}$ captures the effect of N and $\overline{\mu}$. Ignoring the effect of constants $L, R,$ and G: $\mathcal{C} = O\left(\frac{1}{c} + \frac{c^2 N^{3/2}}{(1-\overline{\mu})^4} + \frac{cN}{(1-\overline{\mu})^2}\right)$. Assuming that nodes know $\overline{\mu}, L, N$, we can further set the optimized step-size $c = \frac{(1-\overline{\mu})^{4/3}}{2N^{1/2}L}$, and obtain: $\mathcal{C} = O\left(\frac{N^{1/2}}{(1-\overline{\mu})^{4/3}}\right)$. We now estimate how the quantity $\overline{\mu}$ depends on N for several standard models (chain, ring, geometric network, and expanders) under link failures. Recall the definition of the supergraph $\mathcal{G} = (\mathcal{N}, E)$. We adopt a spatio-temporally independent link failure model, where each link fails with equal probability $1-p$. We assume that p does not depend on N. Whenever a link is online, we set its weight to a constant w_0, specified further ahead. Let $\mathcal{L}$ be the $N \times N$ (deterministic) symmetric graph Laplacian matrix associated with $\mathcal{G}$, defined by: 1) $\mathcal{L}_{ij} = -1$, $i \neq j$, $\{i,j\} \in E$; $\mathcal{L}_{ij} = 0$, $i \neq j$, $\{i,j\} \notin E$, and $\mathcal{L}_{ii} = -\sum_{j \neq i}\mathcal{L}_{ij}$. From ([39], (18) and (21)), it can be shown that: $\mathbb{E}\left[W^2(k)\right] = I - 2pw_0(1-w_0(1-p))\mathcal{L} + p^2 w_0^2 \mathcal{L}^2$. Denote by η_i the i-th smallest eigenvalue of $\mathcal{L}$. Setting $w_0 = 1/\eta_N$, it can be shown that: $\overline{\mu} = \left(\|\mathbb{E}[W^2(k)]\| - J\right)^{1/2} = 1 - \Omega(\eta_2/\eta_N)$. Ignoring logarithmic factors, and applying standard results (see, e.g., [2]) on the quantity $\eta_2/\eta_N = 1 - \Omega(1/N^2)$ (chain/ring); $1 - \Omega(1/N)$ (geometric); and $1 - \Omega(1)$ (expander), we obtain the following estimates on the constant $\mathcal{C}$ with the mD–NG algorithm: $\mathcal{C} = O\left(N^{19/6}\right)$ (chain/ring); $O(N^{11/6})$ (geometric); and $O(N^{1/2})$ (expander). On random networks, D–NG may diverge (see Section VI.)

2) Network Dependence – Static Networks: We consider the static network case when the link failure probability $1-p = 0$. It can be shown that, in this case, the bound on $\mathcal{C}$ improves to: $\mathcal{C} = O\left(\frac{1}{c} + \frac{c^2}{(1-\overline{\mu})^4} + \frac{cN^{1/2}}{(1-\overline{\mu})^2}\right)$. Setting the optimized step size $c = \frac{(1-\overline{\mu})^{4/3}}{2N^{1/4}L}$, we obtain: $\mathcal{C} = O\left(\frac{N^{1/4}}{(1-\overline{\mu})^{4/3}}\right)$, which gives $\mathcal{C} = O(N^{35/12})$ (chain/ring); $O(N^{19/12})$ (geometric); and $O(N^{1/4})$ (expander). The static network scaling of mD–NG is worse than D–NG, which achieves in the same setting at least $\mathcal{C} = O\left(\frac{N^{1/4}}{(1-\overline{\mu})^{1+\xi}}\right)$, with $\xi > 0$ arbitrarily small. Therefore, compared with D–NG, mD–NG slightly compromises the convergence constant but enjoys resilience to link failures.

Proofs of Theorems 4 and 5

In this subsection, we prove Theorems 4 and 5.

Proof of Theorem 4: Through this proof and the rest of the paper, we establish certain equalities and inequalities on random quantities of interest. These equalities and inequalities further ahead hold either: 1) almost surely, or: 2) in expectation. From the notation, it is clear which of the two cases is in force. For notational simplicity, we perform the proof of Theorem 4 for the case $d = 1$, but the proof extends for generic $d > 1$. The proof has three steps. In Step 1, we derive the dynamic equation for the disagreement $\widetilde{z}(k) = \left(\widetilde{y}(k)^\top, \widetilde{x}(k)^\top\right)^\top$. In Step 2, we unwind the dynamic equation, expressing $\widetilde{z}(k)$ in terms of the products $\widetilde{\Phi}(k,t)$ in (5) and $\mathcal{B}(k,t)$ in (16). Finally, in Step 3, we apply the already established bounds on the norms of the latter products.

Step 1. Disagreement: Note that $JW(k) = \frac{1}{N}\mathbf{1}\mathbf{1}^\top W(k) = J$, because $W(k)$ is symmetric stochastic. Also, $\widetilde{W}(k)J = (W(k) - J)J = J - J = 0$. From the last two equalities, $(I - J)W(k) = \widetilde{W}(k) = \widetilde{W}(k) - \widetilde{W}(k)J =$

$\widetilde{W}(k)(I-J)$. Using the latter, multiplying (20)–(21) from the left by $(I-J)$, obtain (recall $B(k)$ in (15)):

$$\widetilde{z}(k) = \left(B(k) \otimes \widetilde{W}(k)\right) \widetilde{z}(k-1) + u(k-1), \tag{25}$$

for $k = 1, 2, \ldots$ and $\widetilde{z}(0) = 0$, where

$$u(k-1) = -\begin{bmatrix} \alpha_{k-1}(1+\beta_{k-1})(I-J)\nabla F(y(k-1)) \\ \alpha_{k-1}(I-J)\nabla F(y(k-1)) \end{bmatrix}. \tag{26}$$

Step 2. Unwinding Recursion (25): Recall $\widetilde{\Phi}(k,t)$ in (5), and $\mathcal{B}(k,t)$ in (16). Then, unwinding (25), and using the Kronecker product property $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, we obtain for all $k = 1, 2, \ldots$

$$\widetilde{z}(k) = \sum_{t=0}^{k-1} \left(\mathcal{B}(k, k-t-2) \otimes \widetilde{\Phi}(k,t)\right) u(t). \tag{27}$$

Step 3. Finalizing the Proof: Consider $u(t)$ in (26). By Assumption 5, we have $\|\nabla F(y(t))\| \leq \sqrt{N}G$. Using this, the step-size $\alpha_t = c/(t+1)$, and $\|I-J\| = 1$, get $\|u(t)\| \leq \frac{\sqrt{5}\,c\,\sqrt{N}\,G}{t+1}$, for any random realization of $u(t)$. With this, $\|A \otimes B\| \leq \|A\|\|B\|$, Lemma 3, and the sub-multiplicative and sub-additive properties of norms, obtain from (27):

$$\|\widetilde{z}(k)\| \leq \left(8\sqrt{5}\,c\,\sqrt{N}\,G\right) \frac{1}{k} \sum_{t=0}^{k-1} \|\widetilde{\Phi}(k,t)\| \,(k-t-1) + \left(5\sqrt{5}\,c\,\sqrt{N}\,G\right) \sum_{t=0}^{k-1} \|\widetilde{\Phi}(k,t)\| \,\frac{1}{t+1}. \tag{28}$$

Taking expectation, and using Lemma 1:

$$\mathbb{E}\left[\|\widetilde{z}(k)\|\right] \leq \left(8\sqrt{5}\,c\,NG\right) \frac{1}{k} \sum_{t=0}^{k-1} \overline{\mu}^{k-t-1}(k-t-1) + \left(5\sqrt{5}\,c\,N\,G\right) \sum_{t=0}^{k-1} \overline{\mu}^{k-t-1} \,\frac{1}{t+1}.$$

Finally, applying Lemma 2 to the last equation with $r = \overline{\mu}$, the result in (22) follows.

Now prove (23). Consider $\|\widetilde{z}(k)\|^2$. From (27):

$$\begin{aligned}\|\widetilde{z}(k)\|^2 &= \sum_{t=0}^{k-1}\sum_{s=0}^{k-1} u(t)^\top \left(\mathcal{B}(k,k-t-2)^\top \otimes \widetilde{\Phi}(k,t)^\top\right) \left(\mathcal{B}(k,k-s-2) \otimes \widetilde{\Phi}(k,s)\right) u(s) \\ &= \sum_{t=0}^{k-1}\sum_{s=0}^{k-1} u(t)^\top \left(\mathcal{B}(k,k-t-2)^\top \mathcal{B}(k,k-s-2)\right) \otimes \left(\widetilde{\Phi}(k,t)^\top \widetilde{\Phi}(k,s)\right) u(s),\end{aligned}$$

where the last equality again uses the property $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$. Further, obtain:

$$\begin{aligned}&\|\widetilde{z}^{(}k)\|^2 \\ &\leq \sum_{t=0}^{k-1}\sum_{s=0}^{k-1} \|\mathcal{B}(k,k-t-2)\| \, \|\mathcal{B}(k,k-s-2)\| \left\|\widetilde{\Phi}(k,t)^\top \widetilde{\Phi}(k,s)\right\| \|u(t)\| \, \|u(s)\| \\ &\leq \sum_{t=0}^{k-1}\sum_{s=0}^{k-1} \left(\frac{8(k-t-1)(t+1)}{k}+5\right)\left(\frac{8(k-s-1)(s+1)}{k}+5\right) \left\|\widetilde{\Phi}(k,t)^\top \widetilde{\Phi}(k,s)\right\| \frac{5\,c^2\,NG^2}{(t+1)(s+1)}.\end{aligned} \tag{29}$$

The last inequality uses Lemma 3 and $\|u(t)\| \leq \left(\sqrt{5}c\sqrt{N}G\right)/(t+1)$. Taking expectation and applying Lemma 1, obtain:

$$\begin{aligned}&\mathbb{E}\left[\|\widetilde{z}(k)\|^2\right] \\ &\leq \left(5c^2N^{5/2}G^2\right) \sum_{t=0}^{k-1}\sum_{s=0}^{k-1} \left(\frac{8(k-t-1)(t+1)}{k}+5\right) \left(\frac{8(k-s-1)(s+1)}{k}+5\right) \frac{\overline{\mu}^{k-t-1+k-s-1}}{(t+1)(s+1)} \\ &= \left(5c^2N^{5/2}G^2\right) \left(\sum_{t=0}^{k-1} \left(\frac{8(k-t-1)(t+1)}{k}+5\right) \frac{\overline{\mu}^{k-t-1}}{t+1}\right)^2 \\ &\leq \frac{50^2\,c^2\,N^{5/2}G^2}{(1-\overline{\mu})^4\,k^2}.\end{aligned}$$

The last inequality applies Lemma 2. Thus, the bound in (23). The proof of Theorem 4 is complete.

Proof of Theorem 5: The proof parallels that of Theorem 5 (a) in [3]. We outline it and refer to ([3], Lemma 2, Lemma 3, Theorem 5 (a), and their proofs.) It is based on the evolution of the global averages $\overline{x}(k) = \frac{1}{N}\sum_{i=1}^N x_i(k)$, and $\overline{y}(k) = \frac{1}{N}\sum_{i=1}^N y_i(k)$. Let:

$$\begin{aligned}\widehat{f}_{k-1} &:= \sum_{i=1}^N \Big(f_i(y_i(k-1)) + \nabla f_i(y_i(k-1))^\top (\overline{y}(k-1) - y_i(k-1))\Big) \\ \widehat{g}_{k-1} &:= \sum_{i=1}^N \nabla f_i\left(y_i(k-1)\right) \\ L_{k-1} &:= \frac{N}{\alpha_{k-1}} \geq 2NLk, \;\; \delta_{k-1} := L\|\widetilde{y}(k-1)\|^2.\end{aligned} \tag{30}$$

Then, it is easy to show that $\overline{x}(k)$, $\overline{y}(k)$ evolve as:

$$\overline{x}(k) = \overline{y}(k-1) - \frac{\widehat{g}_{k-1}}{L_{k-1}} \tag{31}$$

$$\overline{y}(k) = (1+\beta_{k-1})\,\overline{x}(k) - \beta_{k-1}\overline{x}(k-1), \tag{32}$$

$k = 1, 2, \ldots$, with $\overline{x}(0) = \overline{y}(0)$. As shown in [3], $\left(\widehat{f}_{k-1}, \widehat{g}_{k-1}\right)$ is a (L_{k-1}, δ_{k-1}) inexact oracle, i.e., it holds that for all points $x \in \mathbb{R}^d$:

$$f(x) + \widehat{g}_{k-1}^\top(x - \overline{y}(k-1)) \leq f(x) \leq \widehat{f}_{k-1} + \widehat{g}_{k-1}^\top(x - \overline{y}(k-1)) + \frac{L_{k-1}}{2}\|x - \overline{y}(k-1)\|^2 + \delta_{k-1}. \tag{33}$$

From (30), $\widehat{f}_{k-1}$, $\widehat{g}_{k-1}$, and $\widehat{\delta}_{k-1}$ are functions of $y(k-1)$. Inequalities (33) hold for any random realization of $y(k-1)$

and any $x \in \mathbb{R}^d$. We apply now Lemma 2 in [3], with δ_{k-1} as in (30). Get:

$$
\begin{aligned}
&(k+1)^2\left(f(\overline{x}(k)) - f^\star\right) + \frac{2Nk}{c}\|\overline{v}(k) - x^\star\|^2 \\
&\le (k^2-1)\left(f(\overline{x}(k-1)) - f^\star\right) + \frac{2Nk}{c}\|\overline{v}(k-1) - x^\star\|^2 \\
&\quad + (k+1)^2 L \|\widetilde{y}(k-1)\|^2, \qquad (34)
\end{aligned}
$$

where $\overline{v}(k) = \left(\overline{y}(k) - (1-\theta_k)\overline{x}(k)\right)/\theta_k,\ \theta_k = 2/(k+2)$. Dividing (34) by k and unwinding the resulting inequality, get:

$$
\begin{aligned}
\frac{1}{N}\left(f(\overline{x}(k)) - f^\star\right) \le \frac{2}{k\,c}\|\overline{x}(0) - x^\star\|^2 \\
+ \frac{L}{N\,k}\sum_{t=1}^{k}\frac{(t+1)^2}{t}\|\widetilde{y}(t-1)\|^2. \qquad (35)
\end{aligned}
$$

Next, using Assumption 5, obtain, $\forall i$:

$$
\frac{1}{N}\left(f(x_i(k)) - f^\star\right) \le \frac{1}{N}\left(f(\overline{x}(k)) - f^\star\right) + G\|\widetilde{x}(k)\|. \quad (36)
$$

The proof is completed after combining (35) and (36), taking expectation, and using in Theorem 4 the bounds $\mathbb{E}\left[\|\widetilde{x}(k)\|\right] \le \mathbb{E}\left[\|\widetilde{z}(k)\|\right]$ and $\mathbb{E}\left[\|\widetilde{y}(k)\|^2\right] \le \mathbb{E}\left[\|\widetilde{z}(k)\|^2\right]$.

Algorithm mD–NC

We present mD–NC. Section IV-A defines additional random matrices needed for representation of mD–NC and presents mD–NC. Section IV-B states our results on its convergence rate, while Section IV-C proves the results.

E. Model and Algorithm

We consider a sequence of i.i.d. random matrices that obey Assumptions 1 and 2. We index these matrices with two-indices since mD–NC operates in two time scales–an inner loop, indexed by s with τ_k iterations, and an outer loop indexed by k, where:

$$
\tau_k = \left\lceil \frac{3\,\log k + \log N}{-\log \overline{\mu}} \right\rceil. \qquad (37)
$$

For static networks, the term $\log N$ can be dropped. At each inner iteration, nodes utilize one communication round–each node broadcasts a $2d \times 1$ vector to all its neighbors. We denote by $W(k,s)$ the random weight matrix that corresponds to the communication round at the s-th inner iteration and k-th outer iteration. The matrices $W(k,s)$ are ordered lexicographically as

$$
\begin{aligned}
&W(k=1, s=1), W(k=1, s=2), \\
&\qquad \ldots, W(k=1, s=\tau_1), W(k=2, s=1), \ldots
\end{aligned}
$$

This sequence obeys Assumptions 1 and 2.

It will be useful to define the products of the weight matrices $W(k,s)$ over each outer iteration k:

$$
\mathcal{W}(k) := \Pi_{s=0}^{\tau_k - 1} W(k, \tau_k - s). \qquad (38)
$$

The matrices $\{\mathcal{W}(k)\}_{k=1}^{\infty}$ are independent but not identically distributed. Define $\widetilde{\mathcal{W}}(k) := \mathcal{W}(k) - J$, and, for $t = 0, 1, \ldots, k-1$:

$$
\widetilde{\Psi}(k,t) := \widetilde{\mathcal{W}}(k)\widetilde{\mathcal{W}}(k-1)\cdots\widetilde{\mathcal{W}}(t+1). \qquad (39)
$$

The Lemma below is proved in Appendix A.

Lemma 6: Let Assumptions 1 and 2 hold. Then, for all $k = 1, 2, \ldots$, for all $s, t \in \{0, 1, \ldots, k-1\}$:

$$
\mathbb{E}\left[\left\|\widetilde{\mathcal{W}}(k)\right\|^2\right] \le \frac{1}{k^6} \qquad (40)
$$

$$
\mathbb{E}\left[\|\widetilde{\Psi}(k,t)\|\right] \le \frac{1}{k^3(k-1)^3\cdots(t+1)^3} \qquad (41)
$$

$$
\mathbb{E}\left[\|\widetilde{\Psi}(k,t)^\top\widetilde{\Psi}(k,t)\|\right] \le \left(\frac{1}{k^3(k-1)^3\cdots(t+1)^3}\right)^2 \qquad (42)
$$

$$
\begin{aligned}
\mathbb{E}\left[\|\widetilde{\Psi}(k,s)^\top\widetilde{\Psi}(k,t)\|\right] \le \left(\frac{1}{k^3(k-1)^3\cdots(t+1)^3}\right) \\
\left(\frac{1}{k^3(k-1)^3\cdots(s+1)^3}\right). \qquad (43)
\end{aligned}
$$

1) The mD–NC Algorithm: mD–NC, in Algorithm 2, uses constant step-size $\alpha \le 1/(2L)$. Each node i maintains over (outer iterations) k the solution estimate $x_i(k)$ and an auxiliary variable $y_i(k)$. Recall $\overline{\mu}$ in Lemma 1.

Algorithm 2: mD–NC

```
1: Initialization: Node $i$ sets $x_i(0) = y_i(0) \in \mathbb{R}^d$; and $k = 1$.
2: Node $i$ calculates $x_i^{(a)}(k) = y_i(k-1) - \alpha\nabla f_i(y_i(k-1))$.
3: (Consensus) Nodes run average consensus on $\chi_i(s,k)$,
   initialized by $\chi_i(s=0,k) = \left(x_i^{(a)}(k)^\top, x_i(k-1)^\top\right)^\top$:
   $\chi_i(s,k) = \sum_{j \in O_i(k)} W_{ij}(k,s)\chi_j(s-1,k),\ s = 1, 2, \ldots, \tau_k,$
   with $\tau_k$ in (37), and set $x_i(k) := [\chi_i(s=\tau_k,k)]_{1:d}$ and
   $x_i^{(b)}(k-1) := [\chi_i(s=\tau_k,k)]_{d+1:2\,d}$. (Here $[a]_{l:m}$ is a
   selection of $l$-th, $l+1$-th, $\ldots$, $m$-th entries of vector $a$.)
4: Node $i$ calculates $y_i(k) = (1+\beta_{k-1})x_i(k) -$
   $\beta_{k-1}\,x_i^{(b)}(k-1)$.
5: Set $k \mapsto k+1$ and go to step 2.
```

Step 3 has τ_k communication rounds at outer iteration k. Nodes know L, $\overline{\mu}$, and N. Section V relaxes this.

2) mD–NC in Vector Form: Consider the matrices $\mathcal{W}(k)$ in (38). Use the compact notation as in mD–NG for $x(k)$, $y(k)$, and recall $F : \mathbb{R}^{Nd} \to \mathbb{R}$. Then for $k = 1, 2, \ldots$

$$
x(k) = \left(\mathcal{W}(k) \otimes I\right)\left[y(k-1) - \alpha\nabla F(y(k-1)\right] \qquad (44)
$$

$$
y(k) = (1+\beta_{k-1})x(k) - \beta_{k-1}\left(\mathcal{W}(k) \otimes I\right)x(k-1), \qquad (45)
$$

with $x(0) = y(0) \in \mathbb{R}^{Nd}$.

3) Communication and Computational Costs Per Node of Each Inner and Outer Iteration: Assume, for simplicity, a regular static network with degree δ. Each inner iteration s

requires: $2(\delta+1)d$ multiplications and $2\delta d$ additions (computational cost) and $2d$ scalar communications. Each outer iteration k requires $3d$ multiplications, $2d$ additions, and one gradient evaluation (computational cost), and requires no communications. Hence, compared with [1], one inner iteration of mD–NC costs (roughly) as two iterations of [1]. Comparing the rate $O(1/\mathcal{K}^{2-\xi})$ with mD–NC and $\Omega(1/\mathcal{K}^{2/3})$ with [1], it is clear that mD–NC has a faster rate than [1] in terms of the overall cost.

F. Convergence Rate of mD–NC

Define, like for mD–NG, the disagreements $\widetilde{x}_i(k)$, $\widetilde{y}_i(k)$, $\widetilde{x}(k)$, $\widetilde{y}(k)$, and $\widetilde{z}(k) := \left(\widetilde{y}(k)^\top, \widetilde{x}(k)^\top\right)^\top$.

Theorem 7: Consider mD–NC in Algorithm 2 under Assumptions 1–5. Then, for all $k = 1, 2, \ldots$

$$\mathbb{E}^2\left[\|\widetilde{z}(k)\|\right] \le \mathbb{E}\left[\|\widetilde{z}(k)\|^2\right] \le \frac{65^2\,\alpha^2\,NG^2}{k^4}. \tag{46}$$

Theorem 8: Consider mD–NC in Algorithm 2 under Assumptions 1–5. Let $\|\overline{x}(0) - x^\star\| \le R$, $R \ge 0$. Then, after $\mathcal{K}$ communication rounds (after k outer iterations)

$$\mathcal{K} = \sum_{t=1}^{k} \tau_t \le \frac{3}{-\log \overline{\mu}}\left[\,(k+1)\log(N(k+1))\,\right],$$

i.e., $\mathcal{K} = O\left(k \log k\right)$, we have, at any node i, $k = 1, 2, \ldots$

$$\frac{\mathbb{E}\left[f(x_i(k)) - f^\star\right]}{N} \le \frac{\frac{2}{\alpha}R^2 + 220^2\,\alpha^2 LG^2 + 65\alpha\sqrt{N}G^2}{k^2}.$$

We now examine how the optimality gap depends on the number of nodes N and the network connectivity ($\overline{\mu}$). We consider both mD–NC and D–NC, on both random and static networks.

1) Network Dependence – Random Networks: We first consider mD–NC (Theorem 8) and express the optimality gap in terms of $\mathcal{K}$. Fix a small positive number ξ. Using $\log(N(k+1)) \le \xi^{-1}(N(k+1))^\xi$, and $1/(-\log\overline{\mu}) \le 1/(1-\overline{\mu})$, obtain: $\mathcal{K} \le \frac{3}{\xi}\frac{(k+1)^{1+\xi}N^\xi}{(1-\overline{\mu})}$, and therefore, letting $A(\xi) := (3/\xi)^{1/(1+\xi)}$: $\frac{1}{(k+1)^2} \le A^2(\xi)\frac{N^{2\xi/(1+\xi)}}{(1-\overline{\mu})^{2/(1+\xi)}\mathcal{K}^{2/(1+\xi)}} \le A^2(\xi)\frac{N^{2\xi}}{(1-\overline{\mu})^2\mathcal{K}^{2-2\xi}}$. Substituting this in Theorem 8, and replacing ξ with $\xi/2$ ($\xi \in (0,1)$ is arbitrary), we obtain that, after $\mathcal{K}$ communication rounds, $(1/N)(\mathbb{E}[f_i(x_i)] - f^\star)$ is upper bounded by $\mathcal{C}/\mathcal{K}^{2-\xi}$, where the constant $\mathcal{C} = A^2(\xi/2)\frac{N^\xi}{(1-\overline{\mu})^2}\left(\frac{2R^2}{\alpha} + 220^2\alpha^2 LG + 65\alpha\sqrt{N}G^2\right)$ captures the effect of N and $\overline{\mu}$. Ignoring R, L, G, we obtain the network dependence for mD–NG: $\mathcal{C} = O\left(\frac{N^\xi}{(1-\overline{\mu})^2}\left(\frac{1}{\alpha} + \alpha^2 + \alpha\sqrt{N}\right)\right)$. Further, setting the optimized step size $\alpha = 1/(2LN^{1/4})$: $\mathcal{C} = O\left(\frac{N^{1/4+\xi}}{(1-\overline{\mu})^2}\right)$. We specialize the above for the standard random networks (chain, ring, geometric, expander) that we set-up in Section III-B. We obtain that $\mathcal{C} = O(N^{17/4+\xi})$ (chain/ring); $O(N^{9/4+\xi})$ (geometric); and $O(N^{1/4+\xi})$ (expander). For D–NC, convergence rate under the random network model studied in this paper has not been established.

2) Network Dependence – Static Networks: For mD–NC, the network dependence on static networks slightly improves over that of random networks – the small constant $\xi > 0$ can be set to zero. Algorithm D–NC has the same network dependence as mD–NC (on static networks).

Proofs of Theorems 7 and 8

We now prove the convergence rate results for mD–NC.

Proof of Theorem 7: For simplicity, we prove for $d = 1$, but the proof extends to generic $d > 1$. Similarly to Theorem 4, we proceed in three steps. In Step 1, we derive the dynamics for the disagreement $\widetilde{z}(k) = \left(\widetilde{y}(k)^\top, \widetilde{x}(k)^\top\right)$. In Step 2, we unwind the disagreement equation and express $\widetilde{z}(k)$ in terms of the $\widetilde{\Psi}(k,t)$'s in (39) and $\mathcal{B}(k,t)$ in (16). Step 3 finalizes the proof using bounds previously established on the norms of $\widetilde{\Psi}(k,t)$ and $\mathcal{B}(k,t)$.

Step 1. Disagreement Dynamics: We write the dynamic equation for $\widetilde{z}(k)$. Recall $B(k)$ in (15). Multiplying (44)–(45) from the left by $(I-J)$, and using $(I-J)\mathcal{W}(k) = \widetilde{\mathcal{W}}(k)(I-J)$, obtain for $k = 1, 2, \ldots$

$$\widetilde{z}(k) = \left(B(k) \otimes \widetilde{\mathcal{W}}(k)\right)\left(\widetilde{z}(k-1) + u'(k-1)\right), \tag{47}$$

and $\widetilde{z}(0) = 0$, where

$$u'(k-1) = -\begin{bmatrix} \alpha\,(I-J)\nabla F(y(k-1)) \\ 0 \end{bmatrix}. \tag{48}$$

Step 2: Unwinding the Recursion (47): Recall $\mathcal{B}(k,t)$ in (16). Unwinding (47) and using $(A\otimes B)(C\otimes D) = (AC)\otimes(BD)$, obtain for $k = 1, 2, \ldots$

$$\widetilde{z}(k) = \sum_{t=0}^{k-1}\left(\mathcal{B}(k, k-t-2)B(t+1) \otimes \widetilde{\Psi}(k,t)\right)u'(t). \tag{49}$$

The quantities $u'(t)$ and $\widetilde{\Psi}(k,t)$ in (49) are random, while the $\mathcal{B}(k,k-t-2)$'s are deterministic.

Step 3: Finalizing the Proof: Consider $u'(t)$ in (48). By Assumption 5, $\|\nabla F(y(t))\| \le \sqrt{N}G$. From this, obtain $\|u'(t)\| \le \alpha\sqrt{N}\,G$, for any random realization of $u'(t)$. We prove the right inequality in (46). Consider $\|\widetilde{z}(k)\|^2$. Get from (49):

$$\begin{aligned}
&\|\widetilde{z}(k)\|^2 \\
&= \sum_{t=0}^{k-1}\sum_{s=0}^{k-1} u'(t)^\top\left(B(t+1)^\top\mathcal{B}(k,k-t-2)^\top \otimes \widetilde{\Psi}(k,t)^\top\right) \\
&\quad \times \left(\mathcal{B}(k,k-s-2)B(s+1) \otimes \widetilde{\Psi}(k,s)\right)u'(s) \\
&= \sum_{t=0}^{k-1}\sum_{s=0}^{k-1} u'(t)^\top\left(B(t+1)^\top\mathcal{B}(k,k-t-2)^\top\right. \\
&\quad \left.\times\,\mathcal{B}(k,k-s-2)B(s+1)\right) \otimes \left(\widetilde{\Psi}(k,t)^\top\widetilde{\Psi}(k,s)\right)u'(s),
\end{aligned}$$

where the last equality uses $(A\otimes B)(C\otimes D) = (AC)\otimes(BD)$. By the sub-additive and sub-multiplicative properties, $\|A\otimes B\| \le \|A\|\|B\|$, and $\|B(t+1)\| \le 3$, $\forall t$:

$$\begin{aligned}
\|\widetilde{z}(k)\|^2 \le\; & 9\sum_{t=0}^{k-1}\sum_{s=0}^{k-1}\|\mathcal{B}(k,k-t-2)\|\,\|\mathcal{B}(k,k-s-2)\| \\
& \left\|\widetilde{\Psi}(k,t)^\top\,\widetilde{\Psi}(k,s)\right\|\,\|u'(t)\|\,\|u'(s)\| \\
\le\; & 9\sum_{t=0}^{k-1}\sum_{s=0}^{k-1}\left(8(t+1)+5\right)\left(8(s+1)+5\right) \\
& \left\|\widetilde{\Psi}(k,t)^\top\widetilde{\Psi}(k,s)\right\|\,\alpha^2\,NG^2,
\end{aligned}$$

where the last inequality uses $(k-s-1)(s+1)/k \leq s+1$, Lemma 3 and $\|u'(t)\| \leq \alpha\sqrt{N}G$. Taking expectation and applying Lemma 6, we obtain:

$$
\begin{aligned}
&\mathbb{E}\left[\|\widetilde{z}(k)\|^2\right]\\
&\leq 9\left(\alpha^2 N G^2\right)\sum_{t=0}^{k-1}\sum_{s=0}^{k-1}(8(t+1)+5)\\
&(8(s+1)+5)\left(\frac{1}{k^3\cdots(t+1)^3}\right)\left(\frac{1}{k^3\cdots(s+1)^3}\right)\\
&= 9\left(\alpha^2\, N G^2\right)\left(\sum_{t=0}^{k-1}(8(t+1)+5)\,\frac{1}{k^3\cdots(t+1)^3}\right)^2\\
&\leq 9(\alpha^2 N G^2)\left(8(\frac{1}{k^2}+\frac{k-1}{k^3})+\frac{5k}{k^3}\right)^2 \leq \frac{65^2\,\alpha^2\,N G^2}{k^4}.
\end{aligned}
$$

Thus, the right inequality in (46). The left inequality follows by: $\mathbb{E}^2[\|\|\widetilde{z}(k)\|\|] \leq \mathbb{E}[\|\|\widetilde{z}^2(k)\|\|]$. Theorem 7 is proved.

Proof Outline of Theorem 8: We outline the proof since similar to Theorem 8 in [3] (arxiv version v2). Consider the global averages $\overline{x}(k)$ and $\overline{y}(k)$ as in mD–NG. Then, $\overline{x}(k)$ and $\overline{y}(k)$ follow (31)–(32), with $L_{k-1} := N/\alpha$ and $\widehat{g}_{k-1}$ as in (30). Inequalities (33) hold with $L_{k-1} := N/\alpha$ and $\widehat{f}_{k-1}, \widehat{g}_{k-1}, \delta_{k-1}$ as in (30). Applying Lemma 2 in [3] gives:

$$
\begin{aligned}
\frac{1}{N}\left(f(\overline{x}(k))-f^\star\right) \leq{}& \frac{2}{\alpha\, k^2}\|\overline{x}(0)-x^\star\|^2\\
&+\frac{L}{N\,k^2}\sum_{t=1}^{k}\|\widetilde{y}(t-1)\|^2(t+1)^2.
\end{aligned}
$$

(Compare the last equation with (35).) The remainder of the proof proceeds analogously to that of Theorem 5.

III. Discussion and Extensions

We discuss extensions and corollaries: 1) relax the prior knowledge on $L, \overline{\mu}$, and N for both mD–NG and mD–NC; 2) establish rates in the convergence in probability of mD–NG and mD–NC; 3) show almost sure convergence with mD–NC; and 4) establish a convergence rate in the second moment with both methods.

A. Relaxing Knowledge of $L, \overline{\mu}$, and N

Algorithm mD–NG requires only knowledge of L to set the step-size $\alpha_k = c/(k+1)$, $c \leq 1/(2L)$. We show that the rate $O(\log k/k)$ (with a deteriorated constant) still holds if nodes use arbitrary $c > 0$. Initialize all nodes to $x_i(0) = y_i(0) = 0$, suppose that $c > 1/(2L)$, and let $k' = 2\,c\,L$. Applying Lemma 2 in [3], as in the proof of Theorem 5 (b) in [3], for all $k > k'$, a.s.:

$$
\begin{aligned}
\frac{(k+1)^2-1}{k+1}\left(f(\overline{x}(k))-f^\star\right) \leq{}& k'\left(f(\overline{x}(k'-1))-f^\star\right)\\
&+\frac{2N}{c}\left(2\|\overline{v}(k'-1)\|^2+2\|x^\star\|^2\right)\\
&+\sum_{t=1}^{k}\frac{(t+1)^2}{t}L\|\widetilde{y}(t-1)\|^2.
\end{aligned}
\tag{50}
$$

Further, from the proof of Theorem 5 (b) in [3], a.s.:

$$
\|\overline{v}(k'-1)\|^2 \leq (2k'+1)^2\left(3^{k'}\right)^2 4\,c^2\,G^2. \tag{51}
$$

Finally, Theorem 4 holds unchanged for $c > 1/(2L)$. Likewise, one can show $\frac{1}{N}(f(\overline{x}(k'-1))-f^\star) \leq L\left(3^{2k'}4c^2G^2+\|x^\star\|^2\right)$. Thus, $\sum_{t=1}^{k}\frac{(t+1)^2}{t}L\mathbb{E}\left[\|\widetilde{y}(t-1)\|^2\right] = O(\log k)$. Multiplying (50) by $\frac{1}{N}\frac{k+1}{(k+1)^2-1}$, taking expectation on the resulting inequality, and applying Theorem 4, obtain the desired $O(\log k/k)$ rate.

Algorithm mD–NC uses the constant step-size $\alpha \leq 1/(2L)$ and τ_k in (37). To avoid the use of $L, \overline{\mu}$, and N, we set in mD-NC: 1) a diminishing step-size $\alpha_k = 1/(k+1)^p, p \in (0,1]$; and 2) $\tau_k = k$ (as proposed in [28]). We show the adapted mD–NC achieves rate $O(1/k^{2-p})$. Let $k'' = (2L)^{1/p}$. Then, by Lemma 2 in [3], $\forall k \geq k''$, a.s.:

$$
\begin{aligned}
\frac{(k+1)^2-1}{(k+1)^p}\left(f(\overline{x}(k))-f^\star\right) \leq{}& (k')^{2-p}\left(f(\overline{x}(k'-1))-f^\star\right)\\
&+2N\left(2\|\overline{v}(k'-1)\|^2+2\|x^\star\|^2\right)\\
&+\sum_{t=1}^{k}\frac{(t+1)^2}{t^p}L\|\widetilde{y}(t-1)\|^2.
\end{aligned}
\tag{52}
$$

Further, (51) holds here as well (a.s.) with $c = 1$. Modify the argument on the sum in (52). By Lemma 1 and $\tau_k = k$, we have: $\mathbb{E}\left[\|\widetilde{\mathcal{W}}(k)\|^2\right] \leq N\overline{\mu}^{2\,k}$. From this, $\forall k \geq k''' := \max\left\{\left(\frac{\log N+6}{-2\log\overline{\mu}}\right)^2, 3\right\}$: $\mathbb{E}\left[\|\widetilde{\mathcal{W}}(k)\|^2\right] \leq \frac{1}{k^6}$. Next, consider

$$
\begin{aligned}
&\widetilde{\Psi}(k,s)^\top\widetilde{\Psi}(k,t)\\
&\qquad= \left(\widetilde{\mathcal{W}}(k)\cdots\widetilde{\mathcal{W}}(s+1)\right)^\top\left(\widetilde{\mathcal{W}}(k)\cdots\widetilde{\mathcal{W}}(t+1)\right),
\end{aligned}
$$

for arbitrary $k \geq k'''$, and arbitrary $s, t \in \{0, 1, \ldots, k-1\}$. Clearly, $\left\|\widetilde{\Psi}(k,s)^\top\widetilde{\Psi}(k,t)\right\| \leq \left\|\widetilde{\mathcal{W}}(k)\right\|^2$, and hence:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\widetilde{\Psi}(k,s)^\top\widetilde{\Psi}(k,t)\right\|\right] \leq \frac{1}{k^6},\\
&\qquad\qquad \forall s,t \in \{0,1,\ldots,k-1\}, k \geq k'''.
\end{aligned}
$$

Now, from step 3 of the proof of Theorem 7, the above implies: $\mathbb{E}\left[\|\widetilde{y}(k)\|^2\right] \leq \mathbb{E}\left[\|\widetilde{z}(k)\|^2\right] \leq \frac{C}{k^4}$, for all $k \geq k'''$, where $C > 0$ is independent of k. Hence, we obtain the desired bound on the sum: $\sum_{t=1}^{\infty}\frac{(t+1)^2}{t^p}L\mathbb{E}\left[\|\widetilde{y}(t-1)\|^2\right] = O(1)$. Using this, (51), multiplying (52) by $\frac{1}{N}\frac{(k+1)^p}{(k+1)^2-1}$, and taking expectation in (52), obtains the rate $O(1/k^{2-p})$.

B. Convergence in Probability and Almost Sure Convergence

Through the Markov inequality, Theorems 5 and 8 imply, for any $\epsilon > 0, \forall i$, when $k \to \infty$:

$$
\begin{aligned}
&\mathrm{mD-NG}: \mathbb{P}\left(k^{1-\xi}\left(f(x_i(k))-f^\star\right) > \epsilon\right) \to 0\\
&\mathrm{mD-NC}: \mathbb{P}\left(k^{2-\xi}\left(f(x_i(k))-f^\star\right) > \epsilon\right) \to 0,
\end{aligned}
$$

where $\xi > 0$ is arbitrarily small. Furthermore, by the arguments in, e.g., ([40], Section IV-A), with mD–NC, we have that, $\forall i$, $f(x_i(k)) - f^\star \to 0$, almost surely.

C. Convergence Rates in the Second Moment

Consider the following special case of the random network model in Assumptions 1 and 2. Let $G(k)$ be the random graph that supports a random instantiation of $W(k)$: $G(k) = (\mathcal{N}, E(k))$, with $E(k) = \{\{i,j\} : W_{ij}(k) > 0,\, i < j\}$. We assume $G(k)$ is connected with positive probability. This holds with spatio-temporally independent link failures, but not with pairwise gossip. We establish the bounds on the second moment of the optimality gaps:

$$\mathrm{mD-NG}: \mathbb{E}\left[\left(f(x_i(k)) - f^\star\right)^2\right] = O\left(\frac{\log^2 k}{k^2}\right),\ \forall i, \quad (53)$$

$$\mathrm{mD-NC}: \mathbb{E}\left[\left(f(x_i(k)) - f^\star\right)^2\right] = O\left(\frac{1}{k^4}\right),\ \forall i, \quad (54)$$

where (54) holds for mD–NC with a modified value of τ_k (see Appendix C.) We interpret (53), while (54) is similar. Result (53) shows that, not only the mean of the optimality gap decays as $O(\log k/k)$ (by Theorem 5), but also its standard deviation is $O(\log k/k)$.

IV. Simulation Example

We compare mD–NG and mD–NC, D–NG and D–NC in [3], and the methods in [1]. We initialize all to $x_i(0) = y_i(0) = 0$, $\forall i$. We generate one sample path (simulation run), and estimate the average normalized optimality gap $\mathbf{err_f} = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)-f^\star}{f(0)-f^\star}$ versus the total number $\mathcal{K}'$ of scalar transmissions, across all nodes. We count both the successful and failed transmissions. All our plots are in $\log_{10} - \log_{10}$ scales.

A. Setup

Consider a connected geometric supergraph $\mathcal{G} = (\mathcal{N}, E)$ generated by placing 10 nodes at random on a unit 2D square and connecting the nodes whose distance is less than a prescribed radius (26 links). We consider random and static networks. With the random graph, nodes fail with probability .9 For online links $\{i,j\} \in E$, the weights $W_{ij}(k) = W_{ji}(k) = 1/N = 1/10$ and $W_{ii}(k) = 1 - \sum_{j \in O_i(k) - \{i\}} W_{ij}(k), \forall i$. The static network has the same supergraph $\mathcal{G}$, and, $\forall \{i,j\} \in E$, we set $W_{ij} = W_{ji} = 1/N$. With D–NG and mD–NG, the step-size is $\alpha_k = 1/(k+1)$, while with D–NC and mD–NC, $\alpha = 1/2$ and with [1], we use $\alpha_k = 1/\sqrt{k}$. With random networks, for both variants of D–NC, we set τ_k as in (37); with static networks, we use $\tau_k = \left\lceil \frac{3\log k}{-\log \overline{\mu}} \right\rceil$. (As indicated in Section IV, the $\log N$ term is not needed with static networks.)

We use Huber loss cost functions $f(x) = (1/2)\|x - \theta_i\|^2$, if $\|x - \theta_i\| \le 1$, and $f_i(x) = \|x - \theta_i\| - 1/2$, else, $\theta_i \in \mathbb{R}$. (See example 2 in Section I-A.)

B. Results: Link Failures

Fig. 1 (top) shows that the convergence rates (slopes) of mD-NG, mD-NC, and D-NC, are better than that of the method in [1]. All methods converge, even with severe link failures, while D-NG diverges, see Fig. 1 (second from top plot).

C. Results: Static Network

Fig. 1 (second from bottom) shows mD-NG, mD-NC, D-NG, D-NC, and the method in [1] on a static network. As expected with a static network, D-NG performs slightly better

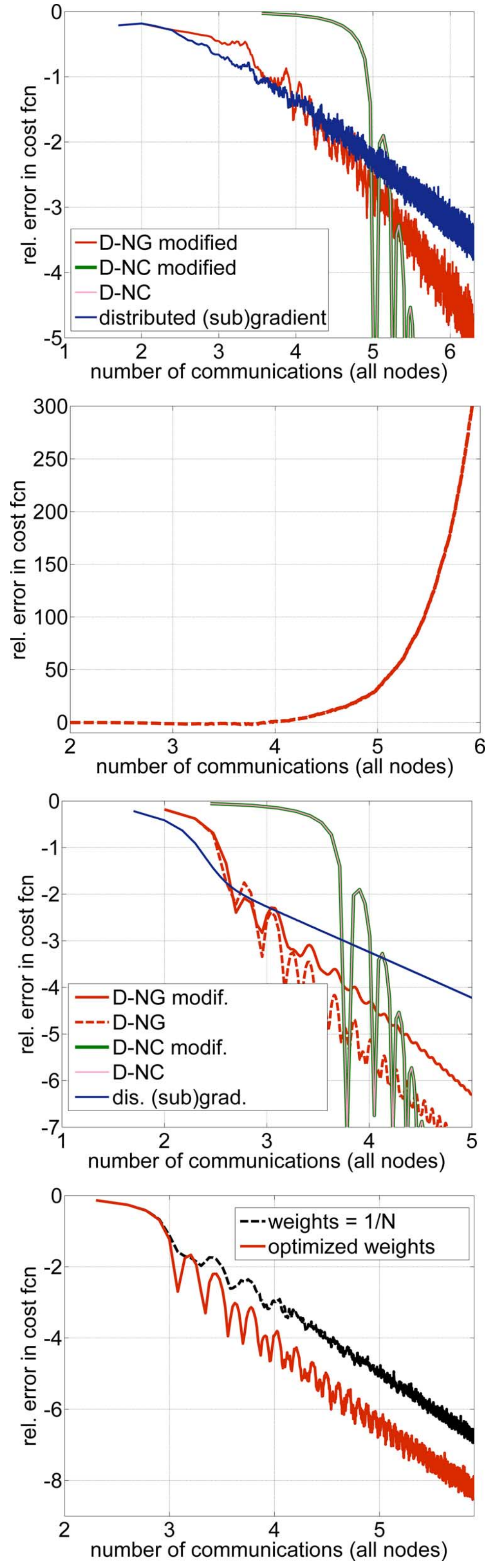


Fig. 1. (Three top plots) Average normalized optimality gap $\mathbf{err_f}$ vs. total number $\mathcal{K}'$ of scalar transmissions, across all nodes ($\log_{10} - \log_{10}$ scale), for $N = 10$-node network. Two top plots: link failures; Second from bottom plot: Static network. Bottom plot: Compare mD–NG method with different weight assignments on an $N = 20$-node network; Red, solid line: optimized weights according to [39]; black, dotted line: $w_{ij} = 1/N, \forall \{i,j\} \in E$.

than mD-NG, and both converge faster than [1]. D-NC and mD-NC perform similarly on both static and random networks. We also compared mD-NG and D-NG when mD-NG is run with Metropolis weights W, [41], while D-NG, because it requires positive definite weights, is run with positive definite $W' = \frac{1.01}{2} I + \frac{0.99}{2} W$. We report that D-NG performs marginally better than mD-NG (Figure omitted for brevity.)

D. Weight Optimization

Fig. 1 shows $\mathbf{err_f}$ versus $\mathcal{K}'$ for uniform weights $1/N$ and the optimized weights in [39] on a 20-node, 91-edge geometric graph (radius $\delta_0 = .55$). Links fail independently in time and space with probabilities $P_{ij} = 0.5 \times \frac{\delta_{ij}^2}{\delta_0^2}$. Here δ_{ij} is the distance between i and j. The losses are Huber $\theta_i = \pm 4(1+\nu_i)$: $+$ for nodes $i = 1 \cdots 7$ and $-$ for $i = 8 \cdots 20$. The ν_i's are as before. The two plots have the same rates (slopes). The optimized weights lead to better convergence constant (agreeing with Theorem 5), reducing the communication cost for the same accuracy.

V. Conclusion

We considered distributed optimization over random networks where N nodes minimize the sum $\sum_{i=1}^N f_i(x)$ of their individual convex costs. We model the random network by a sequence $\{W(k)\}$ of independent, identically distributed random matrices that take values in the set of symmetric, stochastic matrices with positive diagonals. The f_i's are convex and have Lipschitz continuous and bounded gradients. We present mD–NG and mD–NC that are resilient to link failures. We establish their convergence in terms of the expected optimality gap of the cost function at arbitrary node i: mD–NG achieves rates $O(\log k/k)$ and $O(\log \mathcal{K}/\mathcal{K})$, where k is the number of per-node gradient evaluations and $\mathcal{K}$ is the number of per-node communications; and mD–NC has rates $O(1/k^2)$ and $O(1/\mathcal{K}^{2-\xi})$, with $\xi > 0$ arbitrarily small. Simulation examples with link failures and Huber loss functions illustrate our findings.

Appendix

Proofs of Lemmas 1 and 6:

Proof of Lemma 1: We prove (7). For $t = k-1$, $\widetilde{\Phi}(k,t) = I$ and (7) holds. Fix t, $0 \le t \le k-2$. For $N \times N$ matrix A: $\|A\|^2 \le \sum_{i=1}^N \|A\, e_i\|^2$. Applying this for $A = \widetilde{\Phi}(k,t)$, taking expectation:

$$\mathbb{E}\left[\left\|\widetilde{\Phi}(k,t)\right\|^2\right] \le \sum_{i=1}^N \mathbb{E}\left[\left\|\widetilde{\Phi}(k,t)\, e_i\right\|^2\right]. \qquad (55)$$

Next, following proofs in ([36], Section II-B):

$$\mathbb{E}\left[\|\widetilde{\Phi}(k,t)\, e_i\|^2\right] \le (\overline{\mu}^2)^{k-t-1}\, \|e_i\|^2 = (\overline{\mu}^2)^{k-t-1}.$$

Plugging this in (55), (7) follows. Next, (6) follows from (7) and Jensen's inequality. To prove (8), consider $0 \le s < t \le k-2$ ($t < s$ by symmetry). By the independence of the $\widetilde{W}(k)$'s, the sub-multiplicative property of norms, and taking expectation, obtain:

$$\begin{aligned}\mathbb{E}\left[\left\|\widetilde{\Phi}(k,s)^\top \widetilde{\Phi}(k,t)\right\|\right] &\le \mathbb{E}\left[\|\widetilde{W}(t+1)\cdots\widetilde{W}(s+2)\|\right] \\ &\quad \times \mathbb{E}\left[\left\|\widetilde{\Phi}(k,t)\right\|^2\right] \\ \le \left(N^{1/2}\overline{\mu}^{t-s}\right)&\left(N(\overline{\mu}^2)^{k-t-2}\right) = N^{3/2}\overline{\mu}^{(k-t-1)+(k-s-1)}.\end{aligned} \qquad (56)$$

We applied (6) and (7) to get (56); thus, (8) for $s, t \in \{0, \ldots, k-2\}$. If $s = k-1, t < k-1$, $\widetilde{\Phi}(k,s)^\top \widetilde{\Phi}(k,t) = \widetilde{\Phi}(k,t)$ and the result reduces to (7). The case $s < k-1, t = k-1$ is symmetric. Finally, if $s = k-1, t = k-1$, the result is trivial. The proof is complete. ■

Proof of Lemma 6: We prove (40). By (39), $\mathcal{W}(k)$ is the product of τ_k i.i.d. matrices $W(t)$ that obey Assumptions 1 and 2. Hence, by (7), obtain (40):

$$\mathbb{E}\left[\|\mathcal{W}(k)\|^2\right] \le N(\overline{\mu}^2)^{\tau_k} \le N e^{-2(3\log k + \log N)} \le \frac{1}{k^6}.$$

We prove (42). Let $\widetilde{\Psi}(k,t) := \widetilde{\mathcal{W}}(k), \ldots, \widetilde{\mathcal{W}}(t+1), k \ge t+1$. For square matrices A, B: $\|B^\top A^\top A B\| \le \|A^\top A\| \|B^\top B\| = \|B\|^2\, \|A\|^2$. Applying it $k-t$ times, obtain:

$$\left\|\widetilde{\Psi}(k,t)^\top \widetilde{\Psi}(k,t)\right\| \le \left\|\widetilde{\mathcal{W}}(k)\right\|^2 \ldots \left\|\widetilde{\mathcal{W}}(t+1)\right\|^2.$$

Using independence, taking expectation, and applying (40), obtain (42). By Jensen's inequality, (41) follows from (42); relation (43) is proved similarly. ■

Proofs of Lemmas 2 and 3:

Proof of Lemma 2: Let $\frac{d}{dr}h(r)$ be the derivative of $h(r)$. Then (13) follows from:

$$\begin{aligned}\sum_{t=1}^k r^t\, t &= r\,\frac{d}{dr}\left(\sum_{t=1}^k r^t\right) = r\,\frac{d}{dr}\left(\frac{r - r^{k+1}}{1-r}\right) \\ &= \frac{r\left(1-(k+1)r^k(1-r) - r^{k+1}\right)}{(1-r)^2} \\ &\le \frac{r}{(1-r)^2}, \quad \forall k = 1, 2, \ldots\end{aligned}$$

To obtain (14), use (13) and $k/(t+1) \le k-t$:

$$\begin{aligned}\sum_{t=0}^{k-1} r^{k-t-1}\frac{1}{t+1} &= \frac{1}{k}\sum_{t=0}^{k-1} r^{k-t-1}\frac{k}{t+1} \le \frac{1}{k\,r}\sum_{t=0}^{k-1} r^{k-t}(k-t) \\ &\le \frac{1}{k}\,\frac{1}{(1-r)^2}.\end{aligned}$$

■

Proof of Lemma 3: We prove Lemma 3 by first establishing two auxiliary results (Lemmas 9 and 10). Once these are established, we finish by proving Lemma 3.

Lemma 9 (Products $\mathcal{B}(k,t)$): Let $k \ge 3$, $\mathcal{B}(k,t)$ in (16), $a_t := 3/(t+3)$, $t = 0, 1, \ldots$, and:

$$B_1 := \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \quad B_2 := \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \quad B_3 := \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}. \qquad (57)$$

Then, for $t = 1, 2, \ldots, k-2$, with $\sigma_2(k,t)$ and $\sigma_3(k,t)$ as below:

$$
\begin{aligned}
\mathcal{B}(k,t) &= B_1^{t+1} - \sigma_2(k,t)\,B_2 - \sigma_3(k,t)\,B_3 && (58)\\
\sigma_2(k,t) &= a_{k-t-1}\,t + a_{k-t}\beta_{k-t-1}\,(t-1)\\
&\quad + a_{k-t+1}\beta_{k-t}\beta_{k-t-1}(t-2) +\\
&\quad \cdots + a_{k-2}\beta_{k-3}\cdots\beta_{k-t-1} && (59)\\
\sigma_3(k,t) &= a_{k-t-1} + a_{k-t}\beta_{k-t-1}\\
&\quad + a_{k-t+1}\beta_{k-t}\beta_{k-t-1} +\\
&\quad \cdots + a_{k-2}\beta_{k-3}\cdots\beta_{k-t-1}. && (60)
\end{aligned}
$$

Proof: The proof is by mathematical induction on $t = 1, 2, \ldots, k-2$. Using $B_3 B_1 = B_3$, $B_1 B_3 = B_2 + B_3$, $B_3^2 = B_3$, and $1 - a_{k-2} = \beta_{k-2}$, it is easy to verify that the claim for $t = 1$ holds, i.e.,

$$
\begin{aligned}
\mathcal{B}(k,1) &= B_1^2 - a_{k-2}B_2 - (a_{k-2} + a_{k-1}\beta_{k-2})B_3\\
&= B_1^2 - \sigma_2(k,1)\,B_2 - \sigma_3(k,1)\,B_3.
\end{aligned}
$$

Now, suppose the claim holds for some fixed t, $t \in \{1, \ldots, k-3\}$. Using the inductive hypothesis and the definition of $\mathcal{B}(k, t+1)$:

$$
\begin{aligned}
&\mathcal{B}(k,t+1)\\
&\quad = \mathcal{B}(k,t)\,B(k-t-1)\\
&\quad = \left(B_1^{t+1} - \sigma_2(k,t)B_2 - \sigma_3(k,t)B_3\right)\left(B_1 - a_{k-t-2}B_3\right)\\
&\quad = B_1^{t+2} - \sigma_2(k,t)B_2 - \sigma_3(k,t)B_3 && (61)\\
&\quad -a_{k-t-2}((t+1)B_2 + B_3) + \sigma_2(k,t)a_{k-t-2}B_2\\
&\qquad + \sigma_3(k,t)a_{k-t-2}B_3\\
&\quad = B_1^{t+2} - (\sigma_2(k,t)\beta_{k-t-2} + (t+1)a_{k-t-2})B_2\\
&\qquad - (\sigma_3(k,t)\beta_{k-t-2} + a_{k-t-2})B_3. && (62)
\end{aligned}
$$

Equality (61) uses $B_2 B_1 = B_2$, $B_3^2 = B_3$, $B_2 B_3 = B_2$, and the fact that $B_1^{t+1} B_3 = (t+1)B_2 + B_3$. (This is trivial to show by mathematical induction on t.) Next, recognize from (59)–(60) that $\sigma_2(k,t+1) = \sigma_2(k,t)\beta_{k-t-2} + (t+1)a_{k-t-2}$, and $\sigma_3(k,t+1) = \sigma_3(k,t)\beta_{k-t-2} + a_{k-t-2}$. Thus, the claim for $t+1$ and the proof is complete. ■

We establish bounds on the sums $\sigma_2(k,t)$ and $\sigma_3(k,t)$.

Lemma 10: Let $\sigma_2(k,t)$ and $\sigma_3(k,t)$ in (59)–(60), $t = 1, \ldots, k-2$, $k \geq 3$. Then:

$$
\frac{t^2}{k+2} \leq \sigma_2(k,t) \leq t+1, \qquad 0 \leq \sigma_3(k,t) \leq 1. \tag{63}
$$

Proof: We prove each of the four inequalities above.

Proof of the Right Inequality on $\sigma_2(k,t)$: By induction on $t = 1, \ldots, k-2$. The claim holds for $t = 1$, since $\sigma_2(k,1) = a_{k-2} = 3/(k+1) \leq 1 + 1$, $\forall k$. Let it be true for some $t \geq 1$. For $t = 1, \ldots, k-3$, write $\sigma_2(k,t)$ as:

$$
\sigma_2(k,t+1) = a_{k-t-2}(t+1) + \beta_{k-t-2}\sigma_2(k,t). \tag{64}
$$

Using (64) and the induction hypothesis:

$$
\begin{aligned}
\sigma_2(k,t+1) &\leq (t+1)a_{k-t-2} + \beta_{k-t-2}(t+1)\\
&= (a_{k-t-2} + \beta_{k-t-2})(t+1) = t+1 \leq t+2.
\end{aligned}
$$

Thus, the right inequality on $\sigma_2(k,t)$.

Proof of the Left Inequality on $\sigma_2(k,t)$: Again, by induction on t. The claim holds for $t = 1$, since:

$$
\sigma_2(k,1) = a_{k-2} = \frac{3}{k+1} \geq \frac{1^2}{k+2}.
$$

Let the claim be true for some $t \in \{1, 2, \ldots, k-3\}$, i.e.,:

$$
\sigma_2(k,t) \geq \frac{t^2}{k+2}. \tag{65}
$$

We show that $\sigma_2(k,t+1) \geq \frac{(t+1)^2}{k+2}$. Using (64):

$$
\begin{aligned}
\sigma_2(k,t+1) &\geq a_{k-t-2}(t+1) + \beta_{k-t-2}\frac{t^2}{k+2}\\
&= \frac{(t+1)^2}{k+2} + \frac{t(k-t) + (2k+5t+5)}{(k+2)(k-t+1)}\\
&\geq \frac{(t+1)^2}{k+2},
\end{aligned}
$$

where the last equality follows after algebraic manipulations. By induction, the last inequality completes the proof of the lower bound on $\sigma_2(k,t)$.

Proof of Bounds on $\sigma_3(k,t)$: The lower bound is trivial. The upper bound follows by induction. For $t = 1$:

$$
\sigma_3(k,1) = a_{k-2} + a_{k-1}\beta_{k-2} \leq a_{k-2} + \beta_{k-2} = 1.
$$

Let the claim hold for some $t \in \{1, \ldots, k-3\}$, i.e.,: $\sigma_3(k,t) \leq 1$. From (59): $\sigma_3(k,t+1) = \beta_{k-t-2}\,\sigma_3(k,t) + a_{k-t-2}$. Thus, by the induction hypothesis: $\sigma_3(k,t+1) \leq \beta_{k-t-2} + a_{k-t-2} \leq 1$, completing the proof of the upper bound on $\sigma_3(k,t)$. ■

Proof of Lemma 3: We upper bound $\|\mathcal{B}(k, k-t-2)\|$. Fix some $t \in \{1, \ldots, k-2\}$, $k \geq 3$, and consider $\mathcal{B}(k,t)$ in Lemma 9. Note that $B_1^t = t\,B_2 + I$. Thus,

$$
\mathcal{B}(k,t) = (t + 1 - \sigma_2(k,t))\;B_2 + I - \sigma_3(k,t)B_3. \tag{66}
$$

By Lemma 10, the term: $0 \leq t + 1 - \sigma_2(k,t) \leq t + 1 - t^2/(k+2)$. Using in (66) this equation, $\sigma_3(k,t) \leq 1$ (by Lemma 10), $\|B_2\| = 2$, and $\|B_3\| = \sqrt{2} < 2$, get:

$$
\|\mathcal{B}(k,t)\| \leq 2\left(t + 1 - \frac{t^2}{k+2}\right) + 3 = 2\left(t - \frac{t^2}{k+2}\right) + 5, \tag{67}
$$

for all $t = 1, 2, \ldots, k-2$, $k \geq 3$. Next, from (67), for $t = 0, \ldots, k-3$, $k \geq 3$, get:

$$
\begin{aligned}
&\|\mathcal{B}(k, k-t-2)\|\\
&\quad \leq 2\left(k - t - 2 - \frac{(k-t-2)^2}{k+2}\right) + 5\\
&\quad = 2(k-t-2)\frac{t+4}{k+2} + 5 \leq 8(k-t-1)\frac{t+1}{k} + 5,
\end{aligned}
$$

We used $(t+4)/(k+2) \leq 4(t+1)/k$ and proved (17) for $t = 0, \ldots, k-3$, for $k \geq 3$. To complete the proof, we show that (17) holds also for: 1) $t = k-2$, $k \geq 2$; 2) $t = k-1$, $k \geq 1$. Consider first case 1 and $\mathcal{B}(k, k-2) = B(k-1) = B_1 - a_{k-1}B_3$, $k \geq 2$. We have $\|\mathcal{B}(k,k-2)\| \leq \|B_1\| + \|B_3\| < 5$, and so (17) holds for $t = k-2$, $k \geq 2$. Next, consider case 2 and $\mathcal{B}(k, k-1) = I$,

$k \geq 1$. We have that $\|\mathcal{B}(k,k-1)\| = 1 < 5$, and so (17) also holds for $t = k-1$, $k \geq 1$. This proves the Lemma. ■

Proof of (53) – (54): Recall the random graph $G(k)$. We assume that, for a certain connected graph G_0 with the Laplacian matrix $\mathcal{L}_0$, $G(k) = G_0$ with probability $p_G > 0$. This, with Assumption 1, implies $\exists \mu_4 \in [0,1)$: $\mathbb{E}\left[\left\|\widetilde{W}(k)\right\|^4\right] \leq (\mu_4)^4$; by ([42], Lemma 11), μ_4 can be taken as: $(\mu_4)^4 = (1 - p_G) + p_G\left(1 - \underline{w}^2\,\lambda_2\left(\mathcal{L}_0\right)\right)^2 < 1$. Consider (35). Let $\widetilde{f}_k := \frac{1}{N}(f(\overline{x}(k)) - f^\star)$. Squaring (35), taking expectation, and by the Cauchi-Schwarz inequality:

$$\begin{aligned}
&\mathbb{E}\left[(\widetilde{f}_k)^2\right] \\
&\leq \frac{4R^2}{c^2\,k^2} + \frac{4RL}{N\,c}\frac{1}{k^2}\sum_{t=1}^{k}\frac{(t+1)^2}{t}\mathbb{E}\left[\|\widetilde{y}(t-1)\|^2\right] \\
&\quad + \frac{L^2}{N^2\,k^2}\sum_{t=1}^{k}\sum_{s=1}^{k}\frac{(t+1)^2}{t}\frac{(s+1)^2}{s}\sqrt{\mathbb{E}\left[\,\|\widetilde{y}(t-1)\|^4\,\right]} \\
&\quad \times \sqrt{\mathbb{E}\left[\,\|\widetilde{y}(s-1)\|^4\,\right]}. \qquad (68)
\end{aligned}$$

The first term in (68) is $O(1/k^2)$; by Theorem 4, the second is $O(\log k/k^2)$. Recall (29), let $\mathcal{U}_k := \|\widetilde{W}(k)\|$. Fix $s < t$, $s,t \in \{0,1,\ldots,k-1\}$. Let $\widehat{b}(k,t) := \frac{8(k-t-1)(t+1)}{k} + 5$, and $\widehat{\mathcal{U}}(t,s) := \mathcal{U}_t\mathcal{U}_{t-1}\cdots\mathcal{U}_{s+1}$, for $t > s$, and $\widehat{\mathcal{U}}(t,t) = 1$. From (28) and $\|\widetilde{\Phi}(k,t)\| \leq \mathcal{U}_k \ldots \mathcal{U}_{t+2} = \widehat{\mathcal{U}}(k,t+1)$, one can show (details omitted): $\|\widetilde{z}(k)\| \leq \sqrt{5}c\sqrt{N}G\sum_{t=0}^{k-1}\widehat{b}(k,t)\widehat{\mathcal{U}}(k,t+1)(t+1)^{-1}$, and:

$$\begin{aligned}
\mathbb{E}\left[\|\widetilde{z}(k)\|^4\right] &\leq (25c^4N^2G^4)\left(\sum_{t=0}^{k-1}\widehat{b}(k,t)\mu_4^{k-t-1}(t+1)^{-1}\right)^4 \\
&= O(1/k^4),
\end{aligned}$$

and so the third term in (68) is $O(\log^2 k/k^2)$. Thus, $\mathbb{E}\left[(\widetilde{f}_k)^2\right] = O(\log^2 k/k^2)$. Further, one can show $(f(x_i(k)) - f^\star)^2 \leq 2(N\widetilde{f}_k)^2 + 2G^2N^2\|\widetilde{x}(k)\|^2$. Taking expectation and applying Theorem 4, the result (53) follows. For mD–NC, prove (54) like (53) by letting $\tau_k = \lceil\frac{3\log k}{-\log\mu_4}\rceil$.

REFERENCES

[1] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[2] J. Duchi, A. Agarwal, and M. Wainwright, “Dual averaging for distributed optimization: Convergence and network scaling,” *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[3] D. Jakovetic, J. Xavier, and J. M. F. Moura, “Fast distributed gradient methods,” *IEEE Trans. Autom. Control*, 2014, to appear.

[4] C. Lopes and A. H. Sayed, “Adaptive estimation algorithms over distributed networks,” presented at the 21st IEICE Signal Process. Symp., Kyoto, Japan, Nov. 2006.

[5] F. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[6] S. Kar, J. M. F. Moura, and K. Ramanan, “Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication,” *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3575–3605, Jun. 2012.

[7] F. Cattivelli and A. H. Sayed, “Distributed detection over adaptive networks using diffusion adaptation,” *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 1917–1932, May 2011.

[8] D. Bajovic, D. Jakovetic, J. Xavier, B. Sinopoli, and J. M. F. Moura, “Distributed detection via Gaussian running consensus: Large deviations asymptotic analysis,” *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4381–4396, Sep. 2011.

[9] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, “Diffusion strategies for adaptation and learning over networks,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[10] G. Mateos, J. A. Bazerque, and G. B. Giannakis, “Distributed sparse linear regression,” *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5262–5276, Nov. 2010.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends in Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[12] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Proc. 3rd Int. Symp. Inf. Process. in Sens. Netw. (IPSN)*, Berkeley, CA, Apr. 2004, pp. 20–27.

[13] D. Blatt and A. O. Hero, “Energy based sensor network source localization via projection onto convex sets (POCS),” *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3614–3619, 2006.

[14] D. Jakovetic, J. M. F. Moura, and J. Xavier, “Distributed Nesterov-like gradient algorithms,” in *Proc. 51st IEEE Conf. Decision Contr. (CDC)*, Maui, HI, USA, Dec. 2012, pp. 5459–5464.

[15] D. Jakovetic, J. M. F. Moura, and J. Xavier, “Distributed Nesterov-like gradient algorithms,” in *Proc. IEEE Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2012, pp. 1513–1517.

[16] S. S. Ram, A. Nedic, and V. Veeravalli, “Asynchronous gossip algorithms for stochastic optimization,” in *Proc. 48th IEEE Int. Conf. Decision Control (CDC)*, Shanghai, China, Dec. 2009, pp. 3581–3586.

[17] I. Matei and J. S. Baras, “Performance evaluation of the consensus-based distributed subgradient method under random communication topologies,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 754–771, 2011.

[18] M. Zhu and S. Martínez, “On distributed convex optimization under inequality and equality constraints,” *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.

[19] J. A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radio networks by exploiting sparsity,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, Mar. 2010.

[20] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[21] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, “A sparsity promoting adaptive algorithm for distributed learning,” *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[22] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” in *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. La Jolla, CA, USA: Neural Inf. Processing Syst. Foundation/Salk Inst. for Biol. Studies-CNL, 2008, vol. 20, pp. 161–168.

[23] S. Ram, A. Nedic, and V. Veeravalli, “Distributed stochastic subgradient projection algorithms for convex optimization,” *J. Opt. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2011.

[24] I. Lobel, A. Ozdaglar, and D. Feijer, “Distributed multi-agent optimization with state-dependent communication,” *Math. Programm.*, vol. 129, no. 2, pp. 255–284, 2011.

[25] I. Lobel and A. Ozdaglar, “Convergence analysis of distributed subgradient methods over random networks,” in *Proc. 46th Ann. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Sep. 2008, pp. 353–360.

[26] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, “Subgradient methods and consensus algorithms for solving separable distributed control problems,” in *Proc. 47th IEEE Conf. Decision Control (CDC)*, Cancun, Mexico, Dec. 2008, pp. 4185–4190.

[27] K. Tsianos and M. Rabbat, “Distributed consensus and optimization under communication delays,” in *Proc. 49th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Sep. 2011, pp. 974–982.

[28] I.-A. Chen and A. Ozdaglar, “A fast distributed proximal gradient method,” presented at the Allerton Conf. Commun., Control, Comput., Monticello, IL, USA, Oct. 2012.

[29] D. Jakovetic, J. Xavier, and J. M. F. Moura, “Cooperative convex optimization in networked systems: Augmented Lagrangian algorithms with directed gossip communication,” *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 3889–3902, Aug. 2011.

[30] J. Mota, J. Xavier, P. Aguiar, and M. Pueschel, “Distributed basis pursuit,” *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2011.

[31] U. V. Shanbhag, J. Koshal, and A. Nedic, “Multiuser optimization: Distributed algorithms and error analysis,” *SIAM J. Control Optimiz.*, vol. 21, no. 2, pp. 1046–1081, 2011.

[32] H. Terelius, U. Topcu, and R. M. Murray, “Decentralized multi-agent optimization via dual decomposition,” presented at the 18th World Congr Int. Fed. Autom. Control (IFAC), Milano, Italy, Aug. 2011.

[33] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, “Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals,” *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2009.

[34] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, “Asynchronous distributed optimization using a randomized alternating direction method of multipliers,” in *Proc. IEEE 52nd Conf. Decision Control (CDC)*, Florence, Italy, Dec. 2013, [Online].

[35] E. Wei and A. Ozdaglar, “On the $\mathrm{O}(1/\mathrm{k})$ convergence of asynchronous distributed alternating direction method of multipliers,” ArXiv preprint, 2013 [Online]. Available: http://arxiv.org/abs/1307.8254

[36] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[37] A. Dimakis, S. Kar, J. M. F. Moura, M. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[38] Y. E. Nesterov, “A method for solving the convex programming problem with convergence rate $\mathrm{O}(1/k^2)$,” (in Russian) *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[39] D. Jakovetić, J. Xavier, and J. M. F. Moura, “Weight optimization for consensus algorithms with correlated switching topology,” *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3788–3801, Jul. 2010.

[40] A. Tahbaz-Salehi and A. Jadbabaie, “On consensus in random networks,” in *Proc. 44th Annu. Allerton Conf. Commun., Control, Comput.*, Allerton House, IL, USA, Sep. 2006, pp. 1315–1321.

[41] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. Inf. Process. Sens. Netw. (IPSN)*, Los Angeles, CA, 2005, pp. 63–70.

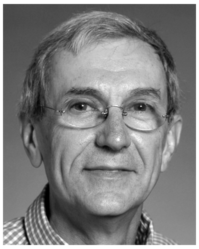
[42] D. Bajovic, J. Xavier, J. M. F. Moura, and B. Sinopoli, “Consensus and products of random stochastic matrices: Exact rate for convergence in probability,” *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2557–2571, May 2013.

Dušan Jakovetić (S’10–M’14) received the dipl. ing. diploma from the School of Electrical Engineering, University of Belgrade, in August 2007, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, and Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal, in May 2013.

Since October 2013, he has been a research fellow at the BioSense Center, University of Novi Sad, Serbia. From June to September 2013, he was a postdoctoral researcher at IST. His research interests include distributed inference and distributed optimization.

João Manuel Freitas Xavier (S’97–M’03) received the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 2002.

Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, IST. He is also a Researcher at the Institute of Systems and Robotics (ISR), Lisbon, Portugal. His current research interests are in the area of optimization and statistical inference for distributed systems.



José M. F. Moura (S’71–M’75–SM’90–F’94) received the engenheiro electrotécnico degree from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and D.Sc. degrees in EECS from MIT, Cambridge, MA.

During 2013-2014, he is a visiting Professor at New York University (NYU) and at CUSP-NYU on sabbatical leave from Carnegie Mellon University (CMU), Pittsburgh, PA, where he is the Philip and Marsha Dowd University Professor. Previously, he was on the faculty at IST and was visiting Professor at MIT. He is founding director of ICTI@CMU, a large education and research program between CMU and Portugal, www.cmuportugal.org. His research interests include statistical, algebraic, and distributed signal and image processing and signal processing on graphs. He has published more than 470 papers, has 10 patents issued by the US Patent Office, and cofounded SpiralGen.

Dr. Moura was the IEEE Division IX Director and member of the IEEE Board of Directors (2012–13) and has served on several IEEE Boards. He was President (2008–2009) of the IEEE Signal Processing Society(SPS), served as Editor in Chief for the IEEE Transactions in Signal Processing, interim Editor in Chief for the IEEE Signal Processing Letters, and member of several Editorial Boards, including the IEEE Proceedings, IEEE Signal Processing Magazine, and the ACM *Transactions on Sensor Networks*. He is member of the US National Academy of Engineering, corresponding member of the Academy of Sciences of Portugal, Fellow of the AAAS. He received the IEEE Signal Processing Society Technical Achievement Award and the IEEE Signal Processing Society Award.