Fast Distributed Gradient Methods

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

***Abstract*—We study distributed optimization problems when N nodes minimize the sum of their individual costs subject to a common vector variable. The costs are convex, have Lipschitz continuous gradient (with constant L), and bounded gradient. We propose two fast distributed gradient algorithms based on the centralized Nesterov gradient algorithm and establish their convergence rates in terms of the per-node communications $\mathcal{K}$ and the per-node gradient evaluations k. Our first method, Distributed Nesterov Gradient, achieves rates $O\left(\log \mathcal{K}/\mathcal{K}\right)$ and $O\left(\log k/k\right)$. Our second method, Distributed Nesterov gradient with Consensus iterations, assumes at all nodes knowledge of L and $\mu(W)$ – the second largest singular value of the $N \times N$ doubly stochastic weight matrix W. It achieves rates $O\left(1/\mathcal{K}^{2-\xi}\right)$ and $O\left(1/k^2\right)$ ($\xi > 0$ arbitrarily small). Further, we give for both methods explicit dependence of the convergence constants on N and W. Simulation examples illustrate our findings.**

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

I. Introduction

DISTRIBUTED computation and optimization have been studied for a long time, e.g., [1], [2], and have received renewed interest, motivated by applications in sensor [3], multi-robot [4], or cognitive networks [5], as well as in distributed control [6] and learning [7]. This paper focuses on the problem where N nodes (sensors, processors, agents) minimize a sum of convex functions $f(x) := \sum_{i=1}^{N} f_i(x)$ subject to a common variable $x \in \mathbb{R}^d$. Each function $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex and known only to node i. The underlying network is generic and connected.

To solve this and related problems, the literature proposes several distributed gradient like methods, including: [8] (see also [9]–[11]); [12] (see also [13]); [14] (see also [3], [15]); and [16]. When the nodes lack global knowledge of the network parameters, [14] establishes, for the distributed dual averaging algorithm therein, rate $O\left(\frac{1}{(1-\mu(W))}\frac{\log(Nk)}{k^{1/2}}\right)$, where k is the number of communicated d-dimensional vectors per node, which also equals the number of iterations (gradient evaluations per node,) and $\mu(W)$ is the second largest singular value of the underlying $N \times N$ doubly stochastic weight matrix W. Further, when $\mu(W)$ is known to the nodes, and after optimizing the step-size, [14] shows the convergence rate to be $O\left(\frac{1}{(1-\mu(W))^{1/2}}\frac{\log(Nk)}{k^{1/2}}\right)$.

1) Setup: The class of functions usually considered in the references above are more general than we consider here, namely, they assume that the f_i's are (possibly) non-differentiable and convex, and: 1) for unconstrained minimization, the f_i's have bounded gradients, while 2) for constrained minimization, they are Lipschitz continuous over the constraint set. In contrast, we assume the class $\mathcal{F}$ of convex f_i's that have Lipschitz continuous and bounded gradients.

It is well established in centralized optimization, [17], that one expects faster convergence rates on classes of more structured functions; e.g., for convex, non-smooth functions, the best achievable rate for centralized (sub)gradient methods is $O(1/\sqrt{k})$, while, for convex functions with Lipschitz continuous gradient, the best rate is $O(1/k^2)$, achieved, e.g., by the Nesterov gradient method [17]. Here k is the number of iterations, i.e., the number of gradient evaluations.

2) Contributions: Building from the centralized Nesterov gradient method, we develop for the class $\mathcal{F}$ two distributed gradient methods and prove their convergence rates, in terms of the number of per-node communications $\mathcal{K}$, the per-node gradient evaluations k, and the network topology. Our first method, the Distributed Nesterov Gradient (D–NG), uses one communication per k (it has $k = \mathcal{K}$) and achieves convergence rate $O\left(\frac{1}{(1-\mu)^{p+\xi}}\left[\frac{\log k}{k} + \frac{\sqrt{N}\log^{1/2} k}{k^{3/2}} + \frac{N}{k^2}\right]\right)$, where $\xi > 0$ is an arbitrarily small quantity, and $p = 3$ when the nodes have no global knowledge of the parameters underlying the optimization problem and the network: L and G the f_i's gradient's Lispchitz constant and the gradient bound, respectively, $\mu := \mu(W)$ the second largest singular value of W, and R a bound on the distance to a solution. When L and μ are known by all, D–NG with optimized step-size achieves the same rate with p reduced to 1.

Our second method, Distributed Nesterov gradient with Consensus iterations (D–NC), assumes global knowledge on μ and L and achieves rates $O\left(\frac{1}{[(1-\mu)\mathcal{K}^{1-\xi}]^2} + \frac{\sqrt{N}}{[(1-\mu)\mathcal{K}^{1-\xi}]^3} + \frac{N}{[(1-\mu)\mathcal{K}^{1-\xi}]^4}\right)$ and $O\left(\frac{1}{k^2} + \frac{\sqrt{N}}{k^3} + \frac{N}{k^4}\right)$. Further, we establish that, for the class $\mathcal{F}$, both our methods (achieving at least $O(\log k/k)$) are strictly better than the distributed (sub)gradient method [8] and the

Manuscript received November 30, 2011; revised August 04, 2012; accepted March 29, 2013. Date of publication January 09, 2014; date of current version April 18, 2014. This work was supported by the Carnegie Mellon|Portugal Program under a grant from the Fundação de Ciência e Tecnologia (FCT) from Portugal, FCT grants CMU-PT/SIA/0026/2009, PTDC/EMS-CRO/2042/2012, SFRH/BD/33518/2008 (through the Carnegie Mellon|Portugal Program managed by ICTI), ISR/IST plurianual funding (POSC program, FEDER), AFOSR grant FA95501010291, and by National Science Foundation (NSF) grant CCF1011903. Recommended by Associate Editor A. Ozdaglar.

D. Jakovetic was with the Institute for Systems and Robotics, Instituto Superior Tecnico (IST), University of Lisbon, Lisbon 1049-001, Portugal, and with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213-3890 USA. He is now with University of Novi Sad, BioSense Center, Novi Sad 21000, Serbia (e-mail: djakovet@uns.ac.rs).

J. Xavier is with the Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST), University of Lisbon, Lisbon 1049-001, 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 15213-3890 USA, and also with CUSP, New York University, Brooklyn, NY 11201 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.

Digital Object Identifier 10.1109/TAC.2014.2298712

distributed dual averaging in [14], even when these algorithms are restricted to functions in $\mathcal{F}$. We show analytically that [8] cannot be better than $\Omega\left(1/k^{2/3}\right)$ and $\Omega\left(1/\mathcal{K}^{2/3}\right)$ (see Section VII-A for details), and by simulation examples that [8] and [14] perform similarly.

3) Distributed Versus Centralized Nesterov Gradient Methods: The centralized Nesterov gradient method does not require bounded gradients – an assumption that we make for our distributed methods. We prove here that if we drop the bounded gradients assumption, the convergence rates that we establish do not hold for either of our algorithms. (It may be possible to *replace* the bounded gradients assumption with a weaker requirement.) In fact, the worst case convergence rates of D–NG and D–NC become arbitrarily slow. (See Section VII-B for details.) This important result illustrates a distinction between the allowed function classes by the centralized and distributed methods. The result is not specific to our accelerated methods; it can be shown that the standard distributed gradient method in [8] is also arbitrarily slow when the assumption of bounded gradients is dropped (while convexity and Lipschitz continuous gradient hold) [18].

Remark: Since we make use here of the bounded gradients assumption, an interesting research direction is to look for a weaker requirement, e.g., boundedness of all $x_i^\star \in \arg\min_{x\in\mathbb{R}^d} f_i(x)$ ($\|x_i^\star\| \leq C < \infty$, $\forall x_i^\star$, $\forall i$.) In fact, with both D–NG and D–NC, we prove elsewhere that we can assume *different* setups (corresponding to broad classes of functions) and still achieve the same convergence rates in terms of k and $\mathcal{K}$. With D–NG, we can replace the bounded gradients assumption with the following: there exists $b, B > 0$ such that, $\forall i$, $f_i(x) \geq b\|x\|$ whenever $\|x\| \geq B$. For a natural extension of D–NC, we can replace the unconstrained problems with Lipschitz continuous and bounded gradients assumed here by a constrained optimization problem (compact, convex constraint set $\mathcal{X}$) where the f_i's have Lipschitz continuous gradient on a certain compact set that includes $\mathcal{X}$. Due to lack of space, these alternatives are pursued elsewhere.

Remark: We comment on references [19] and [20] (see also Section VII-A and [18]). They develop accelerated proximal methods for time varying networks that resemble D–NC. The methods in [19] and [20] use only one consensus algorithm per outer iteration k, while we use two with D–NC. Adapting the results in [19], [20] to our framework, it can be shown that the optimality gap bounds in [19], [20] expressed in terms of $N, 1-\mu(W)$, and $\mathcal{K}$ have the same or worse (depending on the variant of their methods) dependence on $\mathcal{K}$ and $\mu(W)$ than the one we show for D–NC, and a worse dependence on N. (See Section VII-A and [18].)

In addition to distributed gradient methods, the literature also proposes distributed augmented Lagrangian dual or ordinary dual methods [5], [21]–[27]. These are based on the augmented Lagrangian (or ordinary) dual of the original problem. They in general have significantly more complex iterations than the gradient type methods that we consider in this paper, due to solving local optimization problems at each node, at each iteration, but may have a lower total communication cost. Reference [22] uses the Nesterov gradient method to propose an *augmented Lagrangian dual* algorithm but does not analyze its convergence rate. In contrast, ours are *primal gradient algorithms*, with no notion of Lagrangian dual variables, and we establish the convergence rates of our algorithms. References [26], [27] study both the resource allocation and the problems that we consider (see (1)). For (1), [26], [27] apply certain accelerated gradient methods on the *dual problem*, in contrast with our primal gradient methods. Finally, [6] uses the Nesterov gradient algorithm to propose a decomposition method based on a smoothing technique, for a problem formulation different than ours and on the *Lagrangian dual problem*.

4) Paper Organization: The next paragraph introduces notation. Section II describes the network and optimization models that we assume. Section III presents our algorithms, the distributed Nesterov gradient and the distributed Nesterov gradient with consensus iterations, D–NG and D–NC for short. Section IV explains the framework of the (centralized) inexact Nesterov gradient method; we use this framework to establish the convergence rate results for D–NG and D–NC. Sections V and VI prove convergence rate results for the algorithms D–NG and D–NC, respectively. Section VII compares our algorithms D–NG and D–NC with existing distributed gradient type methods, discusses the algorithms' implementation, and discusses the need for our Assumptions. Section VIII provides simulation examples. Finally, we conclude in Section IX. Proofs of certain lengthy arguments are relegated to the Appendix.

Notation: We index by a subscript i a (possibly vector) quantity assigned to node i; e.g., $x_i(k)$ is node i's estimate at iteration k. Further, we denote by: $\mathbb{R}^d$ the d-dimensional real coordinate space; $\mathbf{j}$ the imaginary unit ($\mathbf{j}^2 = -1$); A_{lm} or $[A]_{lm}$ the entry in the l-th row and m-th column of a matrix A; $a^{(l)}$ the l-th entry of vector a; $(\cdot)^\top$ the transpose and $(\cdot)^H$ the conjugate transpose; 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; $\oplus$ and $\otimes$ the direct sum and Kronecker product of matrices, respectively; $\|\cdot\|_l$ the vector (respectively, matrix) l-norm of its vector (respectively, matrix) argument; $\|\cdot\| = \|\cdot\|_2$ the Euclidean (respectively, spectral) norm of its vector (respectively, matrix) argument ($\|\cdot\|$ also denotes the modulus of a scalar); $\lambda_i(\cdot)$ the i-th smallest *in modulus* eigenvalue; $A \succeq 0$ means that a Hermitian matrix A is positive semi-definite; $\lceil a \rceil$ the smallest integer not smaller than a real scalar a; $\nabla\phi(x)$ and $\nabla^2\phi(x)$ the gradient and Hessian at x of a twice differentiable function $\phi : \mathbb{R}^d \to \mathbb{R}$, $d \geq 1$. For two positive sequences a_k and b_k, the following is the standard notation: $b_k = O(a_k)$ if $\limsup_{k\to\infty} \frac{b_k}{a_k} < \infty$; $b_k = \Omega(a_k)$ if $\liminf_{k\to\infty} \frac{b_k}{a_k} > 0$; and $b_k = \Theta(a_k)$ if $b_k = O(a_k)$ and $b_k = \Omega(a_k)$.

II. Problem Model

This section introduces the network and optimization models that we assume.

1) Network Model: We consider a (sparse) network $\mathcal{N}$ of N nodes (sensors, processors, agents,) each communicating only locally, i.e., with a subset of the remaining nodes. The communication pattern is captured by the graph $\mathcal{G} = (\mathcal{N}, E)$, where $E \subset \mathcal{N} \times \mathcal{N}$ is the set of links. The graph $\mathcal{G}$ is connected, undirected and simple (no self/multiple links.)

2) Weight Matrix: We associate to the graph $\mathcal{G}$ a symmetric, doubly stochastic (rows and columns sum to one and all the entries are non-negative), $N\times N$ weight matrix W, with, for $i\neq j$, $W_{ij}>0$ if and only if, $\{i,j\}\in E$, and $W_{ii}=1-\sum_{j\neq i}W_{ij}$. Denote by $\widetilde{W}=W-J$, where $J:=\frac{1}{N}\mathbf{1}\mathbf{1}^\top$ is the ideal consensus matrix. We let $\widetilde{W}=Q\widetilde{\Lambda}Q^\top$, where $\widetilde{\Lambda}$ is the diagonal matrix with $\widetilde{\Lambda}_{ii}=\lambda_i(\widetilde{W})$, and $Q=[q_1,\ldots,q_N]$ is the matrix of the eigenvectors of $\widetilde{W}$. With D–NC, we impose Assumption1(a) below; with D–NG, we require both Assumptions 1(a) and (b). Recall $\mu(W)$—the second largest singular value of W

Assumption 1 (Weight Matrix): We assume that (a) $\mu(W)<1$; and (b) $W\succeq \eta I$, where $\eta<1$ is an arbitrarily small positive quantity.

Note that Assumption 1 (a) can be fulfilled only by a connected network. Assumption 1 (a) is standard and is also needed with the existing algorithms in [8], [14]. For a connected network, nodes can assign the weights W and fulfill Assumption 1 (a), e.g., through the Metropolis weights [28]; to set the Metropolis weights, each node needs to know its own degree and its neighbors' degrees. Assumption 1 (b) required by D–NG is not common in the literature. We discuss the impact of Assumption 1 (b) in Section VII-A.

3) Distributed Optimization Model: The nodes solve the unconstrained problem

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

The function $f_i:\mathbb{R}^d\to\mathbb{R}$ is known only to node i. We impose Assumptions 2 and 3.

Assumption 2 (Solvability; Lipschitz Continuous Gradient):

a) There exists a solution $x^\star\in\mathbb{R}^d$ with $f(x^\star)=\inf_{x\in\mathbb{R}^d}f(x)=:f^\star$.
b) $\forall i$, f_i is convex, differentiable, with Lipschitz continuous derivative with constant $L\in[0,\infty)$: $\|\nabla f_i(x)-\nabla f_i(y)\|\leq L\|x-y\|$, $\forall x,y\in\mathbb{R}^d$.

Assumption 3 (Bounded Gradients): $\exists G\in[0,\infty)$ such that, $\forall i$, $\|\nabla f_i(x)\|\leq G$, $\forall x\in\mathbb{R}^d$.

Examples of f_i's that satisfy Assumptions 2–3 include the logistic and Huber losses (See Section VIII), or the "fair" loss in robust statistics, $\phi:\mathbb{R}\mapsto\mathbb{R}$, $\phi(x)=b_0^2\left(\frac{\|x\|}{b_0}-\log\left(1+\frac{|x|}{b_0}\right)\right)$, where b_0 is a positive parameter, e.g., [29]. Assumption 2 is precisely the assumption required by [17] in the convergence analysis of the (centralized) Nesterov gradient method. With respect to the centralized Nesterov gradient method [17], we additionally require bounded gradients as given by Assumption 3. We explain the need for Assumption 3 in Section VII-B.

III. Distributed Nesterov Based Algorithms

We now consider our two proposed algorithms. Section III-A presents algorithm D–NG, while Section III-B presents algorithm D–NC.

A. Distributed Nesterov Gradient Algorithm (D–NG)

Algorithm D–NG generates the sequence $(x_i(k),y_i(k))$, $k=0,1,2,\ldots$, at each node i, where $y_i(k)$ is an auxiliary variable. D–NG is initialized by $x_i(0)=y_i(0)\in\mathbb{R}^d$, for all i. The update at node i and $k=1,2,\ldots$ is

$$x_i(k)=\sum_{j\in O_i}W_{ij}\,y_j(k-1)-\alpha_{k-1}\nabla f_i(y_i(k-1)) \tag{2}$$

$$y_i(k)=x_i(k)+\beta_{k-1}\left(x_i(k)-x_i(k-1)\right). \tag{3}$$

Here, W_{ij} are the averaging weights (the entries of W), and O_i is the neighborhood set of node i (including i). The step-size α_k and the sequence β_k are:

$$\alpha_k=\frac{c}{k+1},\quad c>0;\qquad \beta_k=\frac{k}{k+3},\quad k=0,1,\ldots. \tag{4}$$

With algorithm (2)–(3), each node i, at each iteration k, performs the following: 1) broadcasts its variable $y_i(k-1)$ to all its neighbors $j\in O_i$; 2) receives $y_j(k-1)$ from all its neighbors $j\in O_i$; 3) updates $x_i(k)$ by weight-averaging its own $y_i(k-1)$ and its neighbors variables $y_j(k-1)$, and performs a negative gradient step with respect to f_i; and 4) updates $y_i(k)$ via the inexpensive update in (3). To avoid notation explosion in the analysis further ahead, we assume throughout the paper, with both D–NG and D–NC, equal initial estimates $x_i(0)=y_i(0)=x_j(0)=y_j(0)$ for all i,j; e.g., nodes can set them to zero.

We adopt the sequence β_k as in the centralized fast gradient method by Nesterov [17]; see also [30], [31]. With the centralized Nesterov gradient, $\alpha_k=\alpha$ is constant along the iterations. However, under a constant step-size, algorithm (2)–(3) does not converge to the exact solution, but only to a solution neighborhood. More precisely, in general, $f(x_i(k))$ does not converge to $f^\star$ (See [32] for details.) We force $f(x_i(k))$ to converge to $f^\star$ with (2)–(3) by adopting a diminishing step-size α_k, as in (4). The constant $c>0$ in (4) can be arbitrary (See also ahead Theorem 5.)

1) Vector Form: Let $x(k)=(x_1(k)^\top,x_2(k)^\top,\ldots,x_N(k)^\top)^\top$, $y(k)=(y_1(k)^\top,\ y_2(k)^\top,\ldots,y_N(k)^\top)^\top$, and introduce $F:\mathbb{R}^{Nd}\to\mathbb{R}^N$ as: $F(x)=F(x_1,x_2,\ldots,x_N)=f_1(x_1)+\ldots+f_N(x_N)$. Then, given initialization $x(0)=y(0)$, D–NG in vector form is

$$x(k)\ =\ (W\otimes I)y(k-1)-\alpha_{k-1}\nabla F(y(k-1)) \tag{5}$$

$$y(k)=x(k)+\beta_{k-1}\left(x(k)-x(k-1)\right),\ k=1,2,\ldots \tag{6}$$

where the identity matrix is of size d – the dimension of the optimization variable in (1).

Algorithm D–NC

Algorithm D–NC uses a *constant step-size* $\alpha\leq 1/(2L)$ and 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 given in (7) and (13) below, respectively. D–NC is Summarized in Algorithm 1.

Algorithm 1 Algorithm D–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: (First consensus) Nodes run average consensus initialized by $x_i^{(c)}(s=0,k) = x_i^{(a)}(k)$:

$$x_i^{(c)}(s,k) = \sum_{j \in O_i} W_{ij} x_j^{(c)}(s-1,k), \;\; s = 1,2,\ldots,\tau_x(k)$$
$$\tau_x(k) = \left\lceil \frac{2\log k}{-\log \mu(W)} \right\rceil \quad (7)$$

and set $x_i(k) := x_i^{(c)}(s = \tau_x(k), k)$.
4: Node i calculates $y_i^{(a)}(k) = x_i(k) + \beta_{k-1}\left(x_i(k) - x_i(k-1)\right)$.
5: (Second consensus) Nodes run average consensus initialized by $y_i^{(c)}(s=0,k) = y_i^{(a)}(k)$:

$$y_i^{(c)}(s,k) = \sum_{j \in O_i} W_{ij} y_j^{(c)}(s-1,k), \;\; s = 1,2,\ldots,\tau_y(k)$$
$$\tau_y(k) = \left\lceil \frac{\log 3}{-\log \mu(W)} + \frac{2\log k}{-\log \mu(W)} \right\rceil \quad (8)$$

and set $y_i(k) := y_i^{(c)}(s = \tau_y(k), k)$.
6: Set $k \mapsto k+1$ and go to step 2.

The number of inner consensus iterations in (7) increases as $\log k$ and depends on the underlying network through $\mu(W)$. Note an important difference between D–NC and D–NG. D–NC uses explicitly a number of consensus steps at each k. In contrast, D–NG does not explicitly use multi-step consensus at each k; consensus occurs implicitly, similarly to [8], [14].

2) Vector Form: Using the same compact notation for $x(k)$, $y(k)$, and $\nabla F(y(k))$ as with D–NG, D–NC in vector form is

$$x(k) = (W \otimes I)^{\tau_x(k)} \left[\, y(k-1) - \alpha\, \nabla F(y(k-1)) \,\right] \quad (9)$$
$$y(k) = (W \otimes I)^{\tau_y(k)} \left[x(k) + \beta_{k-1}(x(k) - x(k-1))\right]. \quad (10)$$

The power $(W \otimes I)^{\tau_x(k)}$ in (9) corresponds to the first consensus in (7), and the power $(W \otimes I)^{\tau_y(k)}$ in (10) corresponds to the second consensus in (8). The connection between D–NC and the (centralized) Nesterov gradient method becomes clearer in Section IV-B. The matrix powers (9)–(10) are implemented in a distributed way through multiple iterative steps – they require respectively $\tau_x(k)$ and $\tau_y(k)$ iterative (distributed) consensus steps. This is clear from the representation in Algorithm 1.

IV. Intermediate Results: Inexact Nesterov Gradient Method

We will analyze the convergence rates of D–NG and D–NC by considering the evolution of the global averages $\overline{x}(k) := \frac{1}{N}\sum_{i=1}^N x_i(k)$ and $\overline{y}(k) := \frac{1}{N}\sum_{i=1}^N y_i(k)$. We will show that, with both distributed methods, the evolution of $\overline{x}(k)$ and $\overline{y}(k)$ can be studied through the framework of the inexact (centralized) Nesterov gradient method, essentially like the one in [33]. Section IV-A introduces this framework and gives the relation for the progress in one iteration. Section IV-B then demonstrates that we can cast our algorithms D–NG and D–NC in this framework.

A. Inexact Nesterov Gradient Method

We next introduce the definition of a (pointwise) inexact first order oracle.

Definition 1 (Pointwise Inexact First Order Oracle): Consider a function $f : \mathbb{R}^d \to \mathbb{R}$ that is convex and has Lipschitz continuous gradient with constant L_f. We say that a pair $\left(\widehat{f}_y, \widehat{g}_y\right) \in \mathbb{R} \times \mathbb{R}^d$ is a (L_y, δ_y) inexact oracle of f at point y if:

$$\widehat{f}_y + \widehat{g}_y^\top (x-y) \le f(x) \le \widehat{f}_y + \widehat{g}_y^\top (x-y) + \frac{L_y}{2}\|x-y\|^2 + \delta_y, \; \forall x \in \mathbb{R}^d. \quad (11)$$

For any $y \in \mathbb{R}^d$, the pair $(f(y), \nabla f(y))$ satisfies Definition 1 with $(L_y = L_f, \delta_y = 0)$. If $\left(\widehat{f}_y, \widehat{g}_y\right)$ is a (L_y, δ_y) inexact oracle at y, then it is also a $\left(L'_y, \delta_y\right)$ inexact oracle at y, with $L'_y \ge L_y$.

Remark: The prefix pointwise in Definition 1 emphasizes that we are concerned with finding $\left(\widehat{f}_y, \widehat{g}_y\right)$ that satisfy (11) with (L_y, δ_y) *at a fixed point* y. This differs from the conventional definition (Definition 1) in [33]. Throughout, we always refer to the inexact oracle in the sense of Definition 1 here and drop the prefix pointwise.

1) Inexact Nesterov Gradient Method: Lemma 2 gives the progress in one iteration of the inexact (centralized) Nesterov gradient method for the unconstrained minimization of f. Consider a point $(\overline{x}(k-1), \overline{y}(k-1)) \in \mathbb{R}^d \times \mathbb{R}^d$, for some fixed $k = 1,2,\ldots$ Let $\left(\widehat{f}_{k-1}, \widehat{g}_{k-1}\right)$ be a (L_{k-1}, δ_{k-1}) inexact oracle of the function f at point $\overline{y}(k-1)$ and

$$\overline{x}(k) = \overline{y}(k-1) - \frac{1}{L_{k-1}}\widehat{g}_{k-1} \quad (12)$$
$$\overline{y}(k) = \overline{x}(k) + \beta_{k-1}\left(\overline{x}(k) - \overline{x}(k-1)\right).$$

Lemma 2 (Progress per Iteration): Consider the update rule (12) for some $k = 1,2,\ldots$ Then

$$(k+1)^2\left(f(\overline{x}(k)) - f(x^\bullet)\right) + 2L_{k-1}\|\overline{v}(k) - x^\bullet\|^2 \le (k^2-1)\left(f(\overline{x}(k-1)) - f(x^\bullet)\right) + 2L_{k-1}\|\overline{v}(k-1) - x^\bullet\|^2 + (k+1)^2\delta_{k-1} \quad (13)$$

for any $x^\bullet \in \mathbb{R}^d$, where $\gamma_k = 2/(k+2)$ and $\overline{v}(k) = \frac{\overline{y}(k) - (1-\gamma_k)\overline{x}(k)}{\gamma_k}$.

Lemma 2 is similar to [[33], Theorem 5], although [33] considers a different accelerated Nesterov method. It is intuitive: the progress per iteration is the same as with the exact Nesterov gradient algorithm, except that it is deteriorated by the "gradient direction inexactness" ($(k+1)^2\delta_{k-1}$). The proof follows the arguments of [33] and [17], [30], [31] and is in [18].

Algorithms D–NG and D–NC in the Inexact Oracle Framework

We now cast algorithms D–NG and D–NC in the inexact oracle framework.

Algorithm D–NG: Recall the global averages $\overline{x}(k) := \frac{1}{N}\sum_{i=1}^N x_i(k)$ and $\overline{y}(k) := \frac{1}{N}\sum_{i=1}^N y_i(k)$, and define

$$\widehat{f}_k = \sum_{i=1}^N \left\{ f_i(y_i(k)) + \nabla f_i(y_i(k))^\top (\overline{y}(k) - y_i(k)) \right\} \quad (14)$$
$$\widehat{g}_k = \sum_{i=1}^N \nabla f_i(y_i(k)).$$

Multiplying (5)–(6) from the left by $(1/N)(\mathbf{1}^\top \otimes I)$, using $(\mathbf{1}^\top \otimes I)(W \otimes I) = \mathbf{1}^\top \otimes I$, letting $L'_{k-1} := \frac{N}{\alpha_{k-1}}$, and using $\widehat{g}_k$ in (14), we obtain that $\overline{x}(k)$, $\overline{y}(k)$ evolve according to

$$\overline{x}(k) = \overline{y}(k-1) - \frac{1}{L'_{k-1}} \widehat{g}_{k-1} \quad (15)$$
$$\overline{y}(k) = \overline{x}(k) + \beta_{k-1}\left(\overline{x}(k) - \overline{x}(k-1)\right).$$

The following Lemma shows how we can analyze convergence of (15) in the inexact oracle framework. Define $\widetilde{y}_i(k) := y_i(k) - \overline{y}(k)$ and $\widetilde{y}(k) := (\widetilde{y}_1(k)^\top, \ldots, \widetilde{y}_N(k)^\top)^\top$. Define analogously $\widetilde{x}_i(k)$ and $\widetilde{x}(k)$. We refer to $\widetilde{x}(k)$ and $\widetilde{y}(k)$ as the disagreement vectors, as they indicate how mutually apart the estimates of different nodes are.

Lemma 3: Let Assumption 2 hold. Then, $(\widehat{f}_k, \widehat{g}_k)$ in (14) is a (L_k, δ_k) inexact oracle of $f = \sum_{i=1}^N f_i$ at point $\overline{y}(k)$ with constants $L_k = 2NL$ and $\delta_k = L\|\widetilde{y}(k)\|^2$.

Lemma 3 implies that, if $L'_{k-1} = \frac{Nk}{c} \geq 2NL$, i.e., if $c \leq \frac{k}{2L}$, then the progress per iteration in Lemma 2 holds for (15) with $\delta_{k-1} := L\|\widetilde{y}(k-1)\|^2$. If $c \leq 1/(2L)$, Lemma 2 applies for *all iterations* $k = 1, 2, \ldots$; otherwise, it holds for all $k \geq 2cL$.

Proof of Lemma 3: For notation simplicity, we re-write $y(k)$ and $\overline{y}(k)$ as y and $\overline{y}$, and $\widehat{f}_k, \widehat{g}_k, L_k, \delta_k$ as $\widehat{f}_y, \widehat{g}_y, L_y, \delta_y$. In view of Definition 1, we need to show inequalities (11). We first show the left one. By convexity of $f_i(\cdot)$: $f_i(x) \geq f_i(y_i) + \nabla f_i(y_i)^\top (x - y_i), \quad \forall x$; summing over $i = 1, \ldots, N$, using $f(x) = \sum_{i=1}^N f_i(x)$, and expressing $x - y_i = x - \overline{y} + \overline{y} - y_i$

$$f(x) \geq \sum_{i=1}^N \left(f_i(y_i) + \nabla f_i(y_i)^\top (\overline{y} - y_i)\right) + \left(\sum_{i=1}^N \nabla f_i(y_i)\right)^\top (x - \overline{y}) = \widehat{f}_y + \widehat{g}_y^\top (x - \overline{y}).$$

We now prove the right inequality in (11). As $f_i(\cdot)$ is convex and has Lipschitz continuous derivative with constant L, we have: $f_i(x) \leq f_i(y_i) + \nabla f_i(y_i)^\top (x - y_i) + \frac{L}{2}\|x - y_i\|^2$, which, after summation over $i = 1, \ldots, N$, expressing $x - y_i = (x - \overline{y}) + (\overline{y} - y_i)$, and using the inequality $\|x - y_i\|^2 = \|(x - \overline{y}) + (\overline{y} - y_i)\|^2 \leq 2\|x - \overline{y}\|^2 + 2\|\overline{y} - y_i\|^2$, gives

$$f(x) \leq \sum_{i=1}^N \left(f_i(y_i) + \nabla f_i(y_i)^\top (\overline{y} - y_i)\right) + \left(\sum_{i=1}^N \nabla f_i(y_i)\right)^\top (x - \overline{y}) + NL\|x - \overline{y}\|^2 + L\sum_{i=1}^N \|\overline{y} - y_i\|^2 = \widehat{f}_y + \widehat{g}_y^\top (x - \overline{y}) + \frac{2NL}{2}\|x - \overline{y}\|^2 + \delta_y$$

and so $(\widehat{f}_y, \widehat{g}_y)$ satisfy the right inequality in (11) with $L_y = 2NL$ and $\delta_y = L\sum_{i=1}^N \|\overline{y} - y_i\|^2$. ■

Algorithm D–NC: Consider algorithm D–NC in (9)–(10). To avoid notational clutter, use the same notation as with D–NG for the global averages: $\overline{x}(k) := \frac{1}{N}\sum_{i=1}^N x_i(k)$, and $\overline{y}(k) := \frac{1}{N}\sum_{i=1}^N y_i(k)$, re-define $\widehat{f}_k, \widehat{g}_k$ for D–NC as in (14), and let $L'_{k-1} := \frac{N}{\alpha}$. Multiplying (9)–(10) from the left by $(1/N)\mathbf{1}^\top \otimes I$, and using $(\mathbf{1}^\top \otimes I)(W \otimes I) = \mathbf{1}^\top \otimes I$, we get that $\overline{x}(k), \overline{y}(k)$ satisfy (15). As $\alpha \leq 1/(2L)$, we have $L'_{k-1} \geq 2NL$, and so, by Lemma 3, the progress per iteration in Lemma 2 applies to $\overline{x}(k), \overline{y}(k)$ of D–NC for all k, with $\delta_{k-1} = L\|\widetilde{y}(k-1)\|^2$.

In summary, the analysis of convergence rates of both D–NG and D–NC boils down to finding the disagreements $\|\widetilde{y}(k)\|$ and then applying Lemma 2.

V. Algorithm D–NG: Convergence Analysis

This section studies the convergence of D–NG. Section V-A bounds the disagreements $\|\widetilde{x}(k)\|$ and $\|\widetilde{y}(k)\|$ with D–NG; Section V-B combines these bounds with Lemma 2 to derive the convergence rate of D–NG and its dependence on the underlying network.

A. Algorithm D–NG: Disagreement Estimate

This subsection shows that $\|\widetilde{x}(k)\|$ and $\|\widetilde{y}(k)\|$ are $O(1/k)$, hence establishing asymptotic consensus – the differences of the nodes' estimates $x_i(k)$ (and $y_i(k)$) converge to zero. Recall the step-size constant $c > 0$ in (4) and the gradient bound G in Assumption 3.

Theorem 4 (Consensus With D–NG): For D–NG in (2)–(4) under Assumptions 1 and 3:

$$\|\widetilde{x}(k)\| \leq \sqrt{N}\, c\, G\, C_{\text{cons}}\, \frac{1}{k} \quad (16)$$
$$\|\widetilde{y}(k)\| \leq 4\sqrt{N}\, c\, G\, C_{\text{cons}}\, \frac{1}{k},\ k = 1, 2, \ldots,$$
$$C_{\text{cons}} = \frac{8\left\{2\mathcal{B}\left(\sqrt{\mu(W)}\right) + \frac{7}{1-\mu(W)}\right\}}{\sqrt{\eta(1-\mu(W))}} \quad (17)$$

with $\mathcal{B}(r) := \sup_{z \geq 1/2}\left(z r^z \log(1+z)\right) \in (0, \infty),\ r \in (0,1)$.

For notational simplicity, we prove Theorem 4 for $d = 1$, but the proof extends to a generic $d > 1$. We model the dynamics of the augmented state $(\widetilde{x}(k)^\top, \widetilde{x}(k-1)^\top)^\top$ as a linear time varying system with inputs $(I - J)\nabla F(y(k))$. We present here the linear system and solve it in the Appendix. Substitute the expression for $y(k-1)$ in (5); multiply the resulting equation from the left by $(I - J)$; use $(I - J)W = \widetilde{W} = \widetilde{W}(I - J)$; and set $\widetilde{x}(0) = 0$ by assumption. We obtain

$$\begin{bmatrix} \widetilde{x}(k) \\ \widetilde{x}(k-1) \end{bmatrix} = \begin{bmatrix} (1+\beta_{k-2})\widetilde{W} & -\beta_{k-2}\widetilde{W} \\ I & 0 \end{bmatrix} \begin{bmatrix} \widetilde{x}(k-1) \\ \widetilde{x}(k-2) \end{bmatrix} - \alpha_{k-1} \begin{bmatrix} (I-J)\nabla F(y(k-1)) \\ 0 \end{bmatrix} \quad (18)$$

for all $k = 1, 2, \ldots$, where β_k, for $k = 0, 1, \ldots$, is in (4), $\beta_{-1} = 0$, and $(\widetilde{x}(0)^\top, \widetilde{x}(-1)^\top)^\top = 0$. We emphasize that system (18) is more complex than the corresponding systems in, e.g., [8], [14], which involve only a single state $\widetilde{x}(k)$; the upper bound on $\|\widetilde{x}(k)\|$ from (18) is an important technical contribution of this paper; see Theorem 4 and Appendix A.

B. Convergence Rate and Network Scaling

Theorem 5 (a) states the $O(\log k/k)$ convergence rate result for D–NG when the step-size constant $c \leq 1/(2L)$; Theorem 5(b) (proved in [18]) demonstrates that the $O(\log k/k)$ convergence rate still holds if $c > 1/(2L)$, with a deterioration in the convergence constant. Part (b) assumes $x_i(0) = y_i(0) = 0, \forall i$, to avoid notational clutter.

Theorem 5: Consider D–NG under Assumptions 1–3. Let $\|\overline{x}(0) - x^\star\| \leq R$, $R \geq 0$. Then:

(a) If $c \leq 1/(2L)$, we have, $\forall i$, $\forall k = 1, 2, \ldots$

$$\frac{f(x_i(k)) - f^\star}{N} \leq \frac{2\,R^2}{c}\left(\frac{1}{k}\right) + 16\,c^2\;L\,C_{\rm cons}^2\,G^2 \times \left(\frac{1}{k}\sum_{t=1}^{k-1}\frac{(t+2)^2}{(t+1)t^2}\right) + c\sqrt{N}\,G^2 C_{\rm cons}\left(\frac{1}{k}\right) \leq \mathcal{C}\left(\frac{1}{k}\sum_{t=1}^{k}\frac{(t+2)^2}{(t+1)t^2}\right)$$

$$\mathcal{C} = \frac{2R^2}{c} + 16c^2 L C_{\rm cons}^2 G^2 + c\sqrt{N}\,G^2 C_{\rm cons}. \tag{19}$$

(b) Let $x_i(0) = y_i(0) = 0, \forall i$. If $c > 1/(2L)$, (19) holds $\forall i$, $\forall k \geq 2\,cL$, with $\mathcal{C}$ replaced with $\mathcal{C}' = \mathcal{C}''(L, G, R, c) + 16c^2 L C_{\rm cons}^2 G^2 + c\sqrt{N}\,G^2 C_{\rm cons}$, and $\mathcal{C}''(L, G, R, c) \in [0, \infty)$ is a constant that depends on L, G, R, c, and *is independent of* N and W.

We prove here Theorem 5 (a); for part (b), see [18].

Proof of Theorem 5 (a): The proof consists of two parts. In the Step 1 of the proof, we estimate the optimality gap $\frac{1}{N}(f(\overline{x}(k)) - f^\star)$ at the point $\overline{x}(k) = \frac{1}{N}\sum_{i=1}^N x_i(k)$ using Lemma 2 and the inexact oracle machinery. In the Step 2, we estimate the optimality gap $\frac{1}{N}(f(x_i(k)) - f^\star)$ at any node i using convexity of the f_i's and the bound on $\|\widetilde{x}(k)\|$ from Theorem 4.

Step 1. Optimality Gap $(f(\overline{x}(k)) - f^\star)$: Recall that, for $k = 1, 2, \ldots$, $(\widehat{f}_k, \widehat{g}_k)$ in (14) is a (L_k, δ_k) inexact oracle of f at point $\overline{y}(k)$ with $L_k = 2NL$ and $\delta_k = L\|\widetilde{y}(k)\|^2$. Note that $(\widehat{f}_k, \widehat{g}_k)$ is also a (L'_k, δ_k) inexact oracle of f at point $\overline{y}(k)$ with $L'_k = N\frac{1}{c}(k+1) = \frac{N}{\alpha_k}$, because $\frac{1}{c} \geq 2L$, and so $L'_k \geq L_k$. Now, we apply Lemma 2 to (15), with $x^\bullet = x^\star$, and the Lipschitz constant $L'_k = 1/(\alpha_k/N)$. Recall that $\overline{v}(k) = \frac{\overline{y}(k) - (1-\gamma_k)\overline{x}(k)}{\gamma_k}$. We get

$$\frac{(k+1)^2}{k}\left(f(\overline{x}(k)) - f^\star\right) + \frac{2N}{c}\|\overline{v}(k) - x^\star\|^2 \leq \frac{k^2-1}{k}\left(f(\overline{x}(k-1)) - f^\star\right) + \frac{2N}{c}\|\overline{v}(k-1) - x^\star\|^2 + L\|\widetilde{y}(k-1)\|^2\frac{(k+1)^2}{k}. \tag{20}$$

Because $\frac{(k+1)^2}{k} \geq \frac{(k+1)^2-1}{k+1}$, and $(f(\overline{x}(k)) - f^\star) \geq 0$, we have

$$\frac{(k+1)^2-1}{k+1}\left(f(\overline{x}(k)) - f^\star\right) + \frac{2N}{c}\|\overline{v}(k) - x^\star\|^2 \leq \frac{k^2-1}{k}\left(f(\overline{x}(k-1)) - f^\star\right) + \frac{2N}{c}\|\overline{v}(k-1) - x^\star\|^2 + L\|\widetilde{y}(k-1)\|^2\frac{(k+1)^2}{k}.$$

By unwinding the above recursion, and using $\overline{v}(0) = \overline{x}(0)$, gives: $\frac{(k+1)^2-1}{k+1}\left(f(\overline{x}(k)) - f^\star\right) \leq \frac{2N}{c}\|\overline{x}(0) - x^\star\|^2 + L\sum_{t=1}^k \|\widetilde{y}(t-1)\|^2\frac{(t+1)^2}{t}$. Applying Theorem 4 to the last equation, and using $\frac{k+1}{(k+1)^2-1} = \frac{k+1}{k(k+2)} \leq \frac{k+2}{k(k+2)} = \frac{1}{k}$, and the assumption $\|\widetilde{y}(0)\| = 0$, leads to, as desired

$$\left(f(\overline{x}(k)) - f^\star\right) \leq \frac{1}{k}\frac{2N}{c}\|\overline{x}(0) - x^\star\|^2 + \frac{16\,c^2\,N}{k}L\,C_{\rm cons}^2 G^2\sum_{t=2}^k\frac{(t+1)^2}{t(t-1)^2}. \tag{21}$$

Step 2. Optimality Gap $(f(x_i(k)) - f^\star)$: Fix an arbitrary node i; then, by convexity of f_j, $j = 1, 2, \ldots, N$: $f_j(\overline{x}(k)) \geq f_j(x_i(k)) + \nabla f_j(x_i(k))^\top(\overline{x}(k) - x_i(k))$, and so: $f_j(x_i(k)) \leq f_j(\overline{x}(k)) + G\|\overline{x}(k) - x_i(k)\|$. Summing the inequalities for $j = 1, \ldots, N$, using $\|\overline{x}(k) - x_i(k)\| \leq \|\widetilde{x}(k)\|$, subtracting $f^\star$ from both sides, from Theorem 4

$$f(x_i(k)) - f^\star \leq f(\overline{x}(k)) - f^\star + GN\|\widetilde{x}(k)\| \leq f(\overline{x}(k)) - f^\star + c\,N\sqrt{N}\,C_{\rm cons}G^2\frac{1}{k} \tag{22}$$

which, with (21) where the summation variable t is replaced by $t+1$, completes the proof. ■

1) Network Scaling: Using Theorem 5, Theorem 6 studies the dependence of the convergence rate on the underlying network – N and W, when: 1) nodes do not know L and $\mu(W)$ before the algorithm run, and they set the step-size constant c to a constant independent of N, L, W, e.g., $c = 1$; and 2) nodes know $L, \mu(W)$, and they set $c = \frac{1-\mu(W)}{2L}$. See [14] for dependence of $1/(1-\mu(W))$ on N for commonly used models, e.g., expanders or geometric graphs.

Theorem 6: Consider the algorithm D–NG in (2)–(4) under Assumptions 1–3. Then, $\frac{1}{N}\left(f(x_i(k)) - f^\star\right)$ is

$$O\left(\frac{1}{(1-\mu)^{p+\xi}}\left[\frac{\log k}{k} + \frac{N^{1/2}\log^{1/2} k}{k^{3/2}} + \frac{N}{k^2}\right]\right)$$

where: (a) $p = 3$ for arbitrary $c = {\rm const} > 0$; and (b) $p = 1$ for $c = \frac{1-\mu(W)}{2L}$.

Proof of Theorem 6: Fix $\eta \in (0,1)$ and $\xi \in (0,1)$ (two arbitrarily small positive constants). By Assumption 1 (b), $\mu = \mu(W) \in [\eta, 1]$. We show that for $C_{\rm cons}$ in (17)

$$C_{\rm cons} \leq A(\xi, \eta)\;\frac{1}{(1-\mu)^{3/2+\xi}},\;\;\forall \mu \in [\eta, 1] \tag{23}$$

where $A(\xi, \eta) \in (0, \infty)$ depends only on ξ, η. Consider $\mathcal{B}(r) = \sup_{z \geq 1/2}\left\{z\,r^z\log(1+z)\right\}$, $r \in (0,1)$; there exists $K_B(\xi) \in (0, \infty)$ such that: $\log(1+z) \leq K_B(\xi)z^\xi, \forall z \geq 1/2$. Thus

$$\mathcal{B}(r) \leq K_B(\xi)\sup_{z \geq 1/2}\left\{z^{1+\xi}r^z\right\} = \frac{K_B(\xi)\,e^{-(1+\xi)}(1+\xi)^{(1+\xi)}}{(-\log r)^{1+\xi}} =: \frac{A'(\xi)}{(-\log r)^{1+\xi}}$$

for all $r \in (0,1)$. From the above equation, and using $1/(-\log\sqrt{u}) \leq 2/(1-u)$, $\forall u \in [0,1)$, we have $\mathcal{B}\left(\sqrt{\mu}\right) \leq 2A'(\xi)/(1-\mu)^{1+\xi}$. The latter, applied to (17), yields (23), with $A(\xi, \eta) := \frac{8}{\sqrt{\eta}}\max\{3A'(\xi), 7\}$.

A scaling result $O\left(\frac{N^{1/2}}{(1-\mu)^{p+\xi}}\frac{\log k}{k}\right)$, $p=1,3$, readily follows by substitution of (23) in Theorem 5 (a) and (b), respectively. To prove Theorem 6, we modify the argument of (22). We first prove claim (b). Namely, at any node i, using Lipschitz continuity of ∇f (with constant NL), $f(x_i(k)) \leq f(\overline{x}(k)) + \nabla f(\overline{x}(k))^\top (x_i(k)-\overline{x}(k)) + \frac{NL}{2}\|x_i(k)-\overline{x}(k)\|^2$, and thus

$$f(x_i(k)) \leq f(\overline{x}(k)) + \|\nabla f(\overline{x}(k))\|\|\widetilde{x}(k)\| + \frac{NL\|\widetilde{x}(k)\|^2}{2} \tag{24}$$

where we use $\|x_i(k)-\overline{x}(k)\| \leq \|\widetilde{x}(k)\|$. From (21), $f(\overline{x}(k)) - f^\star = O\left(\frac{N}{ck} + \frac{Nc^2C_{\text{cons}}^2\log k}{k}\right)$. Using again Lipschitz continuity of ∇f (with constant NL)

$$\begin{aligned}\|\nabla f(\overline{x}(k))\| &\leq \sqrt{2NL}\sqrt{f(\overline{x}(k))-f^\star} \\ &= O\left(\frac{N}{\sqrt{ck}} + \frac{NcC_{\text{cons}}\log^{1/2} k}{\sqrt{k}}\right).\end{aligned}$$

Consider (24). Subtracting $f^\star$ from both sides, dividing by N, and substituting the above bound on $\|\nabla f(\overline{x}(k))\|$ while using Theorem 5 (a), we obtain

$$\begin{aligned}\frac{f(x_i(k))-f^\star}{N} &= O\left(\frac{1}{c\,k} + \frac{c^2C_{\text{cons}}^2\log k}{k}\right. \\ &\left. + \left(\frac{1}{\sqrt{c\,k}} + \frac{cC_{\text{cons}}\log^{1/2} k}{\sqrt{k}}\right)\frac{\sqrt{N}cC_{\text{cons}}}{k} + \frac{Nc^2C_{\text{cons}}^2}{k^2}\right).\end{aligned} \tag{25}$$

We now apply (23) to (25). Claim (b) is proved after setting $c=(1-\mu)/2L$. The proof for claim (a) is completely analogous; the argument only replaces the term $\frac{1}{k}\frac{2N}{c}R^2$ in (21) with $\frac{1}{k}\mathcal{C}''(L,G,R,C)$, see also [18], and sets $c=\Theta(1)$. ■

VI. Algorithm D–NC: Convergence Analysis

We now consider the D–NC algorithm. Section VI-A provides the disagreement estimate, while Section VI-A gives the convergence rate and network scaling.

A. Disagreement Estimate

We estimate the disagreements $\widetilde{x}(k)$, and $\widetilde{y}(k)$ with D–NC.

Theorem 7 (Consensus With D–NC): Let Assumptions 1 (a) and 3 hold, and consider the algorithm D–NC. Then, for $k=1,2,\ldots$: $\|\widetilde{x}(k)\| \leq 2\alpha\sqrt{N}G\frac{1}{k^2}$, and $\|\widetilde{y}(k)\| \leq 2\alpha\sqrt{N}G\frac{1}{k^2}$.

Proof: For notational simplicity, we perform the proof for $d=1$, but it extends to a generic $d>1$. Denote by $B_{t-1} := \max\{\|\widetilde{x}(t-1)\|,\ \|\widetilde{y}(t-1)\|\}$, and fix $t-1$. We want to upper bound B_t. Multiplying (9)–(10) by $(I-J)$ from the left, using $(I-J)W=\widetilde{W}(I-J)$

$$\widetilde{x}(t) = \widetilde{W}^{\tau_x(t)}\,\widetilde{y}(t-1) - \alpha\widetilde{W}^{\tau_x(t)}(I-J)\nabla F(y(t-1)) \tag{26}$$

$$\widetilde{y}(t) = \widetilde{W}^{\tau_y(t)}\left[\,\widetilde{x}(t) + \beta_{t-1}(\widetilde{x}(t)-\widetilde{x}(t-1))\,\right]. \tag{27}$$

We upper bound $\|\widetilde{x}(t)\|$ and $\|\widetilde{y}(t)\|$ from (26), (27). Recall $\|\widetilde{W}\| = \mu(W) := \mu \in (0,1)$; from (7) and (13), we have $\mu^{\tau_x(t)} \leq \frac{1}{t^2}$ and $\mu^{\tau_y(t)} \leq \frac{1}{3t^2}$. From (26), using the sub-additive and sub-multiplicative properties of norms, and using $\|\widetilde{y}(t-1)\| \leq B_{t-1}$, $\mu\in(0,1)$, $\|(I-J)\nabla F(y(t-1))\| \leq \|\nabla F(y(t-1))\| \leq \sqrt{N}G$, $\beta_{t-1}\leq 1$

$$\begin{aligned}\|\widetilde{x}(t)\| &\leq \mu^{\tau_x(t)}\,B_{t-1} + \alpha\mu^{\tau_x(t)}\sqrt{N}G \\ &\leq \frac{1}{t^2}\,B_{t-1} + \alpha\sqrt{N}G\frac{1}{t^2}\end{aligned} \tag{28}$$

$$\begin{aligned}\|\widetilde{y}(t)\| &\leq 2\,\mu^{\tau_y(t)}\|\widetilde{x}(t)\| + \mu^{\tau_y(t)}\|\widetilde{x}(t-1)\| \\ &\leq 2\,\mu^{\tau_x(t)+\tau_y(t)}B_{t-1} + 2\alpha\sqrt{N}G\,\mu^{\tau_x(t)+\tau_y(t)} \\ &\quad + \mu^{\tau_y(t)}B_{t-1} \\ &\leq 3\,\mu^{\tau_y(t)}B_{t-1} + 2\alpha\sqrt{N}G\mu^{\tau_y(t)} \\ &\leq \frac{1}{t^2}B_{t-1} + \alpha\sqrt{N}G\frac{1}{t^2}.\end{aligned} \tag{29}$$

Clearly, from (28) and (29): $B_t \leq \frac{1}{t^2}B_{t-1} + \frac{1}{t^2}\alpha\sqrt{N}G$. Next, using $B_0=0$, unwind the latter recursion for $k=1,2$, to obtain, respectively: $B_1 \leq \alpha\sqrt{N}G$ and $B_2 \leq \alpha\sqrt{N}G/2$, and so the bound in Theorem 7 holds for $k=1,2$. Further, for $k\geq 3$ unwinding the same recursion for $t=k,k-1,\ldots,1$

$$\begin{aligned}B_k &\leq \frac{\alpha\sqrt{N}G}{k^2}\left(1 + \sum_{t=2}^{k-1}\frac{1}{(k-1)^2(k-2)^2\ldots t^2}\right. \\ &\quad \left. + \frac{1}{(k-1)^2(k-2)^2\ldots 2^2}\right) \\ &\leq \frac{\alpha\sqrt{N}G}{k^2}\left(1 + \sum_{t=2}^{k-1}\frac{1}{t^2} + \frac{1}{2^2}\right) \\ &\leq \frac{\alpha\sqrt{N}G}{k^2}\left(\frac{\pi^2}{6}+\frac{1}{4}\right) \leq \frac{2\,\alpha\sqrt{N}G}{k^2}\end{aligned}$$

where we use $1+\sum_{t=2}^{k-1}\frac{1}{t^2} \leq \pi^2/6$, $\forall k\geq 3$. ■

B. Convergence Rate and Network Scaling

We are now ready to state the Theorem on the convergence rate of D–NC.

Theorem 8: Consider the algorithm D–NC under Assumptions 1 (a), 2, and 3. Let $\|\overline{x}(0)-x^\star\| \leq R$, $R\geq 0$. Then, after

$$\begin{aligned}\mathcal{K} &= \sum_{t=1}^{k}(\tau_x(t)+\tau_y(t)) \\ &\leq \frac{2}{-\log\mu(W)}\,(k\log 3 + 2(k+1)\log(k+1)) = O\,(k\log k)\end{aligned}$$

communication rounds, i.e., after k outer iterations, at any node i

$$\begin{aligned}&\frac{1}{N}\,(f(x_i(k))-f^\star) \\ &\leq \frac{1}{k^2}\left(\frac{2}{\alpha}R^2 + 11\,\alpha^2LG^2 + \alpha\sqrt{N}G^2\right),\ \ k=1,2,\ldots.\end{aligned} \tag{30}$$

Proof Outline: The proof is very similar to the proof of Theorem 5 (a) (for details see [18], second version v2); first upper bound $f(\overline{x}(k))-f^\star$, and then $f(x_i(k))-f^\star$. To upper bound $f(\overline{x}(k))-f^\star$, recall that the evolution (15) with $\alpha_k=\alpha$ for $(\overline{x}(k),\overline{y}(k))$ is the inexact Nesterov gradient with the inexact oracle $(\widehat{f}_k,\widehat{g}_k)$ in (14), and $(L_k=2NL,\ \delta_k = L\|\widetilde{y}(k)\|^2)$. Then, apply Lemma 2 with $x^\bullet \equiv x^\star$ and $L'_{k-1}=N/\alpha$, and use Theorem 7, to obtain

$$f(\overline{x}(k))-f^\star \leq \frac{1}{k^2}\left(\frac{2NR^2}{\alpha} + 11\,\alpha^2LNG^2\right). \tag{31}$$

Finally, find the bound on $f(x_i(k)) - f^\star$ analogously to the proof of Theorem 5 (a). ■

1) Network Scaling: We now give the network scaling for algorithm D–NC in Theorem 9. We assume that nodes know L and $\mu(W)$ before the algorithm run.

Theorem 9: Consider D–NC under Assumptions 1 (a), 2, and 3 with step-size $\alpha \leq 1/(2L)$. Then, after k outer iterations and $\mathcal{K}$ communication rounds, at any node i, $\frac{1}{N}\left(f(x_i) - f^\star\right)$ is $O\left(\frac{1}{((1-\mu)\mathcal{K}^{1-\xi})^2} + \frac{\sqrt{N}}{((1-\mu)\mathcal{K}^{1-\xi})^3} + \frac{N}{((1-\mu)\mathcal{K}^{1-\xi})^4}\right)$ and $O\left(\frac{1}{k^2} + \frac{N^{1/2}}{k^3} + \frac{N}{k^4}\right)$.

Proof: Fix $\xi \in (0,1)$, and let $\mathcal{K}$ be the number of elapsed communication rounds after k outer iterations. There exists $C_0(\xi) \in (1,\infty)$, such that, $2\left(k\log 3 + 2(k+1)\log(k+1)\right) \leq C_0(\xi)k^{1+\xi}$, $\forall k \geq 1$. The latter, combined with $1/(-\log\mu(W)) \leq 1/(1-\mu(W))$, $\mu(W) \in [0,1)$, and the upper bound bound on $\mathcal{K}$ in Theorem 8, gives: $1/k \leq (C_0(\xi))\frac{1}{(1-\mu)\mathcal{K}^{1-\xi}}$. Plugging the latter in the optimality gap bound in Theorem 8 gives a scaling result $O(N^{1/2}/((1-\mu)\mathcal{K})^{2-2\xi})$ and $O(N^{1/2}/k^2)$. To prove Theorem 9, we proceed analogously to the proof of Theorem 6. From Theorem 8 and $\|\nabla f(\overline{x}(k))\| \leq \sqrt{2NL}\sqrt{f(\overline{x}(k)) - f^\star}$, $\|\nabla f(\overline{x}(k))\| = O(N/k)$. Consider (24). Subtracting $f^\star$, dividing by N, and using $\|\nabla f(\overline{x}(k))\| = O(N/k)$ and (31), we obtain $\frac{1}{N}(f(x_i(k)) - f^\star) = O(1/k^2 + N^{1/2}/k^3 + N/k^4)$. Finally, substitute $1/k \leq (C_0(\xi))\frac{1}{(1-\mu)\mathcal{K}^{1-\xi}}$ in the last bound. ■

VII. Comparisons With the Literature and Discussion of the Assumptions

Section VII-A compares D–NG, D–NC, and the distributed (sub)gradient algorithms in [8], [14], [19], from the aspects of implementation and convergence rate; Section VII-B gives a detailed discussion on Assumptions 1–3.

A. Comparisons of D–NG and D–NC With the Literature

We first set up the comparisons by explaining how to account for Assumption 1 (b) and by adapting the results in [19], [20] to our framework.

Assumption 1(b): To be fair, we account for Assumption 1(b) with D–NG as follows. Suppose that the nodes are given arbitrary symmetric, doubly stochastic weights W with $\mu(W) < 1$ – the matrix required by D–NC and [8], [14], [19]. (For example, the Metropolis weights W.) As the nodes may not be allowed to check whether the given W obeys Assumption 1 (b) or not, they modify the weights to $W' := \frac{1+\eta}{2}I + \frac{1-\eta}{2}W$, where $\eta \in (0,1)$ can be taken arbitrarily small. The matrix W' obeys Assumption 1 (b), whether W obeys it or not. The modification is done without any required knowledge of the system parameters nor inter-node communication; node i sets: 1) $W'_{ij} = \frac{1-\eta}{2}W_{ij}$, for $\{i,j\} \in E$, $i \neq j$; 2) $W'_{ij} = 0$, for $\{i,j\} \notin E$, $i \neq j$; and 3) $W'_{ii} := 1 - \sum_{j\neq i} W'_{ij}$. To be fair, when we compare D–NG with other methods (either theoretically as we do here or numerically as done in Section VIII), we set its weights to W'. For theoretical comparisons, from Theorem 5, the convergence rate of D–NG depends on W' through the inverse spectral gap $1/(1-\mu(W'))$. It can be shown that $\frac{1}{1-\mu(W')} = \frac{2}{1-\eta}\frac{1}{1-\mu(W)}$, i.e., the spectral gaps of W and W' differ only by a constant factor and the weight modification does not affect the convergence rate (up to a numerical constant); henceforth, we express the theoretical rate for D–NG in terms of W.

1) References [19], [20]: These works develop and analyze non-accelerated and accelerated distributed gradient and proximal gradient methods for time-varying networks and convex f_i's that have a differentiable component with Lipschitz continuous and bounded gradient and a non-differentiable component with bounded gradient. To compare with [20], we adapt it to our framework of static networks and differentiable f_i's. (We set the non-differentiable components of the f_i's to zero.) [19], [20] assume deterministic time-varying networks. To adapt their results to our static network setup in a fair way, we replace the parameter γ in [19] (see [19, equation (7)]) with $\mu(W)$. The references propose two variants of the accelerated algorithm: the first (see [19, (6a)–(6d)]) has k inner consensus iterations at the outer iteration k, while the second one has $\lceil 4\log(k+1)/(-\log\mu)\rceil$ (See [19, Subsection III-C].) The bounds established in [19] for the second variant give its rate: $O\left(\frac{N^2}{(1-\mu(W))^2\mathcal{K}^{2-\xi}}\right)$, when nodes know $\mu(W)$ and L. The first variant has a slower rate [18].

Algorithm Implementation and Convergence Rate: Table I compares D–NG, D–NC, the algorithm in [14] and the second algorithm in [19] with respect to implementation and the number of communications $\mathcal{K}(\epsilon; N, W)$ to achieve ϵ-accuracy. Here $\mathcal{K}(\epsilon; N, W)$ is the smallest number of communication rounds $\mathcal{K}$ after which $\frac{1}{N}(f(x_i) - f^\star) \leq \epsilon$, $\forall i$. Regarding implementation, we discuss the knowledge required a priori by all nodes for: 1) convergence (row 1); and 2) both stopping and optimizing the step-size (s.s.) (row 2). Stopping determines a priori the (outer) iteration k_0 such that $\frac{1}{N}(f(x_i(k)) - f^\star) \leq \epsilon$, $\forall k \geq k_0$, $\forall i$. Optimizing the step size here means finding the step-size that minimizes the established upper bound (in the reference of interest) on the optimality gap (e.g., the bound for D–NG in Theorem 5 (a).) We assume, with all methods, that W is already given (e.g., Metropolis.) Regarding $\mathcal{K}(\epsilon; N, W)$, we neglect the logarithmic and ξ-small factors and distinguish two cases: 1) the nodes have no global knowledge (row 3); and 2) the nodes know $L, \mu(W) =: \mu$ (row 4). We can see from Table I that, without global knowledge (row 3), D–NG has better dependence on ϵ than [14] and worse dependence on N, μ. Under global knowledge (row 4), D–NC has better complexity than [19] and has better dependence on ϵ, μ than [14] and a worse dependence on N. Further, while D–NG and [14] require no knowledge of any global parameters for convergence (row 1), D–NC and the second algorithm in [19] need L and $\mu(W)$. The first variant in [19] requires only L. Also, Table I for [14] holds for a wider class of functions, and in row 4, only μ is needed [14].

2) Global Knowledge $\mu(W), L, G, R$: (as needed, e.g., by D–NG for stopping) can be obtained as follows. Consider L and suppose each node knows a Lipschitz constant L_i of its own f_i. Then, L can be taken as $L = \max_{i=1,\ldots,N} L_i$. Thus, each node can compute L if nodes run a distributed algorithm for maximum computation, e.g., ([34, (1)]); all nodes get L after $O(\mathrm{Diam})$ per-node communicated scalars, where Diam is the

TABLE I
COMPARISONS OF ALGORITHMS D–NG, D–NC, [14], AND [19] (ALGORITHMS 1 AND 2)

	D–NG	D–NC	[14]	[19]
Kn. for conver.	none	L, μ	none	L, μ
Kn. for stop.; s.s.	μ, R, G, L, N	μ, R, G, L, N	μ, R, G, N	μ, R, G, L, N
$\mathcal{K}(\epsilon; N, W)$: No kn.	$\frac{1}{(1-\mu)^3\epsilon}+\frac{N^{1/3}}{(1-\mu)^2\epsilon^{2/3}}+\frac{N^{1/2}}{(1-\mu)^{3/2}\epsilon^{1/2}}$	not guarant.	$\frac{1}{(1-\mu)^2\epsilon^2}$	not studied
$\mathcal{K}(\epsilon; N, W)$: L, μ	$\frac{1}{(1-\mu)\epsilon}+\frac{N^{1/3}}{(1-\mu)^{2/3}\epsilon^{2/3}}+\frac{N^{1/2}}{(1-\mu)^{1/2}\epsilon^{1/2}}$	$\frac{1}{(1-\mu)\epsilon^{1/2}}+\frac{N^{1/6}}{(1-\mu)\epsilon^{1/3}}+\frac{N^{1/4}}{(1-\mu)\epsilon^{1/4}}$	$\frac{1}{(1-\mu)\epsilon^2}$	$\frac{N}{(1-\mu)\epsilon^{1/2}}$

network diameter. Likewise, a gradient bound G can be taken as $G = \max_{i=1,\ldots,N} G_i$, where G_i is a gradient bound for the f_i. The quantity $\mu(W)$ (equal to the second largest eigenvalue of W) can be computed in a distributed way, e.g., by algorithm DECENTRALOI, proposed in [35] and adapted to the problem like ours in [[36, Subsection IV-A, p. 2519]]. With DECENTRALOI, node i obtains q_i^μ, the i-th coordinate of the $N \times 1$ eigenvector q^μ of W that corresponds to $\mu(W)$, (up to ϵ-accuracy) after $O\left(\frac{\log^2(N/\epsilon)\log N}{1-\mu(W)}\right)$ per-node communicated scalars [35]; then, node i obtains $\mu(W)$ as: $\frac{\sum_{j\in O_i} W_{ij} q_j^\mu}{q_i^\mu}$.

Consider now D–NC when nodes do not have available their local gradient Lipschitz constants L_i. Nodes can take a diminishing step size $\alpha_k = 1/(k+1)^p$, $p \in (0, 1]$, and still guarantee convergence, with a deteriorated rate $O\left(\frac{1}{k^{2-p-\xi}}\right)$. In alternative, it may be possible to employ a "distributed line search," similarly to [37]. Namely, in the absence of knowledge of the gradient's Lipschitz constant L, the centralized Nesterov gradient method with a backtracking line search achieves the same rate $O(1/k^2)$, with an additional computational cost per iteration k; see [31], [38]. It is an interesting research direction to develop a variant of distributed line search for D–NC type methods and explore the amount of incurred additional communications/computations per outer iteration k; due to lack of space, this is left for future work.

3) The $\Omega(1/k^{2/3})$ Lower Bound on the Worst-Case Optimality Gap for [8]: We focus on the dependence on k and $\mathcal{K}$ only (assuming a finite, fixed $1/(1-\mu(W))$.) We demonstrate that D–NG has a strictly better worst-case convergence rate in k (and $\mathcal{K}$) than [8], when applied to the f_i's defined by Assumptions 2 and 3. Thus, D–NC also has a better rate.

Fix a generic, connected network $\mathcal{G}$ with N nodes and W that obeys Assumption 1. Let $\mathcal{F} = \mathcal{F}(L, G)$ be the class of all N-element sets of functions $\{f_i\}_{i=1}^N$, such that: 1) each $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex, has Lipschitz continuous derivative with constant L, and bounded gradient with bound G; and 2) Assumption 2 (a) holds. Consider (1) with $\{f_i\}_{i=1}^N \in \mathcal{F}$, for all i; consider D–NG with the step-size $\alpha_k = \frac{c}{(k+1)}$, $k = 0, 1, \ldots$, $c \le 1/(2L)$. Denote by

$$\mathcal{E}^{\mathrm{D-NG}}(k, R) = \sup_{\{f_i\}_{i=1}^N \in \mathcal{F}} \sup_{\{\overline{x}(0):\, \|\overline{x}(0)-x^\star\| \le R\}} \max_{i=1,\ldots,N} \{f(x_i(k)) - f^\star\}$$

the optimality gap at the k-th iteration of D–NG for the worst $\{f_i\}_{i=1}^N \in \mathcal{F}$, and the worst $\overline{x}(0)$ (provided $\|\overline{x}(0) - x^\star\| \le R$.) From Theorem 5 (a), *for any* $k = 1, 2, \ldots$: $\mathcal{E}^{\mathrm{D-NG}}(k, R) \le \mathcal{C}\frac{\log k}{k} = O(\log k/k)$, with $\mathcal{C}$ in (19). Now, consider the algorithm in [8] with the step-size $\alpha_k = \frac{c}{(k+1)^\tau}$, $k = 0, 1, \ldots$, where $c \in [\, c_0,\, 1/(2L)\,]$, $\tau \ge 0$ are the degrees of freedom, and c_0 is an arbitrarily small positive number. With this algorithm, $k = \mathcal{K}$. We show that, for the $N = 2$-node connected network, the weight matrix W with $W_{ii} = 7/8$, $i = 1, 2$, and $W_{ij} = 1/8$, $i \ne j$ (which satisfies Assumption 1), and $R = \sqrt{2}$, $L = \sqrt{2}$ and $G = 10$, with [8]

$$\inf_{\tau \ge 0,\, c \in [\, c_0,\, 1/(2L)\,]} \mathcal{E}(k, R; \tau, c) = \Omega\left(\frac{1}{k^{2/3}}\right) \tag{32}$$

where

$$\mathcal{E}(k, R; \tau, c) = \sup_{\{f_i\}_{i=1}^N \in \mathcal{F}} \sup_{\{\overline{x}(0):\, \|\overline{x}(0)-x^\star\| \le R\}} \max_{i=1,\ldots,N} \{f(x_i(k)) - f^\star\}$$

is the worst-case optimality gap when the step-size $\alpha_k = \frac{c}{(k+1)^\tau}$ is used. We perform the proof by constructing a "hard" example of the functions $f_i \in \mathcal{F}(L, G)$ and a "hard" initial condition to upper bound $\mathcal{E}(k, R; \tau, c)$; for any fixed k, c, τ, we set: $x_i(0) =: (1, 0)^\top$, $i = 1, 2$; $f_i =: f_i^{\theta_k}$, where

$$f_i^\theta(x) = \begin{cases} \frac{\theta(x^{(1)}+(-1)^i)^2}{2} + \frac{(x^{(2)}+(-1)^i)^2}{2} \\ \text{if } \theta(x^{(1)}+(-1)^i)^2 + (x^{(2)}+(-1)^i)^2 \le \overline{\chi}^2 \\ \overline{\chi}\left(\left[\theta(x^{(1)}+(-1)^i)^2 + (x^{(2)}+(-1)^i)^2\right]^{1/2} - \frac{\overline{\chi}}{2}\right) \\ \text{else;} \end{cases} \tag{33}$$

$\theta_k = \frac{1}{\sum_{t=0}^{k-1}(t+1)^{-\tau}}$; and $\overline{\chi} = 6$. The proof of (32) is in the Appendix. We convey here the underlying intuition. When τ is ϵ-smaller (away) from one, we show

$$\max_{i=1,2}(f^{\theta_k}(x_i(k)) - f^{\star,\theta_k}) \ge \Omega\left(\frac{1}{k^{1-\tau}} + \frac{1}{k^{2\tau}}\right).$$

The first summand is the "optimization term," for which a counterpart exists in the centralized gradient method also. The second, "distributed problem" term, arises because the gradients $\nabla f_i(x^\star)$ of the individual nodes functions are non-zero at the solution $x^\star$. Note the two opposing effects with respect to τ: $\frac{1}{k^{1-\tau}}$ (the smaller $\tau \ge 0$, the better) and $\frac{1}{k^{2\tau}}$ (the larger $\tau \ge 0$, the better.) To balance the opposing effects of the two summands, one needs to take a diminishing step-size; $\tau = 1/3$ strikes the needed balance to give the $\Omega(1/k^{2/3})$ bound.

B. Discussion on Assumptions

We now discuss what may occur if we drop each of the Assumptions made in our main results–Theorems 4 and 5 for D–NG, and Theorems 7 and 8 for D–NC.

Assumption 1(a): Consider Theorems 4 and 7. If Assumption 1(a) is relaxed, then $\widetilde{x}(k)$ with both methods may not converge to zero. Similarly, consider Theorems 5 and

8. Without Assumption 1(a), $f(x_i(k))$ may not converge to $f^\star$ at any node; e.g., take $N=2$, $W=I$, and f_i, $i=1,2$, in the next paragraph.

Assumption 1(b): Assumption 1(b) is imposed only for D–NG – Theorems 4 and 5. We show by simulation that, if relaxed, $\|\widetilde{x}(k)\|$ and $f(x_i(k))-f^\star$ may grow unbounded. Take $N=2$ and $W_{11}=W_{22}=1/10$, $W_{12}=W_{21}=9/10$; the Huber losses $f_i:\mathbb{R}\to\mathbb{R}$, $f_i(x)=\frac{1}{2}(x-a_i)^2$ if $\|x-a_i\|\leq 1$ and $f_i(x)=\|x-a_i\|-1/2$ else, $a_i=(-1)^{i+1}$; $c=1$, and $x(0)=y(0)=(0,0)^\top$. Then, we verify by simulation [18] that $\|\widetilde{x}(k)\|$ and $\min_{i=1,2}(f(x_i(k))-f^\star)$ grow unbounded.

Assumption 2: Assumption 2 is not needed for consensus with D–NG and D–NC (Theorems 4 and 7), but we impose it for Theorems 5 and 8 (convergence rates of D–NG and D–NC). This Assumption is standard and widely present in the convergence analysis of gradient methods, e.g., [17]. Nonetheless, we consider what may occur if we relax the requirement on the Lipschitz continuity of the gradient of the f_i's. For both D–NG and D–NC, we borrow the example functions $f_i:\mathbb{R}\to\mathbb{R}$, $i=1,2$, from [20, pages 29–31]: $f_1(x)=4x^3+\frac{3x^2}{2}$, $x\geq 1$; $f_1(x)=\frac{15x^2}{2}-2$, $x<1$; and $f_2(x):=f_1(-x)$. Then, for D–NG with $W_{11}=W_{22}=1-W_{12}=1-W_{21}=9/10$, $c=1$, and $x(0)=y(0)=(-1,1)^\top$, simulations show that $\|x(k)\|$ and $f(x_i(k))-f^\star$, $i=1,2$, grow unbounded. Similarly, with D–NC, for the same W, $\alpha=0.1$, and $x(0)=y(0)=(-1,1)^\top$, simulations show that $f(x_i(k))-f^\star$, $i=1,2$, stays away from zero when k grows [18].

Assumption 3: First consider Theorems 5 and 8 on the convergence rates of D–NG and D–NC. Define the class $\overline{\mathcal{F}}(L)$ to be the collection of all N-element sets of convex functions $\{f_i\}_{i=1}^N$, where each $f_i:\mathbb{R}^d\to\mathbb{R}$ has Lipschitz continuous gradient with constant L, and problem (2) is solvable in the sense of Assumption 2 (a). (Assumption 3 relaxed.) With the D–NC for the 2-node connected network, arbitrary weight matrix W obeying Assumption 1 (a), and the step-size $\alpha=1/(2L)$, we show for $L=1$, $R\geq 0$, that, for any $k\geq 10$ and arbitrarily large $M>0$

$$\mathcal{E}(k;R;\alpha=1/(2L))= \sup_{\{f_i\}\in\overline{\mathcal{F}}(L=1)}\ \sup_{\overline{x}(0):\|\overline{x}(0)-x^\star\|\leq R}\ \max_{i=1,2}(f(x_i(k))-f^\star)\geq M. \tag{34}$$

Note that the above means $\mathcal{E}(k;R;\alpha=1/(2L))=+\infty$, $\forall k\geq 10$, $\forall R\geq 0$. That is, no matter how large the (outer) iteration number k is, the worst case optimality gap is still arbitrarily large.

We conduct the proof by making a "hard" instance for $\{f_i\}_{i=1}^N$: for a fixed k,M, we set $x_i(0)=y_i(0)=0$, $i=1,2$, $f_i:\mathbb{R}\to\mathbb{R}$, to $f_i=f_i^{\theta(k,M)}$, where $\theta=\theta(k,M)=8\sqrt{M}\,k^2$ and

$$f_i^\theta(x)=\frac{1}{2}\left(x+(-1)^i\theta\right)^2,\quad i=1,2,\quad \theta>0. \tag{35}$$

Similarly to D–NC, with D–NG we show in [18] that (34) also holds for the 2-node connected network, the symmetric W with $W_{12}=W_{21}=1-W_{11}=1-W_{22}=\frac{1}{2}\left(1-10^{-6}\right)$ (this W obeys Assumption 1), $\alpha_k=c/(k+1)$, and $c=\frac{1}{4}\times 10^{-6}$. The candidate functions are in (35), where, for fixed $k\geq 5$, $M>0$, $\theta(k,M)=8\times 10^6\,k\sqrt{M}$.

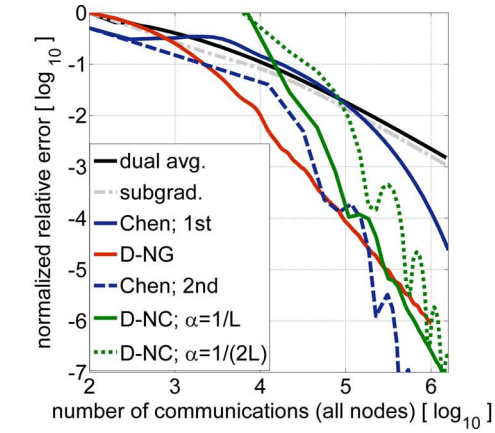
We convey here the intuition why (34) holds for D–NG and D–NC, while the proof is in the Appendix. Note that the solution to (1) with the f_i's in (35) is $x^\star=0$, while $x_i^\star:=\arg\min_{x\in\mathbb{R}}f_i(x)=(-1)^{i+1}\theta$, $i=1,2$. Making $x_1^\star$ and $x_2^\star$ to be far apart (by taking a large θ), problem (1) for D–NG and D–NC becomes "increasingly difficult." This is because the inputs to the disagreement dynamics (18)$(I-J)\nabla F(y(k-1))=(I-J)y(k-1)-(-\theta,\theta)^\top$ are arbitrarily large, even when $y(k-1)$ is close to the solution $y(k-1)\approx(0,0)^\top$.

Finally, we consider what occurs if we drop Assumption 3 with Theorems 4 and 7. We show with D–NG and the above "hard" examples that $\|\widetilde{x}(k)\|\geq\frac{\sqrt{2}\,c\,\theta}{2\,k}$, $\forall k\geq 5$. Hence, $\|\widetilde{x}(k)\|$ is arbitrarily large by choosing θ large enough. (see [18].) Similarly, with D–NC: $\|\widetilde{x}(k)\|\geq\frac{\alpha\,\theta\sqrt{2}}{4\,k^2}$, $\forall k\geq 10$. (see Appendix C and [18].)

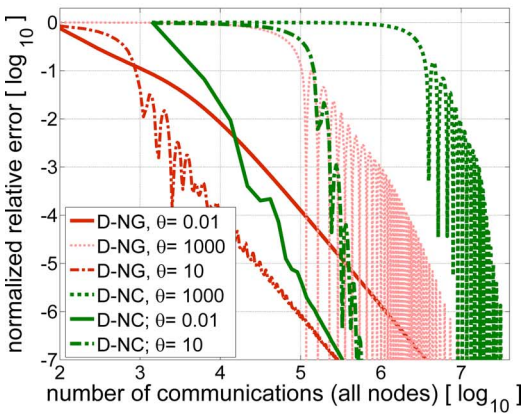
VIII. Simulations

We compare the proposed D–NG and D–NC algorithms with [8], [14], [19] on the logistic loss. Simulations confirm the increased convergence rates of D–NG and D–NC with respect to [8], [14] and show a comparable performance with respect to [19]. More precisely, D–NG achieves an accuracy ϵ faster than [8], [14] for all ϵ, while D–NC is faster than [8], [14] at least for $\epsilon\leq 10^{-2}$. With respect to [19], D–NG is faster for lower accuracies (ϵ in the range 10^{-1} to $10^{-4}-10^{-5}$), while [19] becomes faster for high accuracies ($10^{-4}-10^{-5}$ and finer); D–NC performs slower than [19].

1) Simulation Setup: We consider distributed learning via the logistic loss; see, e.g., [7] for further details. Nodes minimize the logistic loss: $f(x)=\sum_{i=1}^N f_i(x)=\sum_{i=1}^N\log\left(1+e^{-b_i(a_i^\top x_1+x_0)}\right)$, where $x=(x_1^\top,x_2^\top)^\top$, $a_i\in\mathbb{R}^2$ is the node i's feature vector, and $b_i\in\{-1,+1\}$ is its class label. The functions $f_i:\mathbb{R}^d\mapsto\mathbb{R}$, $d=3$, satisfy Assumptions 2 and 3. The Hessian $\nabla^2 f(x)=\sum_{i=1}^N\frac{e^{-c_i^\top x}}{(1+e^{-c_i^\top x})^2}c_ic_i^\top$, where $c_i=(b_ia_i^\top,b_i)^\top\in\mathbb{R}^3$. A Lipschitz constant L should satisfy $\|\nabla^2 f(x)\|\leq NL$, $\forall x\in\mathbb{R}^d$. Note that $\nabla^2 f(x)\preceq\frac{1}{4}\sum_{i=1}^N c_ic_i^\top$, because $\frac{e^{-c_i^\top x}}{(1+e^{-c_i^\top x})^2}\leq 1/4$, $\forall x$. We thus choose $L=\frac{1}{4N}\left\|\sum_{i=1}^N c_ic_i^\top\right\|\approx 0.3053$. We generate a_i independently over i; each entry is drawn from the standard normal distribution. We generate the "true" vector $x^\star=(x_1^{\star\top},x_0^\star)^\top$ by drawing its entries independently from the standard normal distribution. The labels are $b_i=\text{sign}\left({x^\star_1}^\top a_i+{x^\star}_0+\epsilon_i\right)$, where the ϵ_i's are drawn independently from a normal distribution with zero mean and variance 3. The network is a geometric network: nodes are placed uniformly randomly on a unit square and the nodes whose distance is less than a radius are connected by an edge. There are $N=100$ nodes, and the relative degree $\left(=\frac{\text{number of links}}{N(N-1)/2}\right)\approx 10\%$. We initialize all nodes by $x_i(0)=0$ (and $y_i(0)=0$ with D–NG, D–NC, and [19]). With all algorithms except D–NG, we use the Metropolis weights W [28]; with D–NG, we use $W'=\frac{1+\eta}{2}I+\frac{1-\eta}{2}W$, with $\eta=0.1$. The step-size α_k is: $\alpha_k=1/(k+1)$, with D–NG; $\alpha=1/(2L)$ and $1/L$, with D–NC; $1/L$, with [19] (both the first and second algorithm variants – see Section VII-A); and



(a)



(b)

Fig. 1. Normalized (average) relative error $\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)-f^\star}{f(x_i(0))-f^\star}$ versus the number of communications (all nodes) $N\mathcal{K}$; Top: Logistic loss; Bottom: Huber loss.

$1/(k+1)^{1/2}$, with [8] and [14].[1] We simulate the normalized (average) error $\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)-f^\star}{f(x_i(0))-f^\star}$ versus the total number of communications at all nodes ($=N\mathcal{K}$.)

2) Results: Fig. 1 (top) compares D–NG, D–NC (with step-sizes $\alpha=1/(2L)$ and $1/L$), [8], [14], [19] (both first and second variant with $\alpha=1/L$.) We can see that D–NG converges faster than other methods for accuracies ϵ in the range 10^{-1} to $3\cdot 10^{-5}$. For example, for $\epsilon=10^{-2}$, D–NG requires about 10^4 transmissions; [19] (second variant) $\approx 3.16\cdot 10^4$; D–NC ($\alpha=1/L$) $\approx 4.65\cdot 10^4$, and D–NC with $\alpha=1/(2L)\approx 1.1\cdot 10^5$; and [19] (first variant), [8], and [14] – at least $\approx 1.3\cdot 10^5$. For high accuracies, $2\cdot 10^{-5}$ and finer, [19] (second variant) becomes faster than D–NG. Finally, [19] (second) converges faster than D–NC, while [19] (first) is slower than D–NC.

3) Further Comparisons of D–NG and D–NC: Huber Loss: We provide an additional experiment to further compare the D–NG and D–NC methods. We show that the relative performance of D–NC with respect to D–NG improves when the instance of (1) becomes easier (in the sense explained below.) We consider a $N=20$-node geometric network with $\frac{\text{number of links}}{N(N-1)/2}\approx 32\%$ and Huber losses $f_i:\mathbb{R}\to\mathbb{R}$, $f_i(x)=\frac{1}{2}\|x-a_i\|^2$ if $\|x-a_i\|\leq 1$, and $f_i(x)=\|x-a_i\|-1/2$, else, with $a_i\in\mathbb{R}$. We divide the set of nodes in two groups. For the first group, $i=1,\ldots,6$, we generate the a_i's as $a_i=\theta+\nu_i$, where $\theta>0$ is the "signal" and ν_i is the uniform noise on $[-0.1\theta,\,0.1\theta]$. For the second group, $i=7,\ldots,20$, we set $a_i=-\theta+\nu_i$, with the ν_i's from the same uniform distribution. Note that any $x_1^\star\in\arg\min_{x\in\mathbb{R}}\sum_{i=1}^{6}f_i(x)$ is in $[0.9\theta,\,1.1\theta]$, while any $x_2^\star\in\arg\min_{x\in\mathbb{R}}\sum_{i=7}^{20}f_i(x)$ lies in $[-1.1\theta,\,-0.9\theta]$. Intuitively, by making $\theta>0$ large, we increase the problem difficulty. For a small θ, we are in the "easy problem" regime, because the solutions $x_1^\star$ and $x_2^\star$ of the two nodes' groups are close; for a large θ, we are in the "difficult problem" regime. Fig. 1 (bottom) plots the normalized average error versus $N\mathcal{K}$ for $\theta\in\{0.01;10;1000\}$ for D–NG with $\alpha_k=1/(k+1)$, D–NC with $\alpha=1/L$, while both algorithms are initialized by $x_i(0)=y_i(0)=0$. We can see that, with D–NC, the decrease of θ makes the convergence faster, as expected. (With D–NG, it is not a clear "monotonic" behavior.) Also, as θ decreases ("easier problem"), the performance of D–NC relative do D–NG improves. For $\theta=0.01$, D–NG is initially better, but the curves of D–NG and D–NC intersect at the value about $4\cdot 10^{-3}$, while for $\theta=1000$, D–NG is better for all accuracies as fine as (at least) 10^{-7}.

We give an intuition on the observed behavior. Consider an "easy" problem with very similar local costs (small θ). In such scenario, over outer iterations k D–NC behaves very similarly to the exact centralized Nesterov gradient method *with a constant step-size* α. However, during each k, D–NC uses $\tau_x(k)+\tau_y(k)$ per-node communications which, for the "easy" problem, are unnecessary and "waste" resources. (These communications are necessary for "difficult" problems.) Hence, D–NC behaves here as the centralized Nesterov gradient method slowed (re-scaled) through (unnecessary) multiple consensus rounds. From the above, it may seem intuitive that the relative performance of D–NC over D–NG is poorer for "easy" problems due to "wastes" in communications; but this does not occur in simulations. To explain why, consider now D–NG for the same "easy" problem. It behaves over k similarly to the exact centralized Nesterov gradient method *with a diminishing step-size* $1/k$. Hence, not only D–NC behaves as a suboptimal centralized gradient method (due to multiple consensus rounds), but also D–NG does, with the source of sub-optimality being the diminishing step-size $1/k$. An intuitive comparison of these two suboptimal methods on "easy" problems is the following. For a given network (given $\mu(W)$), it is natural to expect that D–NC converges at a faster rate (steeper slope) than D–NG, but with the curve "shifted" upwards due to the effect of $\tau_x(k)+\tau_y(k)$. We indeed observe such behavior in Fig. 1, bottom, case $\theta=0.01$. On the other hand, for "difficult" problems (large θ), the dynamics of disagreements play a significant role and cannot be neglected. Hence, it is much harder to intuitively understand the behavior. As our simulation example indicates, for more "difficult" problems (larger θ), the performance of D–NC relative to D–NG actually deteriorates. We also performed a simulation with a deteriorated $\mu(W)$, while all other parameters are the same as in the above simulation. We increase $\mu(W)$ by setting, with both D–NG and D–NC, $W''=0.9I+0.1W$, where W is the Metropolis matrix. The relative behavior of D–NC with respect to D–NG still deteriorates with the increase of θ. (Figure omitted due to lack of space.)

[1]With [8], [14], $\alpha_k=1/(k+1)^p$ and $p=1/2$, gave the best simulation performance among the choices $p\in\{1/3,1/2,1\}$.

IX. Conclusion

We propose fast distributed gradient algorithms for nodes in a network to minimize the sum of their individual cost functions. Existing literature has presented distributed gradient based algorithms to solve this problem and has studied their convergence rates, for a class of convex, non-differentiable costs, with bounded gradients. We asked whether faster convergence rates than the rates established in the literature can be achieved for more structured costs – convex, with Lipschitz continuous gradient (with constant L) and bounded gradient. Building from the centralized Nesterov gradient method, we answer affirmatively this question by proposing two distributed gradient algorithms. Our algorithm D–NG achieves the rates $O\left(\frac{\log \mathcal{K}}{\mathcal{K}}\right)$ and $O\left(\frac{\log k}{k}\right)$. Our algorithm D–NC operates only if L and $\mu(W)$ are available and achieves rates $O\left(\frac{1}{\mathcal{K}^{2-\varepsilon}}\right)$ and $O\left(\frac{1}{k^2}\right)$. We also found convergence constants in terms of the network parameters. Simulations illustrate the performance of the proposed methods.

Appendix

A. Proof of Theorem 4

For notational simplicity, we let $d = 1$, but the proof extends to $d > 1$. We outline the main steps in the proof. First, we unwind the recursion (18) and calculate the underlying time varying system matrices. Second, we upper bound the norms of the time varying system matrices. Finally, we use these bounds and a summation argument to complete the proof of the Theorem.

1) Unwinding (18) and Calculating the System Matrices: Define the $2N \times 2N$ system matrices

$$\Phi(k,t) := \Pi_{s=2}^{k-t+1} \begin{bmatrix} (1+\beta_{k-s})\widetilde{W} & -\beta_{k-s}\widetilde{W} \\ I & 0 \end{bmatrix},\ k > t \tag{36}$$

and $\Phi(k,k) = I$. Unwinding (18), the solution to (18) is

$$\begin{aligned}(\widetilde{x}^\top(k),\, \widetilde{x}^\top(k-1))^\top = \sum_{t=0}^{k-1} \Phi(k,t+1)\,\alpha_t \\ \times (\,(-\nabla F(y(t)))^\top (I-J), 0\,)^\top,\ k = 1, 2 \ldots .\end{aligned} \tag{37}$$

We now show the interesting structure of the matrix $\Phi(k,t)$ in (36) by decomposing it into the product of an orthonormal matrix U, a block-diagonal matrix, and $U^\top$. While U is independent of k and t, the block diagonal matrix depends on k and t, and has 2×2 diagonal blocks. Consider the matrix in (18) with $k - 2 = t$, for a generic $t = -1, 0, 1, \ldots$ Using $\widetilde{W} = Q\widetilde{\Lambda}Q^\top$

$$\begin{aligned}&\begin{bmatrix} (1+\beta_t)\widetilde{W} & -\beta_t\widetilde{W} \\ I & 0 \end{bmatrix} \\ &= (Q \oplus Q)\, P \left(\oplus_{i=1}^N \Sigma_i(t)\right) P^\top (Q \oplus Q)^\top\end{aligned} \tag{38}$$

where P is the $2N \times 2N$ permutation matrix (e_i here is the i–th column of the $2N \times 2N$ identity matrix) $P = [e_1, e_{N+1}, e_2, e_{N+2}, \ldots, e_N, e_{2N}]^\top$, and $\Sigma_i(t)$ is a 2×2 matrix with $= (1+\beta_t)\lambda_i(\widetilde{W})$, $[\Sigma_i(t)]_{12} = -\beta_t\lambda_i(\widetilde{W})$, $[\Sigma_i(t)]_{21} = 1$, and $[\Sigma_i(t)]_{22} = 0$. Using (38), and the fact that $(Q \oplus Q)P$ is orthonormal:$((Q \oplus Q)P) \cdot ((Q \oplus Q)P)^\top = (Q \oplus Q)PP^\top(Q \oplus Q)^\top = (QQ^\top) \oplus (QQ^\top) = I$, we can express $\Phi(k,t)$ in (36) as

$$\begin{aligned}&\Phi(k,t) := \\ &(Q \oplus Q)P \left(\oplus_{i=1}^N \Pi_{s=2}^{k-t+1} \Sigma_i(k-s)\right) P^\top (Q \oplus Q)^\top \\ &\text{for } k > t;\ \ \Phi(k,k) = I.\end{aligned} \tag{39}$$

2) Bounding the Norm of $\Phi(k,t)$: As $(Q \oplus Q)P$ is orthonormal, $\Phi(k,t)$ has the same singular values as $\oplus_{i=1}^N \Pi_{s=2}^{k-t+1} \Sigma_i(k-s)$, and so these two matrices also share the same spectral norm (maximal singular value.) Further, the matrix $\oplus_{i=1}^N \Pi_{s=2}^{k-t+1} \Sigma_i(k-s)$ is block diagonal (with 2×2 blocks $\Pi_{s=2}^{k-t+1} \Sigma_i(k-s)$), and so: $\|\Phi(k,t)\| = \max_{i=1,\ldots,N} \left\|\Pi_{s=2}^{k-t+1} \Sigma_i(k-s)\right\|$. We proceed by calculating $\left\|\Pi_{s=2}^{k-t+1} \Sigma_i(k-s)\right\|$. We distinguish two cases: $i = 1$, and $i > 1$.

Case $i = 1$: As $\lambda_1(\widetilde{W}) = 0$, for all t, $\Sigma_1(t) = \Sigma_1$ is a constant matrix, with $[\Sigma_1]_{21} = 1$, and the entries $(1,1)$, $(1,2)$ and $(2,2)$ of Σ_1 are zero. Note that $\|\Sigma_1\| = 1$, and $(\Sigma_1)^s = 0$, $s \geq 2$. Thus, as long as $k > t+1$, the product $\Pi_{s=2}^{k-t+1} \Sigma_i(k-s) = 0$, and so

$$\left\|\Pi_{s=2}^{k-t+1} \Sigma_1(k-s)\right\| = \begin{cases} 1 & \text{if } k = t+1 \\ 0 & \text{if } k > t+1. \end{cases} \tag{40}$$

Case $i > 1$: To simplify notation, let $\lambda_i := \lambda_i(\widetilde{W})$, and recall $\lambda_i \in (0,1)$; $\Sigma_i(t)$ is: $\Sigma_i(t) = \widehat{\Sigma}_i - \frac{3}{t+3}\Delta_i$, where: 1) $[\widehat{\Sigma}_i]_{11} = 2\lambda_2$, $[\widehat{\Sigma}_i]_{12} = -\lambda_i$, $[\widehat{\Sigma}_i]_{21} = 1$, and $[\widehat{\Sigma}_i]_{22} = 0$; and 2) $[\Delta_i]_{11} = -[\Delta_i]_{12} = \lambda_i$, and $[\Delta_i]_{21} = [\Delta_i]_{22} = 0$.; $\widehat{\Sigma}_i$ is diagonalizable, with $\widehat{\Sigma}_i = \widehat{\mathcal{Q}_i}\widehat{\mathcal{D}}_i\widehat{\mathcal{Q}_i}^{-1}$, and

$$\widehat{\mathcal{Q}_i} = \begin{bmatrix} \lambda_i + \mathbf{j}\sqrt{\lambda_i(1-\lambda_i)} & \lambda_i - \mathbf{j}\sqrt{\lambda_i(1-\lambda_i)} \\ 1 & 1 \end{bmatrix}$$

$$\widehat{\mathcal{D}}_i = \begin{bmatrix} \lambda_i + \mathbf{j}\sqrt{\lambda_i(1-\lambda_i)} & 0 \\ 0 & \lambda_i - \mathbf{j}\sqrt{\lambda_i(1-\lambda_i)} \end{bmatrix}.$$

(Note that the matrices $\widehat{\mathcal{Q}}_i$ and $\widehat{\mathcal{D}}_i$ are complex.) Denote by $\mathcal{D}_i(t) = \widehat{\mathcal{D}}_i - \frac{3}{t+3}\widehat{\mathcal{Q}_i}^{-1}\Delta_i\widehat{\mathcal{Q}_i}$. Then, $\Sigma_i(t) = \widehat{\mathcal{Q}}_i\left(\widehat{\mathcal{D}}_i - \frac{3}{t+3}\widehat{\mathcal{Q}}_i^{-1}\Delta_i\widehat{\mathcal{Q}}_i\right)\widehat{\mathcal{Q}}_i^{-1} = \widehat{\mathcal{Q}_i}\mathcal{D}_i(t)\widehat{\mathcal{Q}_i}^{-1}$. By the sub-multiplicative property of norms, and using $\left\|\widehat{\mathcal{Q}_i}\right\| \leq \sqrt{2}\left\|\widehat{\mathcal{Q}_i}\right\|_\infty = 2\sqrt{2}$, $\left\|\widehat{\mathcal{Q}_i}^{-1}\right\| \leq \sqrt{2}\left\|\widehat{\mathcal{Q}_i}^{-1}\right\|_\infty = \frac{2\sqrt{2}}{\sqrt{\lambda_i(1-\lambda_i)}}$

$$\|\Pi_{s=2}^{k-t+1}\Sigma_i(k-s)\| \leq \frac{8}{\sqrt{\lambda_i(1-\lambda_i)}}\,\Pi_{s=2}^{k-t+1}\|\mathcal{D}_i(k-s)\|. \tag{41}$$

It remains to upper bound $\|\mathcal{D}_i(t)\|$, for all $t = -1, 0, 1, \ldots$ We will show that

$$\|\mathcal{D}_i(t)\| \leq \sqrt{\lambda_i},\ \ \forall t = -1, 0, 1, \ldots . \tag{42}$$

Denote by $a_t = \frac{3}{t+3}$, $t = 0, 1, \ldots$, and $a_{-1} = 1$. After some algebra, the entries of $\mathcal{D}_i(t)$ are: $[\mathcal{D}_i(t)]_{11} = ([\mathcal{D}_i(t)]_{22})^H = \frac{1}{2}(2-a_t)(\lambda_i + \mathbf{j}\sqrt{\lambda_i(1-\lambda_i)})$, $[\mathcal{D}_i(t)]_{12} = ([\mathcal{D}_i(t)]_{21})^H = a_t(\lambda_i + \mathbf{j}\sqrt{\lambda_i(1-\lambda_i)})$, which gives: $[\mathcal{D}_i(t)^H\mathcal{D}_i(t)]_{11} = [\mathcal{D}_i(t)^H\mathcal{D}_i(t)]_{22} = \frac{a_t^2+(2-a_t)^2}{4}\lambda_i$, and $[\mathcal{D}_i(t)^H\mathcal{D}_i(t)]_{12} = \left([\mathcal{D}_i(t)^H\mathcal{D}_i(t)]_{21}\right)^H = \frac{a_t(2-a_t)}{2}\left(2\lambda_i^2 - \lambda_i - 2\mathbf{j}\lambda_i\sqrt{\lambda_i(1-\lambda_i)}\right)$. Next, interestingly:

$\|\mathcal{D}_i{}^H(t)\mathcal{D}_i(t)\|_1 = \left\|\left[\mathcal{D}_i{}^H(t)\mathcal{D}_i(t)\right]_{11}\right\| + \left\|\left[\mathcal{D}_i{}^H(t)\mathcal{D}_i(t)\right]_{12}\right\| = \frac{1}{4}(a_t^2 + (2-a_t)^2)\lambda_i + \frac{1}{2}a_t(2-a_t)\lambda_i = \lambda_i$ for any $a_t \in [0,2]$, which is the case here because $a_t = 3/(t+3)$, $t = 0, 1, \ldots$, and $a_{-1} = 0$. Thus, as $\|A\| \leq \|A\|_1$ for a Hermitean matrix A: $\|\mathcal{D}_i(t)\| = \sqrt{\|\mathcal{D}_i{}^H(t)\mathcal{D}_i(t)\|} \leq \sqrt{\|\mathcal{D}_i{}^H(t)\mathcal{D}_i(t)\|_1} = \sqrt{\lambda_i}$. Applying the last equation and (42) to (41), we get, for $i \neq 1$: $\|\Pi_{s=2}^{k-t+1}\Sigma_i(k-s)\| \leq \frac{8}{\sqrt{\lambda_i(1-\lambda_i)}}\left(\sqrt{\lambda_i}\right)^{k-t}$, $k \geq t+1$. Combine the latter with (40), and use $\|\Phi(k,t)\| = \max_{i=1,\ldots,N}\|\Pi_{s=2}^{k-t+1}\Sigma_i(k-s)\|$, Assumption 1(b) and $\lambda_N(\widetilde{W}) = \mu(W)$, to obtain

$$\|\Phi(k,t)\| \leq \frac{8\left(\sqrt{\mu(W)}\right)^{k-t}}{\min_{i\in\{2,N\}}\sqrt{\lambda_i(\widetilde{W})(1-\lambda_i(\widetilde{W}))}} \leq \frac{8}{\sqrt{\eta(1-\mu(W))}}\left(\sqrt{\mu(W)}\right)^{k-t},\ k \geq t. \quad (43)$$

3) Summation: We apply (43) to (37). Using the sub-multiplicative and sub-additive properties of norms, expression $\alpha_t = c/(t+1)$, and the inequalities $\|\widetilde{x}(k)\| \leq \left\|(\widetilde{x}(k)^\top, \widetilde{x}(k-1)^\top)^\top\right\|$, $\left\|(-(I-J)\nabla F(y(t))^\top, 0^\top)^\top\right\| \leq \sqrt{N}\,G$

$$\|\widetilde{x}(k)\| \leq \frac{8\sqrt{N}\,c\,G}{\sqrt{\eta(1-\mu(W))}} \times \sum_{t=0}^{k-1}\left(\sqrt{\mu(W)}\right)^{k-(t+1)}\frac{1}{(t+1)}. \quad (44)$$

We now denote by $r := \sqrt{\mu(W)} \in (0,1)$. To complete the proof of the Lemma, we upper bound the sum $\sum_{t=0}^{k-1} r^{k-(t+1)}\frac{1}{(t+1)}$ by splitting it into two sums. With the first sum, t runs from zero to $\lceil k/2\rceil$, while with the second sum, t runs from $\lceil k/2\rceil + 1$ to k

$$\begin{aligned}
&\sum_{t=0}^{k-1}\frac{r^{k-(t+1)}}{t+1} = \left(r^{k-1} + r^{k-2}\frac{1}{2} + \ldots + r^{\lceil k/2\rceil}\frac{1}{\lceil k/2\rceil}\right) \\
&+ \left(r^{k-(\lceil k/2\rceil+1)}\frac{1}{\lceil k/2\rceil+1} + \ldots + \frac{1}{k}\right) \\
&\leq r^{k/2}\left(1 + \frac{1}{2} + \ldots + \frac{1}{k/2} + \frac{1}{(k+1)/2}\right) \\
&+ \frac{1}{(k/2)}\left(1 + r + \ldots + r^k\right) \\
&\leq r^{k/2}\left(\log(1+k/2) + 2\right) + \frac{2}{k}\frac{1}{1-r} \quad (45) \\
&= 2\left\{r^{k/2}\log(1+k/2)(k/2)\right\}\frac{1}{k} + \left\{4r^{k/2}(k/2)\right\}\frac{1}{k} \\
&+ \frac{2}{k}\frac{1}{1-r} \quad (46) \\
&\leq 2\sup_{z\geq 1/2}\left\{r^z\log(1+z)z\right\}\frac{1}{k} + 4\sup_{z\geq 1/2}\left\{r^z z\right\}\frac{1}{k} \\
&+ \frac{2}{k}\frac{1}{1-r} \quad (47) \\
&\leq \left(2\mathcal{B}(r) + \frac{4}{e(-\log r)} + \frac{2}{1-r}\right)\frac{1}{k} \quad (48) \\
&\leq \left(2\mathcal{B}(r) + \frac{7}{1-r^2}\right)\frac{1}{k}.
\end{aligned}$$

Inequality (45) uses the inequality $1 + \frac{1}{2} + \ldots + \frac{1}{t} \leq \log t + 1$, $t = 1, 2, \ldots$, and $1 + r + \ldots + r^k \leq \frac{1}{1-r}$; (46) multiplies and divides the first summand on the right hand side of (45) by $k/2$; (47) uses $r^{k/2}\log(1+k/2)(k/2) \leq \sup_{z\geq 1/2} r^z\log(1+z)z$, for all $k = 1, 2, \ldots$, and a similar bound for the second summand in (46); the left inequality in (48) uses $\mathcal{B}(r) := \sup_{z\geq 1/2} r^z\log(1+z)z$ and $\sup_{z\geq 1/2} r^z z \leq \frac{1}{e\,(-\log r)}$ (note that $r^z z$ is convex in z; we take the derivative of $r^z z$ with respect to z and set it to zero); and the right inequality in (48) uses $-1/\log r \leq 1/(1-r)$, $\forall r \in [0,1)$; $1/(1-r) \leq 2/(1-r^2)$, $\forall r \in [0,1)$, and $e = 2.71\ldots$ Applying the last to (44), and using the C_{cons} in (17), Theorem 4 for $\|\widetilde{x}(k)\|$ follows. Then, as $\widetilde{y}(k) = \widetilde{x}(k) + \frac{k-1}{k+2}(\widetilde{x}(k) - \widetilde{x}(k-1))$, we have that $\|\widetilde{y}(k)\| \leq 2\|\widetilde{x}(k)\| + \|\widetilde{x}(k-1)\|$. Further, by Theorem 4: $\|\widetilde{x}(k-1)\| \leq c\sqrt{N}GC_{\text{cons}}\frac{1}{k-1}\frac{k}{k} \leq 2c\sqrt{N}GC_{\text{cons}}\frac{1}{k}$, $k \geq 2$, and $\|\widetilde{x}(0)\| = 0$ (by assumption). Thus, $\|\widetilde{x}(k-1)\| \leq 2c\sqrt{N}GC_{\text{cons}}\frac{1}{k}$, $\forall k \geq 1$. Thus, $\|\widetilde{y}(k)\| \leq 2\|\widetilde{x}(k)\| + \|\widetilde{x}(k-1)\| \leq 4c\sqrt{N}GC_{\text{cons}}\frac{1}{k}$, $\forall k \geq 1$.

B. Proof of the Lower Bound in (32) on the Worst-Case Optimality Gap for [8]

Consider the f_i's in (33), the initialization $x_i(0) = (1,0)^\top$, $i = 1, 2$, and $W_{12} = W_{21} = 1 - W_{11} = 1 - W_{22} = w = 1/8$, as we set in Section VII-A. We divide the proof in four steps. First, we prove certain properties of (1) and the f_i's in (33); second, we solve for the state $x(k) = (x_1(k)^\top, x_2(k)^\top)^\top$ with the algorithm in [8]; third, we upper bound $\|x(k)\|$; finally, we use the latter bound to derive the $\Omega(1/k^{2/3})$ worst-case optimality gap.

Step 1: Properties of the f_i^θ's: Consider the f_i^θ's in (33) for a fixed $\theta \in [0,1]$. The solution to (1), with $f(x) = f_1^\theta(x) + f_2^\theta(x)$, is $x^\star = (0,0)^\top$, and the corresponding optimal value is $f^\star = \theta + 1$. Further, the f_i^θ's belong to the class $\mathcal{F}(L = \sqrt{2}, G = 10)$. (Proof is in [18].)

Step 2: Solving for $x(k)$ With the Algorithm in [8]: Now, consider the algorithm in [8], and consider $x_i(k)$–the solution estimate at node i and time k. Denote by $x^l(k) = (x_1^{(l)}(k), x_2^{(l)}(k))^\top$–the vector with the l-th coordinate of the estimate of both nodes, $l = 1, 2$; and $d^l(k) = \left(\frac{\partial f_1(x_1(k))}{\partial x^{(l)}}, \frac{\partial f_2(x_2(k))}{\partial x^{(l)}}\right)^\top$, $l = 1, 2$. Then, the update rule of [8] is, for the f_1^θ, f_2^θ in (33)

$$x^l(k) = Wx^l(k-1) - \alpha_{k-1}d^l(k-1) \quad k = 1, 2, \ldots, \quad l = 1, 2. \quad (49)$$

Recall the "hard" initialization $x^{\text{I}}(0) = (1,1)^\top$, $x^{\text{II}}(0) = (0,0)^\top$. Under this initialization

$$x_i(k) \in \mathcal{R}_i := \left\{x \in \mathbb{R}^2 : \theta(x^{(1)} + (-1)^i)^2 + (x^{(2)} + (-1)^i)^2 \leq \overline{\chi}^2\right\} \quad (50)$$

for all k, for both nodes $i = 1, 2$ (proof in [18].) Note that $\mathcal{R}_i$ is the region where the f_i^θ in (33) is quadratic. Thus, evaluating ∇f_i^θ's in the quadratic region

$$x^l(k) = \left(W - \alpha_{k-1}\kappa^l I\right)x^{\text{I}}(k-1) - \alpha_{k-1}\kappa^l(-1,1)^\top \quad (51)$$

$l = 1, 2$, where $\kappa^{\mathrm{I}} = \theta$ and $\kappa^{\mathrm{II}} = 1$. We now evaluate $\sum_{i=1}^{2} (f(x_i(k)) - f^\star)$, $f(x) = f_1^\theta(x) + f_2^\theta(x)$. Because $x_i(k) \in \mathcal{R}_i$, $i = 1, 2$, verify, using (33), and $f^\star = 1 + \theta$, that

$$\sum_{i=1}^{2} (f(x_i(k)) - f^\star) = \theta \|x^{\mathrm{I}}(k)\|^2 + \|x^{\mathrm{II}}(k)\|^2. \tag{52}$$

By unwinding (51), and using $x^{\mathrm{I}}(0) = (1,1)^\top$, $x^{\mathrm{II}}(0) = (0,0)^\top$

$$\begin{aligned} x^{\mathrm{I}}(k) = & \\ & (W - \alpha_{k-1}\theta I)(W - \alpha_{k-2}\theta I) \dots (W - \alpha_0 \theta I)(1,1)^\top \\ & + \theta \Big(\sum_{t=0}^{k-2} (W - \alpha_{k-1}\theta I)(W - \alpha_{k-2}\theta I) \dots \\ & \times (W - \alpha_{t+1}\theta I)\alpha_t + \alpha_{k-1} I)(1,-1)^\top \\ x^{\mathrm{II}}(k) = & \\ & \Big(\sum_{t=0}^{k-2} (W - \alpha_{k-1} I)(W - \alpha_{k-2} I) \dots \\ & \times (W - \alpha_{t+1} I)\alpha_t + \alpha_{k-1} I)(1,-1)^\top. \end{aligned}$$

Consider the eigenvalue decomposition $W = Q \Lambda Q^\top$, where $Q = [q_1, q_2]$, $q_1 = \frac{1}{\sqrt{2}}(-1,1)^\top$, $q_2 = \frac{1}{\sqrt{2}}(1,1)^\top$, and Λ is diagonal with the eigenvalues $\Lambda_{11} = \lambda_1 = 1 - 2w = 3/4$, $\Lambda_{22} = \lambda_2 = 1$. The matrix $W - \alpha_{k-1}\theta I$ decomposes as $W - \alpha_{k-1}\theta I = Q(\Lambda - \alpha_{k-1}\theta I)Q^\top$; likewise, $W - \alpha_{k-1} I = Q(\Lambda - \alpha_{k-1} I)Q^\top$. Then, $(W - \alpha_{k-1}\theta I)(W - \alpha_{k-2}\theta I) \dots (W - \alpha_{t+1}\theta I) = Q(\Lambda - \alpha_{k-1}\theta I) \dots (\Lambda - \alpha_{t+1}\theta I)Q^\top$, and $(W - \alpha_{k-1} I) \dots (W - \alpha_{t+1} I) = Q(\Lambda - \alpha_{k-1} I) \dots (\Lambda - \alpha_{t+1} I)Q^\top$. Using these decompositions, and the orthogonality: $q_1^\top (1,1)^\top = 0$, and $q_2^\top (-1,1)^\top = 0$

$$\begin{aligned} x^{\mathrm{I}}(k) = & (1 - \alpha_{k-1}\theta)(1 - \alpha_{k-2}\theta) \dots (1 - \alpha_0 \theta)(1,1)^\top \\ & + \theta (1,-1)^\top \Big(\sum_{t=0}^{k-2} (\lambda_1 - \alpha_{k-1}\theta)(\lambda_1 - \alpha_{k-2}\theta) \dots \\ & \times (\lambda_1 - \alpha_{t+1}\theta)\alpha_t + \alpha_{k-1}) \end{aligned} \tag{53}$$

$$\begin{aligned} x^{\mathrm{II}}(k) = & (1,-1)^\top \Big(\sum_{t=0}^{k-2} (\lambda_1 - \alpha_{k-1})(\lambda_1 - \alpha_{k-2}) \dots \\ & \times (\lambda_1 - \alpha_{t+1})\alpha_t + \alpha_{k-1}). \end{aligned} \tag{54}$$

Step 3: Upper Bounding $\|x(k)\|$: Note that $\lambda_1 - \alpha_{k-1}\theta = 3/4 - \frac{c\theta}{k^\tau} \geq 1/4$, for all k, τ, c. Also, $\lambda_1 - \alpha_{k-1}\theta \leq \lambda_1 = 3/4$, for all k, τ, c. Similarly, we can show $1 - \alpha_{k-1}\theta \in [1/2, 1]$; then, $(1 - \alpha_{k-1}\theta) \dots (1 - \alpha_0 \theta) \geq 0$, $(\lambda_1 - \alpha_{k-1}\theta) \dots (\lambda_1 - \alpha_{t+1}\theta) \geq 0$, and $(\lambda_1 - \alpha_{k-1}) \dots (\lambda_1 - \alpha_{t+1}) \geq 0$, $\forall t$. Thus: $\|x^{\mathrm{I}}(k)\| \geq (1 - \alpha_{k-1}\theta)(1 - \alpha_{k-2}\theta) \dots (1 - \alpha_0 \theta)$. Set $\theta = \theta_k = 1/(s_k(\tau)) \leq 1$, where $s_k(\tau) := \sum_{t=0}^{k-1} (t+1)^{-\tau}$; use $(1 - a_1)(1 - a_2) \dots (1 - a_n) \geq 1 - (a_1 + a_2 + \dots + a_n)$, $a_i \in [0, 1)$, $\forall i$; and $\alpha_k = \frac{c}{(k+1)^\tau}$. We obtain: $\|x^{\mathrm{I}}(k)\| \geq 1 - c\,\theta_k\, s_k(\tau)$, and so: $\theta_k \|x^{\mathrm{I}}(k)\|^2 \geq \frac{(1 - c_{\max})^2}{s_k(\tau)}$, where we denote $c_{\min} := c_0$ and $c_{\max} := 1/(2L) = 1/(2\sqrt{2})$. Further, from (54): $\|x^{\mathrm{II}}(k)\|^2 \geq \alpha_{k-1}^2 \geq \frac{c_{\min}^2}{k^{2\tau}}$, and we obtain:

$$\theta_k \|x^{\mathrm{I}}(k)\|^2 + \|x^{\mathrm{II}}(k)\|^2 \geq \frac{(1 - c_{\max})^2}{s_k(\tau)} + \frac{c_{\min}^2}{k^{2\tau}}. \tag{55}$$

Step 4: Upper Bounding the Optimality Gap From (55): From (55), and using (52)

$$\begin{aligned} \max_{i=1,2} (f(x_i(k)) - f^\star) & \geq \frac{1}{2} \sum_{i=1}^{2} (f(x_i(k)) - f^\star) \\ & \geq \frac{(1 - c_{\max})^2}{2 s_k(\tau)} + \frac{c_{\min}^2}{2\, k^{2\tau}} =: e_k(\tau) \end{aligned} \tag{56}$$

$\forall k \geq 1, \forall \tau \geq 0$. We further upper bound the right hand side in (56) by taking the infimum of $e_k(\tau)$ over $\tau \in [0, \infty)$; we split the interval $[0, \infty)$ into $[0, 3/4]$; $[3/4, 1]$, and $[1, \infty)$, so that

$$\inf_{[0,\infty)} e_k(\tau) \geq \min \left\{ \inf_{[0,3/4]} e_k(\tau),\ \inf_{[3/4,1]} e_k(\tau),\ \inf_{[1,\infty)} e_k(\tau) \right\}. \tag{57}$$

It is easy to prove that: 1) $\inf_{[0,3/4)} e_k(\tau) = \Omega(1/k^{2/3})$; 2) using $s_k(\tau) \leq 3(\log k)(k+1)^{1-\tau}$, $\forall k \geq 3$, $\forall \tau \in [0, 1]$, that $\inf_{[3/4,1]} e_k(\tau) = \Omega\left(\frac{1}{(\log k) k^{1/4}}\right)$; and 3) $\inf_{[1,\infty)} e_k(\tau) = \Omega\left(\frac{1}{\log k}\right)$. (see [18].) Combining the latter bounds with (57) completes the proof of (32).

C. Relaxing Bounded Gradients: Proof of (34) for D–NC

We prove (34) for D–NC while the proof for D–NG is similar and is in [18]. Fix arbitrary $\theta > 0$ and take the f_i's in (35). From (9)–(10), evaluating the ∇f_i's

$$\begin{aligned} x(k) &= (1 - \alpha) W^{\tau_x(k)} y(k-1) + \alpha\, \theta\, W^{\tau_x(k)} (1,-1)^\top \\ y(k) &= W^{\tau_y(k)} \left(x(k) + \beta_{k-1}(x(k) - x(k-1))\right) \end{aligned} \tag{58}$$

for $k = 1, 2, \dots$ We take the initialization at the solution $x(0) = y(0) = (0,0)^\top$. Consider the eigenvalue decomposition $W = Q \Lambda Q^\top$, with $Q = [q_1, q_2]$, $q_1 = \frac{1}{\sqrt{2}}(1,-1)^\top$, $q_2 = \frac{1}{\sqrt{2}}(1,1)^\top$, and Λ is diagonal with $\Lambda_{11} = \lambda_1$, $\Lambda_{22} = \lambda_2 = 1$. Define $z(k) = Q^\top x(k)$ and $w(k) = Q^\top y(k)$. Multiplying (58) from the left by $Q^\top$, and using $Q^\top (1,-1)^\top = (\sqrt{2}, 0)^\top$

$$\begin{aligned} z(k) &= (1 - \alpha) \Lambda^{\tau_x(k)} w(k-1) + \alpha\, \theta\, \Lambda^{\tau_x(k)} (\sqrt{2}, 0)^\top \\ w(k) &= \Lambda^{\tau_y(k)} \left[z(k) + \beta_{k-1}(z(k) - z(k-1))\right] \end{aligned} \tag{59}$$

$k = 1, 2, \dots$, and $z(0) = w(0) = (0,0)^\top$. Next, note that

$$\begin{aligned} \max_{i=1,2} (f(x_i(k)) - f^\star) \geq \frac{1}{2} \sum_{i=1}^{2} (f(x_i(k)) - f^\star) &= \frac{\|x(k)\|^2}{2} \\ = \frac{\|z(k)\|^2}{2} \geq \frac{(z^{(1)}(k))^2}{2}. \end{aligned} \tag{60}$$

Further, from (59) for the first coordinate $z^{(1)}(k), w^{(1)}(k)$, recalling that $\mu := \lambda_1$

$$\begin{aligned} \|z^{(1)}(k)\| &\leq \mu^{\tau_x(k)} \|w^{(1)}(k-1)\| + \sqrt{2}\, \alpha\, \theta\, \mu^{\tau_x(k)} \\ \|w^{(1)}(k)\| &\leq \mu^{\tau_y(k)} \left(2\|z^{(1)}(k)\| + \|z^{(1)}(k-1)\|\right) \end{aligned} \tag{61}$$

$k = 1, 2, \dots$ Note that (61) is analogous to (28)–(29) with the identification $\widetilde{x}(k) \equiv z^{(1)}(k)$, $\widetilde{y}(k) \equiv w^{(1)}(k)$, $\sqrt{N} G \equiv \sqrt{2}\theta$; hence, analogously to the proof of Theorem 7, from (61): $\|w^{(1)}(k-1)\| \leq \frac{2\sqrt{2}\,\alpha\,\theta}{(k-1)^2}$, $k = 2, 3 \dots$ Using the latter, (59), and $\frac{1}{k^2} \geq \mu^{\tau_x(k)} \geq \frac{1}{e\, k^2}$ (see (7)):$\|z^{(1)}(k)\| \geq$

$\alpha\,\theta\sqrt{2}\mu^{\tau_x(k)} - \mu^{\tau_x(k)}\|w^{(1)}(k-1)\| \geq \frac{\alpha\theta\sqrt{2}}{e\,k^2}\left(1 - \frac{2\,e}{(k-1)^2}\right) \geq \frac{\alpha\,\theta\sqrt{2}}{4\,k^2} > 0$, $\forall k \geq 10$. Thus, from (60) and the latter inequality, $\max_{i=1,2}(f(x_i(k)) - f^\star) \geq \frac{\alpha^2\theta^2}{16\,k^4}$, which is, for $\alpha = 1/(2L) = 1/2$, greater or equal M for $\theta = \theta(k, M) = 8\sqrt{M}\,k^2$.

ACKNOWLEDGMENT

The authors wish to thank an anonymous reviewer whose instructive comments led them to develop algorithm D–NC, the anonymous reviewers and the associate editor for several useful suggestions regarding the presentation and organization of the paper, and J. F. C. Mota for pointing them to relevant references and for useful discussions.

REFERENCES

[1] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[2] J. N. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D., Elect. Eng. Comp. Sci., MIT, Cambridge, MA, 1984.

[3] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd Int. Symp. Inform. Process. Sensor Networks (IPSN'04)*, Berkeley, CA, USA, Apr. 2004, pp. 20–27.

[4] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, "On decentralized negotiation of optimal consensus," *Autom.*, vol. 44, no. 4, pp. 1175–1179, 2008.

[5] 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.

[6] I. Necoara and J. A. K. Suykens, "Application of a smoothing technique to decomposition in convex optimization," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2674–2679, Dec. 2008.

[7] 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 Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[8] 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.

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

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

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

[12] C. Lopes and A. H. Sayed, "Adaptive estimation algorithms over distributed networks," in *Proc. 21st IEICE Signal Process. Symp.*, Kyoto, Japan, Nov. 2006.

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

[14] 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.

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

[16] 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.

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

[18] D. Jakovetic, J. Xavier, and J. M. F. Moura, Fast Distributed Gradient Methods [Online]. Available: http://arxiv.org/abs/1112.2972

[19] A. Chen and A. Ozdaglar, "A fast distributed proximal gradient method," in *Proc. 50th Allerton Conf. Commun., Control Comput.*, Monticello, IL, Oct. 2012, pp. 601–608.

[20] A. Chen, "Fast Distributed First-Order Methods," M.S. thesis, Mass. Inst. Technol. (MIT), Cambridge, 2012.

[21] 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.

[22] J. Mota, J. Xavier, P. Aguiar, and M. Pueschel, "Basis pursuit in sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'11)*, Prague, Czech Republic, May 2011, pp. 2916–2919.

[23] J. Mota, J. Xavier, P. Aguiar, and M. Pueschel, "Distributed basis pursuit," *IEEE Trans. Sig. Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[24] U. V. Shanbhag, J. Koshal, and A. Nedic, "Multiuser optimization: distributed algorithms and error analysis," *SIAM J. Control Optim.*, vol. 21, no. 2, pp. 1046–1081, 2011.

[25] H. Terelius, U. Topcu, and R. M. Murray, "Decentralized multi-agent optimization via dual decomposition," in *Proc. 18th World Congr. o Int. Fed. Autom. Control (IFAC)*, Milano, Italy, Aug. 2011, pp. 7391–7397.

[26] E. Ghadimi, I. Shames, and M. Johansson, "Accelerated gradient methods for networked optimization," in *Proc. Amer. Control Conf. (ACC'11)*, San Francisco, CA, Jun. 2011, pp. 1668–1673.

[27] E. Ghadimi, I. Shames, and M. Johansson, Accelerated Gradient Methods for Networked Optimization 2012 [Online]. Available: http://arxiv.org/abs/1211.2132

[28] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Inform. Process. Sensor Networks (IPSN'05)*, Los Angeles, CA, 2005, pp. 63–70.

[29] D. Blatt, A. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *Siam J. Optim.*, vol. 18, no. 1, pp. 29–51, 2009.

[30] P. Tseng, "On accelerated proximal-gradient methods for convex-concave optimization," *SIAM J. Optim*, submitted for publication.

[31] L. Vandenberghe, Optimization Methods for Large-Scale Systems 2010, Lecture notes [Online]. Available: http://www.ee.ucla.edu/~vandenbe/ee236c.html

[32] D. Jakovetic, J. M. F. Moura, and J. Xavier, "Distributed Nesterov-like gradient algorithms," in *Proc. 51st IEEE Conf. Decision Control (CDC'12)*, Dec. 2012, pp. 5459–5464.

[33] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Math. Programm.*, submitted for publication.

[34] G. Shi and K. H. Johansson, "Finite-Time and Asymptotic Convergence of Distributed Averaging and Maximizing Algorithms," Tech. Rep., 2012 [Online]. Available: http://arxiv.org/pdf/1205.1733.pdf

[35] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. 36th Annu. ACM Symp. Theory Comput.*, Chicago, IL, Aug. 2004, pp. 561–568.

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

[37] M. Zargham, A. Ribeiro, and A. Jadbabaie, "A distributed line search for network optimization," in *Proc. Amer. Control Conf.*, Montréal, QC, Canada, Jun. 2012, pp. 472–477.

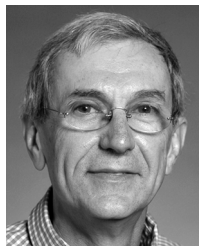
[38] Y. Nesterov, "Gradient Methods for Minimizing Composite Objective Function," Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL),, Tech. Rep. 76, 2007.

Dušan Jakovetić (S'10) received the dipl. ing. diploma from the School of Electrical Engineering, University of Belgrade, Belgrade, Serbia, in 2007, and the Ph.D. degree in electrical and computer engineering from both the Carnegie Mellon University, Pittsburgh, PA, and the Instituto de Sistemas e Robótica (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal, in 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 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 the Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA.

In 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, a large education and research program between CMU and Portugal. He has published over 470 papers, has 12 patents issued by the U.S. Patent Office, and cofounded SpiralGen. His research interests include statistical, algebraic, and distributed signal and image processing, signal processing on graphs, and data science.

Dr. Moura received the IEEE Signal Processing Society Technical Achievement Award and the IEEE Signal Processing Society Society Award. He is a member of the U.S. National Academy of Engineering, corresponding member of the Academy of Sciences of Portugal, and a Fellow of the AAAS. He was IEEE Division IX Director and member of the IEEE Board of Directors (2012–13) and has served on several IEEE Boards. He was President (2008–09) of the *IEEE Signal Processing Society*, 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 *IEEE Proceedings*, *IEEE SP Magazine*, and the *ACM Transactions on Sensor Networks*.