Fast Distributed Gradient Methods

Dušan Jakovetić, João Xavier, and José M. F. Moura

Abstract

We study distributed optimization problems when N nodes minimize the sum \( \sum_{i=1}^{N} f_i(x) \) of their individual costs subject to a common optimization variable \( x \in \mathbb{R}^d \). The sequences \( f_i \)’s 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 and establish their convergence rates in terms of the per-node communications \( K \), the per-node gradient evaluations \( k \), and the network topology. Our first method, Distributed Nesterov Gradient, achieves rate \( \mathcal{O}\left(\frac{1}{(1-\mu(N))^{\frac{3}{2}} \log K} \right) \) and \( \mathcal{O}\left(\frac{1}{(1-\mu(N))^{\frac{3}{2}} k^2} \right) \) (\( \xi > 0 \) arbitrarily small) when the nodes lack knowledge of the global parameters, the Lipschitz constant \( L \) and the spectral gap \( 1 - \mu(N) \). When the nodes know \( L \) and \( \mu(N) \), the rate is \( \mathcal{O}\left(\frac{1}{(1-\mu(N))^{\frac{3}{2}} \log k} \right) \) and \( \mathcal{O}\left(\frac{1}{(1-\mu(N))^{\frac{3}{2}} k^2} \right) \), for the optimized step size. Our second method, Distributed Nesterov gradient with consensus iterations, assumes \( L \) and \( \mu(N) \) known by all. It achieves rate \( \mathcal{O}\left(\frac{1}{(1-\mu(N))^{2} \frac{1}{K^2}} \right) \) and \( \mathcal{O}\left(\frac{1}{k^2} \right) \). While involving only computationally simple iterations, the methods we propose have strictly faster rates than existing distributed (sub)gradient methods, which have rates at most \( \Omega(1/K^{2/3}) \) and \( \Omega(1/k^{2/3}) \). Simulation examples with the logistic and Huber losses demonstrate that our algorithms outperform existing distributed algorithms.

Keywords: Distributed optimization, convergence rate, Nesterov gradient, consensus.

I. INTRODUCTION

Cooperative convex optimization over networks has received much attention recently, motivated by applications in sensor [1], multi-robot [2], or cognitive networks [3], [4], as well as in distributed learning [5]. This paper focuses on the problem where \( N \) nodes (sensors, processors, agents) cooperatively minimize a sum of convex functions \( f(x) := \sum_{i=1}^{N} f_i(x) \) subject to a common optimization 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, sparse, and connected.

To solve this and related problems, the existing literature proposes at least three distributed (sub)gradient type algorithms: a distributed (sub)gradient method in [6] that is further analyzed...
in [7], [8], [9], [10], [11], [12], [13]; distributed dual averaging in [14], analyzed in [1], [15]; and a primal-dual (sub)gradient method in [16].

When the nodes lack global knowledge of the network parameters, reference [14] establishes, for the distributed dual averaging algorithm, rate $O\left(\frac{1}{(1-\mu(N))^{3+\xi} \log k}\right)$, where $k$ is the number of communicated $d$-dimensional vectors per node, which also equals the number of iterations (gradient evaluations per node,) and $1-\mu(N)$ is the spectral gap. Further, when $\mu(N)$ is known to the nodes, and after optimizing the step-size, [14] shows convergence rate of $O\left(\frac{1}{(1-\mu(N))^{1+\xi} \log k}\right)$.

**Contributions.** The class of functions considered in these references 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 with Lipschitz continuous bounded gradients. It is well established in centralized optimization, [17], that one should expect faster convergence rates on classes of more structured functions. For example, 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 (centralized) Nesterov gradient [17]. (Here $k$ is the number of iterations, i.e., the number of gradient evaluations.) In this paper, for the class $\mathcal{F}$ and building from the centralized Nesterov gradient, we develop two distributed gradient methods and prove their convergence rates, in terms of the number of per-node communications $K$, the per-node gradient evaluations $k$, and the network topology. Our first method, the Distributed Nesterov Gradient (D–NG), when the nodes have no global knowledge of $L, G, \mu(N), R$, achieves convergence rate $O\left(\frac{1}{(1-\mu(N))^{3+\xi} \log k}\right)$. Here, $L$ and $G$ are the Lipschitz constant and the gradient bound, $1-\mu(N)$ is the spectral gap, $R$ is the distance to the solution, and $\xi > 0$ is an arbitrarily small quantity. When $L$ and $\mu(N)$ are known to the nodes, the distributed Nesterov gradient with optimized step-size achieves $O\left(\frac{1}{(1-\mu(N))^{1+\xi} \log k}\right)$. Our second method, Distributed Nesterov gradient with Consensus iterations (D–NC), assumes global knowledge or at least upper bounds on $\mu(N)$ and $L$. It achieves convergence rate $O\left(\frac{1}{(1-\mu(N))^{3} \log k/k}\right)$ in the number of communications per node $K$, and $O\left(\frac{1}{k^2}\right)$ in the number of gradient evaluations. Further, we establish that, for the class $\mathcal{F}$, both our methods (achieving at least $O(\log k/k)$) are strictly

---

1We denote by $\mu(N)$ the modulus of the second largest eigenvalue (in modulus) of the underlying weight matrix $W$. Note that $\mu(N)$ depends on $W$, and hence the network topology.
better than the distributed (sub) gradient method [6] and the distributed dual averaging algorithm in [14]. We show analytically that [6] cannot be better than $\Omega \left( \frac{1}{k^{2/3}} \right)$ (see Subsection VII-B for a precise statement), and by simulation that [6] and [14] perform very similarly.

Our results imply that, ignoring “$\xi$-small” and logarithmic factors, the distributed Nesterov gradient requires $\mathcal{K}(N; \epsilon) = O \left( \frac{1}{(1-\mu(N))\epsilon} \right)$ per-node communications to achieve $\epsilon$-accuracy, while the distributed Nesterov gradient with consensus requires $O \left( \frac{1}{(1-\mu(N))\epsilon^{1/2}} \right)$. Interestingly, the latter shows that, $\mathcal{K}(N; \epsilon)$ scales (at most) linearly in the inverse of the spectral gap, i.e., $\mathcal{K}(N; \epsilon) \approx a(\epsilon) \frac{1}{1-\mu(N)}$, with both our algorithms, and the algorithm in [14]. Simulations show very good match with the theoretical linear scaling predictions. The simulations also show that, although the distributed Nesterov gradient with consensus has the best asymptotic slope $a(\epsilon)$ as $\epsilon \to 0$, for practical accuracies $\epsilon$, the distributed Nesterov gradient is the best among all algorithms.

We remark that, in addition to distributed gradient based methods, different types of distributed methods have also been proposed, namely distributed augmented Lagrangian dual or ordinary dual methods [4], [18], [19], [20], [21], [22]. These are based on the augmented Lagrangian (or ordinary) dual of the original problem. These methods in general have significantly more complex iterations than the gradient type methods, due to solving local optimization problems at each node, at each iteration, but may have a lower total communication cost. Finally, we reference [19] that has applied Nesterov gradient method to propose an augmented Lagrangian dual algorithm, but that does not study its convergence rate. In contrast, ours are primal gradient algorithms, with no notion of Lagrangian dual variables, and we prove their convergence rates.

**Paper organization.** The next paragraph introduces notation. Section II describes the network and optimization models. Section III presents our algorithms, distributed Nesterov gradient and distributed Nesterov gradient with consensus iterations, D–NG and D–NC for short. Section IV explains the framework of Nesterov gradient under inexact oracle; 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 algorithms D–NG and D–NC, respectively. Section VII compares our algorithms D–NG and D–NC with existing distributed gradient type methods and discusses the algorithms’ implementation. Section VIII provides simulation examples. Finally, we conclude in Section IX.

**Notation.** We deal with both real and complex scalars, vectors, and matrices, and the notation we define here refers to both, unless explicitly stated otherwise. We denote by $\mathbb{R}^d$ the $d$-dimensional
real coordinate space. 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: \(j\) the imaginary unit \((j^2 = -1)\); \(A_{im}\) or \([A]_{im}\) the entry in the \(l\)-th row and \(m\)-th column of a matrix \(A\); and \(a^{(l)}\) the \(l\)-th entry of vector \(a\); \(A^\top\) the transpose of a real matrix \(A\); \(A^H\) the conjugate transpose of \(A\); \(I, 0, 1,\) and \(e_i\), respectively, the identity matrix, the zero matrix, the column vector with unit entries, and the \(i\)-th column of \(I\); \(J\) the \(N \times N\) ideal consensus matrix \(J := (1/N)11^\top\); \(\oplus\) 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; \(A \succ 0\) means that the Hermitian matrix \(A\) is positive definite; \([a]\) the integer part of a real scalar \(a\); \(\nabla \phi(x)\) and \(\nabla^2 \phi(x)\) the gradient and Hessian at \(x\) of a twice differentiable function \(\phi : \mathbb{R}^d \to \mathbb{R}, d \geq 1\); \(\zeta(s) = \sum_{t=1}^\infty \frac{1}{t^s}\) the Riemann zeta function; \(h_k = \sum_{t=1}^k \frac{1}{t}\) the \(k\)-th harmonic number; and \(C_h = 0.577215\ldots\) the Euler–Mascheroni constant. For two positive sequences \(\eta_n\) and \(\chi_n\), \(\eta_n = O(\chi_n)\) means that \(\lim \sup_{n \to \infty} \frac{\eta_n}{\chi_n} < \infty\); \(\eta_n = \Omega(\chi_n)\) means that \(\lim \inf_{n \to \infty} \frac{\eta_n}{\chi_n} > 0\); and \(\eta_n = \Theta(\chi_n)\) means that \(\eta_n = O(\chi_n)\) and \(\eta_n = \Omega(\chi_n)\).

II. PROBLEM MODEL

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

**Network model.** We consider a (sparse) network \(\mathcal{N}\) of \(N\) nodes (sensors, processors, agents,) each communicating only 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.

**Assumption 1 (Network)** Graph \(\mathcal{G}\) is connected, undirected, and simple (no self/multiple links.)

**Weight matrix.** We associate to graph \(\mathcal{G}\) a symmetric, stochastic (rows 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 \(\tilde{W} = W - J\). We require, with the D–NG algorithm, that:

\[
\mu(N) := \|\tilde{W}\| < 1 \quad \text{and} \quad W \succ 0, \quad (1)
\]

while, with D–NC, we require only the first condition in (1). First condition in (1) is standard; the second condition of \(W\) being positive definite is what we additionally require with respect
to existing work, e.g., [6], [14]. However, we emphasize that, for both conditions in (1) to hold, nodes require no global knowledge about the network parameters, e.g., they do not need to know \( N \). They can fulfill (1) if each node \( i \) knows its own degree \( d_i \) and the degrees of its immediate neighbors. A possible choice is 
\[
W_{ij} = \frac{1}{1 + 3 \max\{d_i, d_j\}} \quad \text{for} \quad \{i,j\} \in E; \\
W_{ij} = 0 \quad \text{for} \quad i \neq j \quad \text{and} \quad \{i,j\} \notin E; \quad \text{and} \quad W_{ii} = 1 - \sum_{j \neq i} W_{ij}.
\]
We order the eigenvalues of \( \tilde{W} \) with \( \lambda_N(\tilde{W}) = \mu(N) \) being largest; we have \( \lambda_i(\tilde{W}) \in (0, 1), \ i > 1, \) and \( \lambda_1(\tilde{W}) = 0 \), with the eigenvector \( q_1 = \frac{1}{\sqrt{N}} \mathbf{1} \). We let \( \tilde{W} = \tilde{Q} \tilde{\Lambda} \tilde{Q}^\top \), where \( \tilde{\Lambda} \) is the diagonal matrix with \( \tilde{\Lambda}_{ii} = \lambda_i(\tilde{W}) \), and \( \tilde{Q} = [q_1, \ldots, q_N] \) is the matrix of the eigenvectors of \( \tilde{W} \).

**Distributed optimization model.** The nodes solve the unconstrained problem:

\[
\minimize \sum_{i=1}^{N} f_i(x) =: f(x). \tag{2}
\]

The function \( f_i : \mathbb{R}^d \to \mathbb{R} \) is known only to node \( i \). We impose the following Assumption.

**Assumption 2 (Solvability and Lipschitz continuity of the derivative)**

(a) Problem (2) is solvable.

(b) For all \( i \), \( f_i \) is convex and has Lipschitz continuous derivative with constant \( L \in [0, \infty) \):

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

We denote by \( x^\star \) a solution to (2) and the optimal value \( f^\star := f(x^\star) \).

**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 the \( f_i \)'s that satisfy Assumptions 2–3 include the logistic and Huber losses (See Section VIII), or the “fair” function 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., [23].

**III. DISTRIBUTED NESTEROV BASED ALGORITHMS**

Subsection III-A presents algorithm D–NG, while subsection III-B presents algorithm D–NC.

**A. 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. Given the initialization \( x_i(0) = y_i(0) \), for all \( i \), the update for
At node $i$ of algorithm D–NG is:

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

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

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 $\alpha_k$ and the sequence $\beta_k$ are:

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

With algorithm (3)–(4), each node $i$, at each iteration $k$: 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 (4).

To avoid notation explosion in the analysis further ahead, we assume throughout 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 $\beta_k$ as proposed in the centralized fast gradient method by Nesterov [17]; see also [24]. With the centralized Nesterov gradient, $\alpha_k = \alpha$ is constant along the iterations. However, algorithm (7)–(8) under a constant step-size 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 $f^*$ (See [25] for details.) We force $f(x_i(k))$ to converge to $f^*$ with (7)–(8) by adopting a diminishing step-size $\alpha_k$, as in (5). The constant $c > 0$ in (5) can be arbitrary (See also ahead Theorem 5 and the remark below.)

**Compact form.** We re-write (3)–(4) in compact form. Let $x(k) = (x_1(k)^\top, x_2(k)^\top, \ldots, x_N(k)^\top)^\top$, and $y(k) = (y_1(k)^\top, y_2(k)^\top, \ldots, y_N(k)^\top)^\top$, and introduce the map $F : \mathbb{R}^{Nd} \to \mathbb{R}^N$ as:

$$F(x) = F(x_1, x_2, \ldots, x_N) = (f_1(x_1)^\top, f_2(x_2)^\top, \ldots, f_N(x_N)^\top)^\top.$$  

Then, given initialization $x(0) = y(0)$, D–NG in compact form is:

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

$$y(k) = x(k) + \beta_{k-1} (x(k) - x(k - 1)), \quad k = 1, 2, \ldots,$$

where $W \otimes I$ is the Kronecker product of $W$ and the $d \times d$ identity.
B. Algorithm D–NC

Algorithm D–NC uses a constant step-size $\alpha$ 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 an auxiliary variable $y_i(k)$ (as with the D–NG);\(^2\) in the inner iterations $s$, nodes perform three rounds of the average-consensus algorithm with the total number of $\tau_k$ inner iterations ($\tau_k$ is specified further ahead.) Briefly recall the average-consensus algorithm. For a fixed $k$, given the initialization $z_i(s = 0, k) \in \mathbb{R}^d$, $i = 1, \ldots, N$, node $i$’s update with average-consensus is:

$$z_i(s, k) = \sum_{j \in O_i} W_{ij} z_j(s - 1, k), \ s = 1, 2, \ldots \tag{9}$$

We now detail D–NC. Given the initialization $x_i(0) = y_i(0) \in \mathbb{R}^d$, at iteration $k$ (given $x_i(k - 1)$ and $y_i(k - 1)$) node $i$ first calculates $\nabla f_i(y_i(k - 1))$. Then, nodes run $\tau_{k-1}$ iterations of (9) with the initialization $z_i(s = 0, k - 1) = \nabla f_i(y_i(k - 1))$, so that each node obtains $g_i(k - 1)$ – an inexact version of $\frac{1}{N} \sum_{i=1}^{N} \nabla f_i(y_i(k - 1))$. Subsequently, node $i$ performs the update:

$$x_i^{\text{aux}}(k) = y_i(k - 1) - \alpha g_i(k - 1). \tag{10}$$

Then, nodes jointly run the second average-consensus (9) with $\tau_{k-1}$ iterations, but now with the initialization $z_i(s = 0, k - 1) = x_i^{\text{aux}}(k)$, to obtain $x_i(k)$–an inexact version of $\frac{1}{N} \sum_{i=1}^{N} x_i^{\text{aux}}(k)$. Subsequently, node $i$ calculates $y_i^{\text{aux}}(k)$ via:

$$y_i^{\text{aux}}(k) = x_i(k) + \beta_{k-1} (x_i(k) - x_i(k - 1)), \tag{11}$$

where $\beta_k$ is in (5). Finally, nodes run the third (last) average-consensus (9) with the initialization $z_i(s = 0, k - 1) = y_i^{\text{aux}}(k)$ and $\tau_{k-1}$ iterations, so that node $i$ obtains $y_i(k)$–an inexact version of $\frac{1}{N} \sum_{i=1}^{N} y_i^{\text{aux}}(k)$. Algorithm D–NC is summarized in Algorithm 1.

**Step-size, initialization, and the number of inner (consensus) iterations $\tau_k$.** We require the step-size $\alpha$ to satisfy $\alpha \leq 1/(2L)$. With D–NC, this condition is critical for convergence. (See also Subsection VIII-A and Figure 1, left.) Also, we let the number of the inner (consensus) iterations $\tau_k$ at the outer iteration $k$ to equal:

$$\tau_k = \lceil \frac{\log 3}{-\log \mu(N)} + \frac{3 \log(k + 1)}{-\log \mu(N)} \rceil. \tag{12}$$

\(^2\)To avoid notation explosion, we use the same letters to denote the iterates of D–NG and D–NC.
the second largest eigenvalue of the weight matrix $\mu_{D-NG}$.

Number of outer iterations $k$.

The total number of (vector) communications per node $i$.

Note that $K_2$, the node's estimate of the solution at a certain stage of the algorithm operation. We are interested in how $K_2$'s optimality gap depends (decreases) with: 1) the number of (outer) iterations $k$; and 2) the total number of (vector) communications per node $K$. Note that, with both algorithms, the number of outer iterations $k$ equals the number of gradient evaluations per node. Further, with $D-NG$, $k = K$, i.e., there is one and only one per-node communication in each iteration $k$. With $D-NC$, however, there are multiple per-node communications in each iteration $k$. Throughout, we refer to $K$ as the number of communication rounds.

**Algorithm 1 Algorithm D–NC**

1: Initialization: Each node $i$ sets: $x_i(0) = y_i(0) \in \mathbb{R}^d$, and $k = 1$.
2: Each node $i$ calculates $\nabla f_i(y_i(k - 1))$.
3: (First consensus) Nodes run (9) for $s = 1, 2, ... \tau_{k-1}$, with $z_i(s = 0, k - 1) = \nabla f_i(y_i(k - 1))$, so that node $i$ obtains $g_i(k - 1) := z_i(s = \tau_{k-1}, k - 1)$.
4: Each node $i$ calculates $x_i^{aux}(k)$ via (10).
5: (Second consensus) Nodes run (9) for $k = 1, 2, ... \tau_{k-1}$, with $z_i(s = 0, k - 1) = x_i^{aux}(k)$ so that node $i$ obtains $x_i(k) := z_i(s = \tau_{k-1}, k - 1)$.
6: Each node $i$ calculates $y_i^{aux}(k)$ via (11).
7: (Third consensus) Nodes run (9) for $k = 1, 2, ... \tau_{k-1}$, with $z_i(s = 0, k - 1) = y_i^{aux}(k)$ so that node $i$ obtains $y_i(k) := z_i(s = \tau_{k-1}, k - 1)$.
8: Set $k \mapsto k + 1$ and go to step 2.

Note that $\tau_k$ increases as a logarithm with $k$; also, it depends on the underlying network (through the second largest eigenvalue of the weight matrix $\mu(N)$.) As with $D-NG$, we assume $x_i(0) = y_i(0) = x_j(0) = y_j(0)$, for all $i, j$.

**Compact form.** We write $D-NC$ in compact form. Use the same compact notation for $x(k)$, $y(k)$, and $\nabla F(y(k))$ as with $D-NG$. Then:

$$x(k) = (W \otimes I)^{\tau_{k-1}} \cdot [y(k - 1) - \alpha(W \otimes I)^{\tau_{k-1}} \nabla F(y(k - 1))]$$

$$y(k) = (W \otimes I)^{\tau_{k-1}} \cdot [x(k) + \beta_{k-1}(x(k) - x(k - 1))]$$

Note that the right matrix power $(W \otimes I)^{\tau_{k-1}}$ in (13) corresponds to the first consensus on the $\nabla f_i(y_i(k - 1))$'s; the left matrix power $(W \otimes I)^{\tau_{k-1}}$ in (13) corresponds to the second consensus on the $x_i^{aux}(k)$'s; and the power $(W \otimes I)^{\tau_{k-1}}$ in (14) corresponds to the third consensus on $y_i^{aux}(k)$'s.

**Performance metrics.** With both algorithms $D-NG$ and $D-NC$, we are interested in estimating the optimality gap in the objective function at each node $i$: $\frac{1}{N}(f(x_i) - f^*)$, where $x_i$ is the node $i$'s estimate of the solution at a certain stage of the algorithm operation. We are interested in how the node $i$'s optimality gap depends (decreases) with: 1) the number of (outer) iterations $k$; and 2) the total number of (vector) communications per node $K$. Note that, with both algorithms, the number of outer iterations $k$ equals the number of gradient evaluations per node. Further, with $D-NG$, $k = K$, i.e., there is one and only one per-node communication in each iteration $k$. With $D-NC$, however, there are multiple per-node communications in each iteration $k$. Throughout, we refer to $K$ as the number of communication rounds.
IV. INTERMEDIATE RESULTS: NESTEROV GRADIENT UNDER INEXACT ORACLE

Subsection IV-A introduces the framework of (centralized) Nesterov gradient under inexact oracle and proves its relation for the progress in one iteration. Subsection IV-B shows how we can cast our algorithms D–NG and D–NC in this framework.

A. Nesterov gradient under inexact oracle

Throughout this subsection, we consider a convex function \( \phi : \mathbb{R}^d \rightarrow \mathbb{R} \) with Lipschitz continuous gradient with constant \( L_\phi \).

Definition 1 (Pointwise inexact first order oracle) Consider a function \( \phi : \mathbb{R}^d \rightarrow \mathbb{R} \) that is convex and has Lipschitz continuous gradient with constant \( L_\phi \). We say that a pair \( (\tilde{\phi}_y, \tilde{g}_y) \in \mathbb{R} \times \mathbb{R}^d \) is a \( (L_y, \delta_y) \) inexact oracle of \( \phi \) at point \( y \) if:

\[
\phi(x) \geq \tilde{\phi}_y + \tilde{g}_y^\top (x - y), \quad \forall x \in \mathbb{R}^d \tag{15}
\]

\[
\phi(x) \leq \tilde{\phi}_y + \tilde{g}_y^\top (x - y) + \frac{L_y}{2} \|x - y\|^2 + \delta_y, \quad \forall x \in \mathbb{R}^d. \tag{16}
\]

Note that the pair \( (\phi(y), \nabla \phi(y)) \) satisfies Definition 1 with \( (L_y = L_\phi, \delta_y = 0) \). Also, if \( (\tilde{\phi}_y, \tilde{g}_y) \) is a \( (L_y, \delta_y) \) inexact oracle at \( y \), then it is also a \( (L'_y, \delta_y) \) local inexact oracle at \( y \), with \( L'_y \geq L_y \).

Remark. The prefix point-wise in Definition 1 emphasizes that the constants \( (L_y, \delta_y) \) are attached to a fixed point \( y \). That is, Definition 1 is concerned with finding \( (\tilde{\phi}_y, \tilde{g}_y) \) that satisfy inequalities (15)–(16) with \( (L_y, \delta_y) \) at a fixed point \( y \). Note the difference from the conventional definition (Definition 1) in [26]. Throughout the paper, we always refer to the inexact first order oracle in the sense of our Definition 1, and hence we drop the prefix point-wise.

Nesterov gradient under inexact oracle. Lemma 2 gives the progress in one iteration of the (centralized) Nesterov gradient under inexact oracle for the unconstrained minimization of \( \phi \). Consider a point \( (\hat{x}(k-1), \hat{y}(k-1)) \in \mathbb{R}^d \times \mathbb{R}^d \), for some fixed \( k = 1, 2, \ldots \). Let \( (\tilde{\phi}_{k-1}, \tilde{g}_{k-1}) \) be a \( (L_{k-1}, \delta_{k-1}) \) inexact oracle of the function \( \phi \) at point \( \hat{y}(k-1) \) and:

\[
\hat{x}(k) = \hat{y}(k-1) - \frac{1}{L_{k-1}} \tilde{g}_{k-1} \tag{17}
\]

\[
\hat{y}(k) = \hat{x}(k) + \beta_{k-1} (\hat{x}(k) - \hat{x}(k-1)). \tag{18}
\]
Lemma 2 (Progress in one iteration under inexact oracle) Consider the update rule (17)–(18) for some $k = 1, 2, \ldots$ Then, for any $x^* \in \mathbb{R}^d$:

\[
(k + 1)^2 \left( \phi(\hat{x}(k)) - \phi(x^*) \right) + 2L_{k-1} \|\hat{v}(k) - x^*\|^2 \leq (k^2 - 1) \left( \phi(\hat{x}(k-1)) - \phi(x^*) \right) + 2L_{k-1} \|\hat{v}(k-1) - x^*\|^2 + (k + 1)^2 \delta_{k-1},
\]

where $\theta_k = 2/(k + 2)$ and

\[
\hat{v}(k) = \frac{\tilde{y}(k) - (1 - \theta_k)\hat{x}(k)}{\theta_k}.
\]

**Proof:** **Step 1.** We first prove the following two auxiliary equalities:

\[
\begin{align*}
\hat{v}(k) &= \hat{v}(k-1) - \frac{1}{\theta_{k-1}L_{k-1}} \tilde{g}_{k-1} = \hat{v}(k-1) + \frac{1}{\theta_{k-1}}(\hat{x}(k) - \tilde{y}(k-1)) \quad (21) \\
\theta_{k-1}\hat{v}(k) &= \hat{x}(k) - (1 - \theta_{k-1})\hat{x}(k-1). \quad (22)
\end{align*}
\]

We prove (21). Using $\theta_k = \frac{2}{k+2}$, $1 - \theta_k = \frac{k}{k+2}$, the definition of $\hat{v}(k)$ in (20), and (17):

\[
\hat{v}(k) = \frac{1}{\theta_k} (\tilde{y}(k) - (1 - \theta_k)\hat{x}(k)) = \frac{1}{\theta_k} \left( \hat{x}(k) + \frac{k - 1}{k + 2} (\hat{x}(k) - \hat{x}(k-1)) - \frac{k}{k + 2} \hat{x}(k) \right) = \frac{k + 2}{2} \left( \frac{k - 1}{k + 2} \hat{x}(k) - \frac{k - 1}{k + 2} \hat{x}(k-1) \right) = \frac{k + 1}{2} \hat{x}(k) - \frac{k - 1}{2} \hat{x}(k-1).
\]

By rewriting the last equation, while using again the update rule for $\hat{x}(k)$ in (17), and equalities $\frac{k+1}{2} = \frac{1}{\theta_{k-1}}$ and $\frac{k-1}{2} = \frac{1 - \theta_{k-1}}{\theta_{k-1}}$:

\[
\hat{v}(k) = \frac{1}{\theta_{k-1}} (\tilde{y}(k-1) - \frac{1}{L_{k-1}} \tilde{g}_{k-1}) - \frac{1 - \theta_{k-1}}{\theta_{k-1}} \hat{x}(k-1) = \frac{\tilde{y}(k-1) - (1 - \theta_{k-1})\hat{x}(k-1)}{\theta_{k-1}} - \frac{1}{\theta_{k-1}L_{k-1}} \tilde{g}_{k-1} = \hat{v}(k-1) - \frac{1}{\theta_{k-1}L_{k-1}} \tilde{g}_{k-1}.
\]

In the last equality, we used again the definition of $\hat{v}(k-1)$ derived from (20).

We now prove (22). Multiplying (21) by $\theta_{k-1}$ and using (20):

\[
\theta_{k-1}\hat{v}(k) = \theta_{k-1}\hat{v}(k-1) + (\hat{x}(k) - \tilde{y}(k-1)) = \tilde{y}(k-1) - (1 - \theta_{k-1})\hat{x}(k-1) + (\hat{x}(k) - \tilde{y}(k-1)) = \hat{x}(k) - (1 - \theta_{k-1})\hat{x}(k-1).
\]
Step 2. We prove the following relation:

\[
\phi(\tilde{x}(k)) \leq \phi(z) + L_{k-1}(\tilde{x}(k) - \tilde{y}(k-1))^\top (z - \tilde{x}(k)) + \frac{L_{k-1}}{2} \|\tilde{x}(k) - \tilde{y}(k-1)\|^2 + \delta_{k-1}, \; \forall z \in \mathbb{R}^d.
\]

Using the inexact oracle property (16):

\[
\phi(\tilde{x}(k)) \leq \tilde{\phi}_{k-1} + \tilde{\gamma}_{k-1}^\top (\tilde{x}(k) - \tilde{y}(k-1)) + \frac{L_{k-1}}{2} \|\tilde{x}(k) - \tilde{y}(k-1)\|^2 + \delta_{k-1}.
\]

Further, \(\forall z \in \mathbb{R}^d: 0 = 0^\top (z - \tilde{x}(k)) = (\tilde{x}(k) - \tilde{y}(k-1) + \frac{1}{L_{k-1}} \tilde{\gamma}_{k-1})^\top (z - \tilde{x}(k)), \) and so:

\[
\tilde{\gamma}_{k-1}^\top (z - \tilde{x}(k)) + L_{k-1}(\tilde{x}(k) - \tilde{y}(k-1))^\top (z - \tilde{x}(k)) = 0.
\]

From property (15): \(\phi(z) \geq \tilde{\phi}_{k-1} + \tilde{\gamma}_{k-1}^\top (z - \tilde{y}(k-1)), \) and so, using the last equation and adding (24) and (25), the claim (23) follows.

Step 3. We finalize the proof of Lemma 2 by proving (19). We start by using relation (23). Namely: 1) setting \(z = \tilde{x}(k-1)\) in (23) and multiplying inequality (23) by \(1 - \theta_{k-1}\); 2) setting \(z = x^*\) in (23) and multiplying inequality (23) by \(\theta_{k-1}\); and 3) adding the corresponding two inequalities:

\[
\theta_{k-1} \{\phi(\tilde{x}(k)) - \phi(x^*)\} + (1 - \theta_{k-1}) \{\phi(\tilde{x}(k)) - \phi(\tilde{x}(k-1))\}
\]

\[= \{\phi(\tilde{x}(k)) - \phi(x^*)\} - (1 - \theta_{k-1}) \{\phi(\tilde{x}(k-1)) - \phi(x^*)\}
\]

\[\leq \theta_{k-1} L_{k-1} (\tilde{x}(k) - \tilde{y}(k-1))^\top (x^* - \tilde{x}(k)) + (1 - \theta_{k-1}) L_{k-1} (\tilde{x}(k) - \tilde{y}(k-1))^\top (\tilde{x}(k-1) - \tilde{x}(k))
\]

\[+ \frac{L_{k-1}}{2} \|\tilde{x}(k) - \tilde{y}(k-1)\|^2 + \delta_{k-1}
\]

\[= L_{k-1} (\tilde{x}(k) - \tilde{y}(k-1))^\top (\theta_{k-1} x^* + (1 - \theta_{k-1}) \tilde{x}(k-1) - \tilde{x}(k)) + \frac{L_{k-1}}{2} \|\tilde{x}(k) - \tilde{y}(k-1)\|^2 + \delta_{k-1}
\]

\[= \frac{L_{k-1}}{2} (2(\tilde{x}(k) - \tilde{y}(k-1))^\top (\theta_{k-1} x^* + (1 - \theta_{k-1}) \tilde{x}(k-1) - \tilde{x}(k)) + \|\tilde{x}(k) - \tilde{y}(k-1)\|^2) + \delta_{k-1}.
\]

Denote by:

\[\mathcal{M}_{k-1} = (2(\tilde{x}(k) - \tilde{y}(k-1))^\top (\theta_{k-1} x^* + (1 - \theta_{k-1}) \tilde{x}(k-1) - \tilde{x}(k)) + \|\tilde{x}(k) - \tilde{y}(k-1)\|^2).
\]

Then, inequality (26) is written simply as:

\[
\{\phi(\tilde{x}(k)) - \phi(x^*)\} - (1 - \theta_{k-1}) \{\phi(\tilde{x}(k-1)) - \phi(x^*)\} \leq \frac{L_{k-1}}{2} \mathcal{M}_{k-1} + \delta_{k-1}.
\]
Now, we simplify the expression for $\mathcal{M}_{k-1}$ as follows. Using the identity:

$$\|\hat{x}(k) - \hat{y}(k-1)\|^2 = 2(\hat{x}(k) - \hat{y}(k-1))^\top \hat{x}(k) + \|\hat{y}(k-1)\|^2 - \|\hat{x}(k)\|^2,$$

we have:

$$\mathcal{M}_{k-1} = 2(\hat{x}(k) - \hat{y}(k-1))^\top (\theta_{k-1} x^* + (1 - \theta_{k-1}) \hat{x}(k-1)) - \|\hat{x}(k)\|^2 + \|\hat{y}(k-1)\|^2$$

$$= \|\hat{y}(k-1) - ((1 - \theta_{k-1}) \hat{x}(k-1) + \theta_{k-1} x^*)\|^2 - \|\hat{x}(k) - ((1 - \theta_{k-1}) \hat{x}(k-1) + \theta_{k-1} x^*)\|^2$$

$$= \theta_{k-1}^2 \|\hat{v}(k-1) - x^*\|^2 - \theta_{k-1}^2 \|\hat{v}(k) - x^*\|^2,$$

(28)

where the last equality follows by the definition of $\hat{v}(k-1)$ in (20) and by the identity (22).

Now, combining (27) and (28):

$$\left(\phi(\hat{x}(k)) - \phi(x^*)\right) - (1 - \theta_{k-1})(\phi(\hat{x}(k-1)) - \phi(x^*))$$

$$\leq \frac{L_{k-1} \theta_{k-1}^2}{2} \left(\|\hat{v}(k-1) - x^*\|^2 - \|\hat{v}(k) - x^*\|^2\right) + \delta_{k-1}.$$

Finally, multiplying the last equation by $\frac{4}{\theta_{k-1}}$, and using $\theta_{k-1} = 2/(k+1)$, we get the result.

**B. 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.** Denote by $\overline{y}(k) := \frac{1}{N} \sum_{i=1}^{N} y_i(k)$ and $\overline{x}(k) := \frac{1}{N} \sum_{i=1}^{N} x_i(k)$ the global averages of the individual nodes’ iterates $y_i(k)$’s and $x_i(k)$’s, respectively. Then, by multiplying (7)–(8) from the left by $(1/N)(1^\top \otimes I)$, and using $(1^\top \otimes I)(W \otimes I) = 1^\top \otimes I$:

$$\overline{x}(k) = \overline{y}(k - 1) - \frac{\alpha_k}{N} \sum_{i=1}^{N} \nabla f_i(y_i(k - 1)),$$

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

(29)

for $k = 1, 2, \ldots$, and $\overline{x}(0) = \overline{y}(0)$. Thus, with D–NG, $(\overline{x}(k), \overline{y}(k))$ evolves (almost) according to the centralized Nesterov gradient method with step-size $\frac{\alpha_k}{N}$ to minimize $f := \sum_{i=1}^{N} f_i$, except that the gradient $\nabla f(\overline{y}(k-1)) = \sum_{i=1}^{N} \nabla f_i(\overline{y}(k-1))$ is replaced by $\sum_{i=1}^{N} \nabla f_i(y_i(k-1))$.

Denote by:

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

(30)
Then, algorithm (29) is, defining $L'_{k-1} = \frac{N}{\alpha_{k-1}}$:

$$
\bar{x}(k) = \bar{y}(k-1) - \frac{1}{L'_{k-1}} \bar{g}_{k-1}, \quad \bar{y}(k) = \bar{x}(k) + \beta_{k-1} (\bar{x}(k) - \bar{x}(k-1)), \quad (31)
$$

for $k = 1, 2, \ldots$. The following Lemma puts algorithm D–NG in the inexact oracle framework.

**Lemma 3** Let Assumption 2 hold. Then, $(\bar{f}_k, \bar{g}_k)$ in (30) is a $(L_k, \delta_k)$ inexact oracle of $f = \sum_{i=1}^N f_i$ at point $\bar{y}(k)$ with constants $L_k = 2NL$ and $\delta_k = L \sum_{i=1}^N \|\bar{y}(k) - y_i(k)\|^2$.

**Remark.** If $\alpha_{k-1} \leq \frac{1}{2L_k}$, then $L'_{k-1} = \frac{N}{\alpha_{k-1}} = 2NLk \geq L_{k-1} = 2NL$, and so the iteration progress shown in Lemma 2 applies to $\bar{x}(k), \bar{y}(k)$ with D–NG, by setting $\phi \equiv f$, $\bar{x}(k) \equiv \bar{x}(k)$, $\bar{y}(k) \equiv \bar{y}(k)$, and $\bar{x}^* \equiv \bar{x}^*$. Hence, we can establish the optimality gap $f(\bar{x}(k)) - f^*$ by using Lemma 2 and, additionally, by finding $\delta_k$, i.e., $\|\bar{y}(k) - y_i(k)\|$. This is detailed in Section V.

**Proof of Lemma 3:** For notation simplicity, we re-write $y(k)$ and $\bar{y}(k)$ as $y$ and $\bar{y}$, and $\bar{f}_k, \bar{g}_k, L_k, \delta_k$ as $\bar{f}_y, \bar{g}_y, L_y, \delta_y$. In view of Definition 1, we need to show inequalities (15) and (16). We first show (15). 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 $x - y_i = x - \bar{y} + \bar{y} - y_i$:

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

We now prove (16). 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 $x - y_i = (x - \bar{y}) + (\bar{y} - y_i)$, and using the inequality $\|x - y_i\|^2 = \|(x - \bar{y}) + (\bar{y} - y_i)\|^2 \leq 2\|x - \bar{y}\|^2 + 2\|\bar{y} - y_i\|^2$, gives:

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

**Algorithm D–NC.** Denote (the same as with D–NG to avoid notational clutter) $\bar{x}(k) := \frac{1}{N} \sum_{i=1}^N x_i(k)$, and $\bar{y}(k) := \frac{1}{N} \sum_{i=1}^N y_i(k)$. Multiplying (13)–(14) from the left by $(1/N) 1 \top \otimes I$, and using $(1 \top \otimes I)(W \otimes I) = 1 \top \otimes I$, we get that $(\bar{x}(k), \bar{y}(k))$ satisfy (29), with $\alpha_{k-1}$ replaced by the constant $\alpha$. Hence, with both D–NG and D–NC, the global averages $\bar{x}(k)$ and $\bar{y}(k)$ follow

October 2, 2012

DRAFT
the same rule (29), except for different step-sizes. What is also different, however, are the values of disagreements $\|\vec{y}(k) - y_i(k)\|$. Define $\tilde{f}_k, \tilde{g}_k$ for D–NC as in (30), and consider (31) and Lemma 3. Because $\alpha \leq 1/(2L)$, we have $L'_{k-1} = \frac{N}{\alpha} \geq 2NL$, and so, Lemma 2 applies to $(\vec{\pi}(k), \vec{y}(k))$ of D–NC by setting $\phi \equiv f, \vec{x}(k) \equiv \vec{\pi}(k), \vec{g}(k) \equiv \vec{y}(k),$ and $x^* \equiv x^*$. Hence, as with D–NG, convergence analysis of D–NC boils down to finding the disagreement $\|\vec{y}(k) - y_i(k)\|$ and then applying Lemma 2. This is considered in Section VI.

V. ALGORITHM D–NG: CONVERGENCE ANALYSIS

Subsection V-A bounds the disagreement $\|\vec{y}(k) - y_i(k)\|$ with D–NG. Subsection V-B combines this bound with Lemma 2 to derive the convergence rate of D–NG; this subsection also studies how the convergence rate depends on the underlying network.

A. Algorithm D–NG: Disagreement estimate

In this subsection, to avoid notational clutter, we let $d = 1$, but all the results hold for a generic $d$. We consider algorithm D–NG and derive a generic bound on the differences between the estimates $x_i(k)$ at different nodes. More precisely, we derive an upper bound on $\|\vec{x}(k)\|$, where $\vec{x}(k) := x(k) - \vec{\pi}(k) = (I - J)x(k)$. Denote also by $\vec{y}(k) := y(k) - \vec{y}(k)1 = (I - J)y(k)$, and recall $\vec{W} := W - J, \mu(N) = \|\vec{W}\|$. 

Lemma 4 (Consensus estimate) Consider algorithm D–NG with step-size $\alpha_k = c/(k + 1)$ under Assumptions 1 and 3. Then, for $k = 1, 2, \ldots$:

$$\|\vec{x}(k)\| \leq \sqrt{N} c G C_{\text{cons}} \frac{1}{k} \quad \text{and} \quad \|\vec{y}(k)\| \leq 4 \sqrt{N} c G C_{\text{cons}} \frac{1}{k},$$

where $C_{\text{cons}}$ is a constant that depends on the matrix $\vec{W}$ and equals:

$$C_{\text{cons}} = \min \left\{ \frac{8}{\sqrt{\lambda_2(\vec{W})} (1 - \lambda_2(\vec{W})), \sqrt{\mu(N)} (1 - \mu(N))} \right\}$$

$$\times \left\{ 2 B \left( \sqrt{\mu(N)} \right) + \frac{2(2C_h + 1)}{e} \frac{1}{(-\log \mu(N))} + \frac{2}{1 - \sqrt{\mu(N)}} \right\},$$

where $B : (0, 1) \to \mathbb{R}, B(r) = \sup_{z \geq 1/2} (z r^z \log(1 + z))$.

We note that $B(r)$ is finite whenever $r \in (0, 1)$, which is the case here.
Proof: We outline the main steps in the proof. First, we model the dynamics of \((\tilde{x}(k)^T, \tilde{x}(k-1)^T)^T\) as a linear time varying system with \((I - J)\nabla F(y(k))\) being the system inputs and determine its solution. Second, we calculate the underlying time varying system matrices. Third, we upper bound the norms of the time varying system matrices. Finally, in the fourth step, we use these bounds and a summation argument to complete the proof of the Lemma.

1) Recursion for \((\tilde{x}(k)^T, \tilde{x}(k-1)^T)^T\): Consider (7)–(8) with \(d = 1\) (Here \(W \otimes I = W\)) Substituting the expression for \(y(k-1)\) in (7); multiplying the resulting equation from the left by \((I - J)\), and using \((I - J)W = \tilde{W} = \tilde{W}(I - J)\), obtain:

\[
\begin{align*}
\tilde{x}(k) &= (1 + \beta_{k-2})\tilde{W}\tilde{x}(k-1) - \beta_{k-2}\tilde{W}\tilde{x}(k-2) - \alpha_{k-1}(I - J)\nabla F(y(k-1)), \ k = 1, ... \\
\tilde{x}(0) &= (I - J)x(0), \ \tilde{x}(-1) = 0,
\end{align*}
\]

and \(\beta_k\), for \(k = 0, 1, ...\), is in (5), and \(\beta_{-1} = 0\). Recall that we assumed \(x_i(0) = x_j(0)\) for all \(i, j\), so that \(\tilde{x}(0) = 0\). Next, the recursion for the \(2N \times 1\) augmented state \((\tilde{x}(k)^T, \tilde{x}(k-1)^T)^T\) becomes, for \(k = 1, 2, ...\):

\[
\begin{bmatrix}
\tilde{x}(k) \\
\tilde{x}(k-1)
\end{bmatrix} =
\begin{bmatrix}
(1 + \beta_{k-2})\tilde{W} & -\beta_{k-2}\tilde{W} \\
I & 0
\end{bmatrix}
\begin{bmatrix}
\tilde{x}(k-1) \\
\tilde{x}(k-2)
\end{bmatrix} - \alpha_{k-1}
\begin{bmatrix}
(I - J)\nabla F(y(k-1)) \\
0
\end{bmatrix},
\]

(33)

with \((\tilde{x}(0)^T, \tilde{x}(-1)^T)^T = 0\). Define the \(2N \times 2N\) system matrices:

\[
\Phi(k, t) := \Pi_{s=2}^{k-t+1}
\begin{bmatrix}
(1 + \beta_{k-s})\tilde{W} & -\beta_{k-s}\tilde{W} \\
I & 0
\end{bmatrix}, \ k > t,
\]

(34)

and \(\Phi(k, k) = I\). Then, the solution to (33) is:

\[
\begin{bmatrix}
\tilde{x}(k) \\
\tilde{x}(k-1)
\end{bmatrix} = \sum_{t=0}^{k-1} \Phi(k, t + 1) \alpha_t
\begin{bmatrix}
-(I - J)\nabla F(y(t)) \\
0
\end{bmatrix}, \ k = 1, 2, ...
\]

(35)

2) Calculating \(\Phi(k, t)\): We now show the interesting structure of the matrix \(\Phi(k, t)\) in (34) by decomposing it into the product of an orthonormal matrix \(U\), a block-diagonal matrix, and \(U^T\). While \(U\) is independent of \(k, t\), the block diagonal matrix depends on \(k, t\), and has \(2 \times 2\) diagonal blocks. Consider the matrix in (33) with \(k - 2 = t\), for a generic \(t = -1, 0, 1, ...\) Using
the eigenvalue decomposition $\widetilde{W} = Q\Lambda Q^T$:

$$
\begin{bmatrix}
(1 + \beta_i)\widetilde{W} & -\beta_i\widetilde{W} \\
I & 0
\end{bmatrix} = (Q \oplus Q) \begin{bmatrix}
(1 + \beta_i)\Lambda & -\beta_i\Lambda \\
I & 0
\end{bmatrix} (Q \oplus Q)^T
$$

$$
= (Q \oplus Q) \left( \oplus_{i=1}^N \Sigma_i(t) \right) P^T (Q \oplus Q)^T, \quad (36)
$$

where $P$ is the $2N \times 2N$ permutation matrix ($e_i$ here is the $i$–th column of the $2N \times 2N$ identity matrix) $P = [e_1, e_{N+1}, e_2, e_{N+2}, \ldots, e_N, e_{2N}]^T$, and $\Sigma_i(t)$ is a $2 \times 2$ matrix:

$$
\Sigma_i(t) = \begin{bmatrix}
(1 + \beta_i)\lambda_i(\widetilde{W}) & -\beta_i\lambda_i(\widetilde{W}) \\
1 & 0
\end{bmatrix}. \quad (37)
$$

Using (36), and the fact that $(Q \oplus Q)P$ is orthonormal: $((Q \oplus Q)P) \cdot ((Q \oplus Q)P)^T = (Q \oplus Q)PP^T(Q \oplus Q)^T = (QQ^T) \oplus (QQ^T) = I$, we can express $\Phi(k, t)$ in (34) as:

$$
\Phi(k, t) := (Q \oplus Q)P \left( \oplus_{i=1}^N \Pi_{s=2}^{k-t+1} \Sigma_i(k - s) \right) P^T (Q \oplus Q)^T, \quad \text{for } k > t; \quad \Phi(k, k) = I. \quad (38)
$$

3) Bounding the norm of $\Phi(k, t)$: We next upper bound $\|\Phi(k, t)\|$. 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 $2 \times 2$ blocks $\Pi_{s=2}^{k-t+1} \Sigma_i(k - s)$), and so:

$$
\|\Phi(k, t)\| = \max_{i = 1, \ldots, N} \|\Pi_{s=2}^{k-t+1} \Sigma_i(k - s)\|.
$$

We proceed by calculating $\|\Pi_{s=2}^{k-t+1} \Sigma_i(k - s)\|$. We distinguish two cases: $i = 1$, and $i > 1$.

Case $i = 1$. As $\lambda_1(\widetilde{W}) = 0$, we have that, for all $t$, $[\Sigma_1]_{21} = [\Sigma_1(t)]_{21} = 1$, and the entries $(1, 1), (1, 2)$ and $(2, 2)$ of $\Sigma_1$ 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:

$$
\|\Pi_{s=2}^{k-t+1} \Sigma_i(k - s)\| = \begin{cases}
1 & \text{if } k = t + 1 \\
0 & \text{if } k > t + 1.
\end{cases} \quad (39)
$$

Case $i > 1$. To simplify notation, let $\lambda_i := \lambda_i(\widetilde{W})$, and recall $\lambda_i \in (0, 1); \Sigma_i(t)$ is:

$$
\Sigma_i(t) = \hat{\Sigma}_i - \frac{3}{t + 3} \Delta_i, \quad \hat{\Sigma}_i = \begin{bmatrix}
2\lambda_i & -\lambda_i \\
1 & 0
\end{bmatrix}, \quad \Delta_i = \begin{bmatrix}
\lambda_i & -\lambda_i \\
0 & 0
\end{bmatrix}.
$$
The matrix $\widehat{\Sigma}_i$ is diagonalizable, with $\widehat{\Sigma}_i = \widehat{Q}_i \widehat{D}_i \widehat{Q}_i^{-1}$, and:

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

Also, $\widehat{Q}_i^{-1} = \frac{1}{2i \sqrt{\lambda_i(1 - \lambda_i)}} \begin{bmatrix} 1 - \lambda_i + j \sqrt{\lambda_i(1 - \lambda_i)} \\ -1 + \lambda_i + j \sqrt{\lambda_i(1 - \lambda_i)} \end{bmatrix}$. (Note that the matrices $\widehat{Q}_i$ and $\widehat{D}_i$ are complex.) Denote by $\mathcal{D}_i(t) = \widehat{D}_i - \frac{3}{t+3} \widehat{Q}_i^{-1} \Delta_i \widehat{Q}_i$. Then, $\Sigma_i(t) = \widehat{Q}_i \left( \widehat{D}_i - \frac{3}{t+3} \widehat{Q}_i^{-1} \Delta_i \widehat{Q}_i \right) \widehat{Q}_i^{-1} = \widehat{Q}_i \mathcal{D}_i(t) \widehat{Q}_i^{-1}$, and so, by the sub-multiplicative property of norms:

$$\|\Pi_{s=2}^{k-t+1} \Sigma_i(k - s)\| \leq \left\| \widehat{Q}_i \right\| \left\| \widehat{Q}_i^{-1} \right\| \Pi_{s=2}^{k-t+1} \|\mathcal{D}_i(k - s)\|. \quad (40)$$

Further, the norms of $\widehat{Q}_i$ and $\widehat{Q}_i^{-1}$ are upper bounded as:

$$\left\| \widehat{Q}_i \right\| \leq \sqrt{2}, \quad \left\| \widehat{Q}_i^{-1} \right\| \leq \sqrt{2} \left\| \widehat{Q}_i^{-1} \right\|_{\infty} = \frac{2\sqrt{2}}{\sqrt{\lambda_i(1 - \lambda_i)}}. \quad (41)$$

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

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

Denote by $a_t = \frac{3}{t+3}$, $t = 0, 1, \ldots$, and $a_{-1} = 1$. After some algebra:

$$\mathcal{D}_i(t) = \frac{1}{2} \begin{bmatrix}
(2 - a_t)(\lambda_i + j \sqrt{\lambda_i(1 - \lambda_i)}) & a_t(\lambda_i - j \sqrt{\lambda_i(1 - \lambda_i)}) \\
a_t(\lambda_i + j \sqrt{\lambda_i(1 - \lambda_i)}) & (2 - a_t)(\lambda_i - j \sqrt{\lambda_i(1 - \lambda_i)})
\end{bmatrix}, \quad (42)$$

and:

$$\mathcal{D}_i^H(t) \mathcal{D}_i(t) = \begin{bmatrix}
\frac{a_t^2 + (2 - a_t)^2}{4} \lambda_i & \frac{a_t(2 - a_t)}{2} \left( 2 \lambda_i^2 - \lambda_i - 2 j \lambda_i \sqrt{\lambda_i(1 - \lambda_i)} \right) \\
\frac{a_t(2 - a_t)}{2} \left( 2 \lambda_i^2 - \lambda_i - 2 j \lambda_i \sqrt{\lambda_i(1 - \lambda_i)} \right) & \frac{a_t^2 + (2 - a_t)^2}{4} \lambda_i
\end{bmatrix}. \quad (43)$$

Next, very interestingly: $\|\mathcal{D}_i^H(t) \mathcal{D}_i(t)\|_1 = \|[\mathcal{D}_i^H(t) \mathcal{D}_i(t)]_{11}\| + \|[\mathcal{D}_i^H(t) \mathcal{D}_i(t)]_{12}\| = \frac{1}{4}(a_t^2 + (2 - a_t)^2) \lambda_i + \frac{1}{2} a_t (2 - a_t) \lambda_i = \lambda_i$, 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 Hermitian 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, (41) and (42) to (40), we get, for $i \neq 1$:

$$\|\Pi_{s=2}^{k-t+1} \Sigma_i(k - s)\| \leq \left\| \widehat{Q}_i \right\| \left\| \widehat{Q}_i^{-1} \right\| \left( \sqrt{\lambda_i} \right)^{k-t} = \frac{8}{\sqrt{\lambda_i(1 - \lambda_i)}} \left( \sqrt{\lambda_i} \right)^{k-t}, \quad k \geq t + 1. \quad (43)$$
Finally, using $\|\Phi(k,t)\| = \max_{i=1,...,N} \|\Pi_{s=2}^{k-t+1} \Sigma_i (k-s)\|$, and (39), (43):

$$\|\Phi(k,t)\| \leq \frac{8}{\min_{i \in \{2,N\}} \sqrt{\lambda_i(\tilde{W})(1 - \lambda_i(\tilde{W}))}} \left( \sqrt{\nu(N)} \right)^{k-t}, \quad k \geq t. \quad (44)$$

4) Summation: We apply (44) to (35). Using the sub-multiplicative and sub-additive properties of norms, expression $\alpha_t = c/(t+1)$, and the inequalities $\|\tilde{x}(k)\| \leq \|(\tilde{x}(k)^T, \tilde{x}(k-1)^T)^T\|$, $\|(-I - J) \nabla F(y(t))^T, 0^T)\| \leq \sqrt{NG}$:

$$\|\tilde{x}(k)\| \leq \frac{8 \sqrt{N} c G}{\min_{i \in \{2,N\}} \sqrt{\lambda_i(\tilde{W})(1 - \lambda_i(\tilde{W}))}} \sum_{t=0}^{k-1} \left( \sqrt{\nu(N)} \right)^{k-(t+1)} \frac{1}{(t+1)}. \quad (45)$$

We now denote by $r := \sqrt{\nu(N)} \in (0,1)$. 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 $k$:

$$\sum_{t=0}^{k-1} r^{k-(t+1)} \frac{1}{(t+1)} = \left( r^{k-1} + r^{k-2} \frac{1}{2} + ... + r^{\lfloor k/2 \rfloor} \frac{1}{\lfloor k/2 \rfloor} \right) + \left( r^{\lfloor k/2 \rfloor + 1} \frac{1}{\lfloor k/2 \rfloor + 1} + ... + \frac{1}{k} \right)$$

$$\leq r^{k/2} \left( 1 + \frac{1}{2} + ... + \frac{1}{(k+1)/2} \right) + \frac{1}{(k/2)} (1 + r + ... + r^k).$$

Using the following inequality for the harmonic number $h_t = 1 + \frac{1}{2} + ... + \frac{1}{t}$, $t = 1,2,...$

$$h_t \leq \log t + C_h + \frac{1}{2}, \quad t = 1,2,...$$

we have:

$$\sum_{t=0}^{k-1} r^{k-(t+1)} \frac{1}{(t+1)} \leq r^{k/2} \left( \log(1 + k/2) + C_h + 1/2 \right) + \frac{2}{k} \frac{1}{1-r}$$

$$= 2 \sup_{z \geq 1/2} \left\{ r^{k/2} \log(1 + k/2) \frac{1}{k} + 2(C_h + 1/2) r^{k/2} \frac{1}{k} \right\} \frac{1}{k} + \frac{2}{k} \frac{1}{1-r}$$

$$\leq 2 \sup_{z \geq 1/2} \left\{ r^z \log(1 + z) \frac{1}{k} + 2(C_h + 1/2) \sup_{z \geq 1/2} \{ r^z \frac{1}{k} \} \right\} \frac{1}{k} + \frac{2}{k} \frac{1}{1-r}$$

$$\leq \left( 2B(r) + 2(C_h + 1/2) \frac{1}{e(-\log r)} + \frac{2}{1-r} \right) \frac{1}{k}.$$
B. Convergence rate and network scaling

We now state the convergence rate result for algorithm D–NG.

**Theorem 5** Consider algorithm D–NG under Assumptions 1–3, with step-size $\alpha_k = \frac{c}{(k+1)}$, and $c \leq 1/(2L)$. Let $\|\hat{\theta}(0) - x^*\| \leq R$, $R > 0$. Then, for all $i$, for all $k = 1, 2, \ldots$:

$$\frac{1}{N} (f(x_i(k)) - f^*) \leq \frac{2 R^2}{c} \left( \frac{1}{k} \right) + 16 c^2 L C_{cons}^2 G^2 \left( \frac{1}{k} \sum_{t=1}^{k-1} \frac{(t+2)^2}{(t+1)t^2} \right) + c G^2 C_{cons} \left( \frac{1}{k} \right) \quad (46)$$

Then

$$\leq C(N) \left( \frac{1}{k} \sum_{t=1}^{k} \frac{(t+2)^2}{(t+1)t^2} \right), \quad C(N) = \frac{2 R^2}{c} + 16 c^2 L C_{cons}^2 G^2 + c G^2 C_{cons}.$$  

**Remark.** Theorem 5 extends to the case $c > 1/(2L)$; in that case, $\frac{1}{N} (f(x_i(k)) - f^*) \leq C'(N) \left( \frac{1}{k} \sum_{t=1}^{k} \frac{(t+2)^2}{(t+1)t^2} \right)$, where $C'(N) = C''(L, G, R, c) + 16c^2 L C_{cons}^2 G^2 + c G^2 C_{cons}$, where $C''(L, G, R, c) \in [0, \infty)$ is a constant that depends on $L, G, R, c$, and is independent of $N, \mu(N)$.

The proof of this extension is in Appendix.

**Remark.** Note that distributed Nesterov gradient uses the diminishing step size $\alpha_k = c/(k+1)$. Even the centralized Nesterov gradient with $\alpha_k = c/(k+1)$ achieves the rate $O(1/k)$ (and not $O(1/k^2)$). We use a diminishing step size with distributed Nesterov gradient, and not the constant one – which would give $O(1/k^2)$ in the centralized setting, is that the disagreement estimate $\|\hat{\theta}(k) - y_i(k)\|$ would not converge to zero. Thus, the oracle inexactness $\delta_k = L \sum_{i=1}^{N} \|\hat{\theta}(k) - y_i(k)\|^2$ in Lemma 2 would not converge to zero, and the effect of $\delta_k$ would accumulate too much over iterations $k$, destroying the convergence rate. In fact, algorithms distributed Nesterov gradient and distributed Nesterov+consensus offer two different ways to control $\delta_k$’s. Distributed Nesterov gradient does this through a diminishing step-size $\alpha_k$, while D–NC uses the inner consensus iterations (while using the constant step size $\alpha_k = \alpha$).

**Proof of Theorem 5:** The proof consists of two parts. First, we estimate the optimality gap $\frac{1}{N}(f(\hat{\theta}(k)) - f^*)$ at the point $\hat{\theta}(k) = \sum_{i=1}^{N} x_i(k)$ using Lemma 2 and the inexact oracle machinery. Second, we estimate the optimality gap $\frac{1}{N}(f(x_i(k)) - f^*)$ at any node $i$ using convexity of the $f_i$’s and the bound on the distances $\|x_i(k) - \hat{\theta}(k)\|$ from Lemma 4.

**Step 1. Optimality gap** $\frac{1}{N}(f(\hat{\theta}(k)) - f^*)$. Recall that, for $k = 1, 2, \ldots$, $(\tilde{f}_k, \tilde{g}_k)$ in (30) is a $(L_k, \delta_k)$ inexact oracle of $f$ at point $\tilde{\theta}(k)$ with $L_k = 2NL$ and $\delta_k = L\|\tilde{g}(k)\|^2$. Note that $(\tilde{f}_k, \tilde{g}_k)$ is also a $(L'_k, \delta_k)$ inexact oracle of $f$ at point $\tilde{\theta}(k)$ with $L'_k = N\frac{1}{c}(k+1) = N\frac{1}{\alpha_k}$, because $\frac{1}{c} \geq 2L$, and so $L'_k \geq L_k$. Now, we apply Lemma 2 to (31), where we set $\phi \equiv f$, $\hat{\theta}(k) \equiv \hat{\theta}(k)$, $\tilde{g}(k) \equiv \tilde{g}(k)$, $x^* = x^*$, and the Lipschitz constant $L'_k = 1/(\alpha_k/N) = 2NL(k+1)$. (Note also
that $\delta_k = L \sum_{i=1}^{N} \| y_i(k) - \bar{y}(k) \|^2 = L \| \bar{y}(k) \|^2$.) We get:

$$\frac{(k+1)^2}{k} (f(\bar{x}(k)) - f^*) + \frac{2N}{c} \| \bar{y}(k) - x^* \|^2 \leq \frac{k^2 - 1}{k} (f(\bar{x}(k-1)) - f^*) + \frac{2N}{c} \| \bar{y}(k-1) - x^* \|^2 + L \| \bar{y}(k-1) \|^2 \frac{(k+1)^2}{k}.$$  

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

$$\frac{(k+1)^2 - 1}{k+1} \leq \frac{k^2 - 1}{k} (f(\bar{x}(k-1)) - f^*) + \frac{2N}{c} \| \bar{y}(k-1) - x^* \|^2 \leq \frac{k^2 - 1}{k} \leq \frac{2N}{c} \| \bar{y}(0) - x^* \|^2 + L \sum_{t=1}^{k+1} \| \bar{y}(t-1) \|^2 \frac{(t+1)^2}{t}. $$

Applying Lemma 4 to the last equation, and using $\bar{x}(0) = \bar{x}(0)$, gives: $\frac{(k+1)^2 - 1}{k+1} \leq \frac{2N}{c} \| \bar{y}(0) - x^* \|^2 + L \sum_{t=1}^{k+1} \| \bar{y}(t-1) \|^2 \frac{(t+1)^2}{t}$. Applying Lemma 4 to the last equation, and using $\bar{x}(0) = \bar{x}(0)$, gives:

$$f(\bar{x}(k)) - f^* \leq \frac{1}{k} \frac{2N}{c} \| \bar{y}(0) - x^* \|^2 + \frac{16 c^2 N}{k} L C cons G^2 \sum_{t=2}^{k} \frac{(t+1)^2}{t(t-1)^2}. $$

**Step 2. Optimality gap** $(f(x_i(k)) - f^*)$. Fix an arbitrary node $i$; then, by convexity of $f_j, j = 1, 2, ..., N$: $f_j(\bar{x}(k)) \geq f_j(x_i(k)) + \nabla f_j(x_i(k))^\top (\bar{x}(k) - x_i(k))$, and so: $f_j(x_i(k)) \leq f_j(\bar{x}(k)) + G \| \bar{x}(k) - x_i(k) \|$. Summing the above inequalities for $j = 1, ..., N$, using $\sum_{i=1}^{N} \| \bar{x}(k) - x_i(k) \| = \sqrt{N} \| \bar{x}(k) \|$, subtracting $f^*$ from both sides, and using $\| \bar{x}(k) \| \leq \sqrt{N} c G cons (1/k)$ from Lemma 4:

$$f(x_i(k)) - f^* \leq f(\bar{x}(k)) - f^* + G_K \sqrt{N} \| \bar{x}(k) \| \leq f(\bar{x}(k)) - f^* + \frac{c N C cons G^2}{k} \frac{1}{k}. $$

which, with (49) (in which the summation variable $t$ is replaced by $t+1$) completes the proof.

**Network Scaling.** We examine how the convergence rate depends on $N$ and the network topology. We assume that $L, G, c, R$ do not depend on $N$. Just to formally set up the network scaling, suppose that we have a given sequence of the weight matrices $W^{(1)}, W^{(2)}, ..., W^{(N)}, ...$, where the size of $W^{(N)}$ is $N \times N$. (The matrices increase in size $N$ to emulate the increase of the network.) Then, we consider how $C(N)$ in Theorem 5 depends on the matrix $W = W^{(N)}$. Recall that $\mu(N)$ equals the second largest eigenvalue of $W^{(N)}$. For simplicity, write simply $W$ instead of $W^{(N)}$; likewise, we write $\bar{W} = W^{(N)} - J$. With many network models, $\mu(N) \to 1$ as $N \to \infty$, i.e., the network speed of consensus deteriorates with the increase of $N$. The expander
graphs are exceptions, for which the weight matrix $W$ can be chosen such that $1 - \mu(N) = \Omega(1)$.

We distinguish two cases: 1) nodes do not know $L$ and $\mu(N)$ before the algorithm run, and they set the step-size constant $c$ to a constant independent of $N$, e.g., $c = c(N) = 1$; and 2) nodes know $L, \mu(N)$, and they set $c = c(N) = \Theta (1 - \mu(N))$.

**Theorem 6** Consider algorithm D–NG under Assumptions 1–3, with step-size $\alpha_k = c/(k+1)$.

Also, suppose that $\lambda_2(\tilde{W}) = \Omega(1)$ as $N \to \infty$.

(a) For arbitrary $c = c(N) = \text{const} > 0$:

$$\frac{1}{N} (f(x_i(k)) - f^*) = O \left( \frac{1}{(1 - \mu(N))^{3+\xi} k} \log k \right).$$

(b) For $c = c(N) = \Theta (1 - \mu(N))$:

$$\frac{1}{N} (f(x_i(k)) - f^*) = O \left( \frac{1}{(1 - \mu(N))^{1+\xi} k} \log k \right).$$

Proof of Theorem 6 is in Appendix. We can derive explicitly how $1/(1 - \mu(N))$ depends on $N$ for commonly used models, like grids, geometric graphs, and expanders. The resulting dependence of $1/(1 - \mu(N))$ on $N$ can be found in [14].

VI. **Algorithm D–NC: Convergence Analysis**

Subsection VI-A provides the disagreement estimate, while Subsection VI-A gives the convergence rate and network scaling.

A. **Disagreement estimate**

**Consensus Estimate.** For notational simplicity, throughout this subsection, we set $d = 1$, but all results hold for generic $d$. We now estimate the disagreement $\tilde{x}(k) := x(k) - \bar{x}(k)1 = (I - J)x(k)$, and $\tilde{y}(k) := y(k) - \bar{y}(k)1 = (I - J)y(k)$ with the D–NC.

**Lemma 7** Let Assumptions 1–3 hold, and consider algorithm D–NC. Further, set the number of the inner consensus iterations $\tau_k$ at the $k$-outer iteration $k$ as in (12). Then, for $k = 1, 2, \ldots$:

$$\|\tilde{x}(k)\| \leq 2\alpha\sqrt{NG} \frac{1}{k^{\xi}}, \quad \|\tilde{y}(k)\| \leq 2\alpha\sqrt{NG} \frac{1}{k^{\xi}}.$$

---

This is true, e.g., with the weights example based on the neighbors’ degrees in Section II.
Proof: Denote by $B_{t-1} := \max \{\|\tilde{x}(t-1)\|, \|\tilde{y}(t-1)\|\}$, and fix $t - 1$. We want to upper bound $B_t$. Multiplying (13)–(14) by $(I - J)$ from the left, using $(I - J)W = \hat{W}(I - J)$:

$$
\begin{align*}
\tilde{x}(t) &= \hat{W}^{\tau_t} \tilde{y}(t) - \alpha\hat{W}^{2\tau_t - 1}(I - J)\nabla F(y(t - 1)) \tag{53}\\
\tilde{y}(t) &= \hat{W}^{\tau_t} [ \tilde{x}(t) + \beta_{t-1}(\tilde{x}(t) - \tilde{x}(t - 1)) ] . \tag{54}
\end{align*}
$$

From (53) and (54), using the sub-additive and sub-multiplicative properties of norms, $\|\hat{W}\| = \mu(N) \in (0, 1)$, $\| (I - J)\nabla F(y(t - 1)) \| \leq \|\nabla F(y(t - 1))\| \leq \sqrt{NG}$, and $\beta_{t-1} \leq 1$:

$$
\begin{align*}
\|\tilde{x}(t)\| &\leq (\mu(N))^{\tau_t - 1} \|\tilde{y}(t - 1)\| + \alpha(\mu(N))^{2\tau_t - 1}\sqrt{NG} \\
&\leq (\mu(N))^{\tau_t - 1} B_{t-1} + \alpha(\mu(N))^{2\tau_t - 1}\sqrt{NG} \\
\|\tilde{y}(t)\| &\leq 2(\mu(N))^{\tau_t - 1} \|\tilde{x}(t)\| + (\mu(N))^{\tau_t - 1}\|\tilde{x}(t - 1)\| \\
&\leq 2(\mu(N))^{2\tau_t - 1} B_{t-1} + 2\alpha\sqrt{N}(\mu(N))^{3\tau_t - 1}G + (\mu(N))^{\tau_t - 1}B_{t-1}. \tag{56}
\end{align*}
$$

Using $(\mu(N))^{2\tau_t - 1} \leq (\mu(N))^{\tau_t - 1}$ and $(\mu(N))^{3\tau_t - 1} \leq (\mu(N))^{\tau_t - 1}$ to further upper bound (55) and (56), and taking the maximum over the two resulting inequalities:

$$
B_t \leq 3(\mu(N))^{\tau_t - 1}B_{t-1} + 2\alpha\sqrt{N}(\mu(N))^{\tau_t - 1}G. \tag{57}
$$

Now, recall $\tau_{k-1}$ in (12). We have that $3(\mu(N))^{\tau_t - 1} = 3e^{-\frac{\log 3}{\log \mu(N)}}e^{-\frac{\log 3}{\log \mu(N)}}\log \mu(N) = \frac{1}{\tau^3}$. Applying the latter to (57), and using $(\mu(N))^{\tau_t - 1} \leq (\mu(N))^{-\frac{\log 3}{\log \mu(N)}} = e^{-\frac{\log 3}{\log \mu(N)}}\log \mu(N) = \frac{1}{\tau^3}$:

$$
B_t \leq \frac{1}{\tau^3}B_{t-1} + \frac{1}{\tau^3}2\alpha\sqrt{NG}. \quad \text{Next, using } B_0 = 0, \text{ and unwinding the latter recursion for } t = k, k - 1, \ldots, 1, k \geq 1:
$$

$$
B_k \leq 2\alpha\sqrt{NG} \left( \sum_{t=2}^{k} \frac{1}{k^3(k-1)^3 \cdots t^3} + \frac{1}{k^3(k-1)^3 \cdots 2^3} \right) \leq 2\alpha\sqrt{NG}k \frac{1}{k^3} = 2\alpha\sqrt{NG} \frac{1}{k^2}.
$$

\section*{B. Convergence rate and network scaling}

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

\textbf{Theorem 8} Consider algorithm D–NC under Assumptions 1–3, with the constant step size $\alpha \leq 1/(2L)$, and the number inner iterations $\tau_k$ as in (12). Let $\|x(0) - x^*\| \leq R, R > 0$. Then, after

$$
\mathcal{K} = 3\sum_{t=0}^{k-1} \tau_t \leq \frac{3}{-\log \mu(N)}(k \log 3 + (k + 1) \log (k + 1)) = O(k \log k) \tag{58}
$$

October 2, 2012
communication rounds, i.e., after \( k \) outer iterations, we have, at any node \( i \): 
\[
\frac{1}{N} (f(x_i(k)) - f^*) \leq \frac{1}{k^2} \left( \frac{2}{\alpha} R^2 + 11 \alpha^2 L G^2 + \alpha G^2 \right), \quad k = 1, 2, \ldots 
\] 

**Proof:** As with the proof of Theorem 5, we divide the proof in two steps. In the first step, we upper bound the optimality gap \( f(x_i(k)) - f^* \); in the second step, we upper bound the optimality gap at each node \( f(x_i(k)) - f^* \).

**Step 1: Upper bounding** \( f(x_i(k)) - f^* \). Recall that the evolution (29) with \( \alpha_k = \alpha \) for \((\bar{x}(k), \bar{y}(k))\) is the Nesterov gradient with the inexact oracle \((\bar{f}_k, \bar{g}_k)\) in (30), and \((L_k = 2NL, \delta_k = L\|\bar{y}(k)\|^2)\). Also, note that, from Lemma 7, \( \delta_k \leq 4\alpha^2 L N G^2 \frac{1}{k^3} \). Now, similarly to the proof of Theorem 5, we apply Lemma 2 with \( \phi \equiv f, \bar{x}(k) \equiv \bar{x}(k), \bar{y}(k) \equiv \bar{y}(k), \bar{v}(k) \equiv \bar{v}(k) \), and \( x^* \equiv x^* \) to get:

\[
(k + 1)^2 (f(\bar{x}(k)) - f^*) + 2N\parallel \bar{x}(k) - x^*\parallel^2 
\leq (k^2 - 1) (f(\bar{x}(k-1)) - f^*) + 2N\parallel \bar{x}(k-1) - x^*\parallel^2 + L\parallel \bar{y}(k-1)\parallel^2 (k + 1)^2.
\]

Next, using \((k + 1)^2 (f(\bar{x}(k)) - f^*) \geq ((k + 1)^2 - 1) (f(\bar{x}(k)) - f^*)\), by unwinding the above recursion, and using \( \bar{x}(0) = \bar{x}(0) \):

\[
((k + 1)^2 - 1) (f(\bar{x}(k)) - f^*) \leq \frac{2N}{\alpha} \parallel \bar{x}(0) - x^*\parallel^2 + L \sum_{t=1}^{k} \parallel \bar{y}(t-1)\parallel^2 (t + 1)^2.
\]

The latter equation, using \((k + 1)^2 - 1 = k(k + 2) \geq k^2\), and substituting \( \|\bar{y}(t-1)\| \leq 2\alpha\sqrt{N} G \frac{1}{(t-1)^2} \) for \( t \geq 1 \), and \( \|\bar{y}(t-1)\| = 0 \) for \( t \geq 1 \), obtain:

\[
f(\bar{x}(k)) - f^* \leq \frac{1}{k^2} \left( \frac{2N}{\alpha} \parallel \bar{x}(0) - x^*\parallel^2 + 11 \alpha^2 L N G^2 \right)
\] 

(60)

where we used \( 4 \sum_{t=2}^{\infty} \frac{(t+1)^2}{(t-1)^2} = 4 \sum_{t=1}^{\infty} \frac{(t+2)^2}{t^2} = 4 (\zeta(2) + 4\zeta(3) + 4\zeta(4)) \leq 11 \).

**Step 2:** Upper bounding \( f(x_i(k)) - f^* \). This step is the same as with Theorem 5; setting \( \parallel \bar{x}(k)\parallel \leq \alpha\sqrt{N} G \frac{1}{k^2} \), obtain: \( f(x_i(k)) - f^* \leq f(\bar{x}(k)) - f^* + \alpha NG^2 \frac{1}{k^3} \); combining the latter with (60), we get the desired result. \( \blacksquare \)

**Network scaling.** We now give the network scaling for algorithm D–NC in Theorem 9. As before with algorithm D–NG, we assume that \( L, G, \) and \( R \) do not depend on \( N \). Also, we assume that nodes know \( L \) and \( \mu(N) \) before the algorithm run. Proof of Theorem 9 is omitted.
and is provided in Appendix.

**Theorem 9** Consider algorithm D–NC under Assumptions 1–3 with \( \tau_k \) as in (12) and \( \alpha \leq 1/(2L) \). Then, after \( K \) communication rounds, at any node \( i \), the optimality gap \( \frac{1}{N} (f(x_i) - f^*) \) is \( \mathcal{O} \left( \frac{1}{(1-\mu(N))^2} \frac{\log k}{K^{2-\xi}} \right) \).

**VII. COMPARISONS: D–NG AND D–NC, AND EXISTING ALGORITHMS**

We compare D–NG, D–NC, and existing distributed (sub)gradient algorithms in [6], [14], from the aspects of implementation (Subsection VII-A) and convergence rate (Subsection VII-B).

**A. Comparisons: Algorithm implementation**

We discuss the required (global) knowledge for the algorithms’ convergence, and on the required knowledge for the algorithm stopping and tuning of the step-size.

**Algorithm D–NG.** Interestingly, D–NG does not require any global knowledge for convergence (achieving the rate \( \mathcal{O} \left( \frac{1}{(1-\mu(N))^{3+\xi}} \frac{\log k}{k^{1/2}} \right) \)), except that each node knows only its own and its neighbors’ degrees \( d_j, j \in O_i \).

We now comment on the required knowledge for nodes to a priori determine the stopping iteration \( K \) when \( \frac{1}{N} (f(x_i(k)) - f^*) \leq \epsilon \), for all \( i \), for all \( k \geq K \), and a small \( \epsilon > 0 \). To determine \( K \), all nodes need \( C(N) \) (see Theorems 5), and so they need: a Lipschitz constant \( L \); a gradient upper bound \( G \); an upper bound on \( \mu(N) \); and an upper bound on \( R \). Clearly, the same knowledge is also needed for setting the optimal \( c^* \) – the value of \( c \) that minimizes \( C(N) \) over \( c \in (0, 1/(2L)) \); \( c^* \) can be easily numerically obtained, as \( C(N) \) is convex in \( c \).

**Algorithm D–NC.** In contrast with D–NG, D–NC needs upper bounds on \( \mu(N) \) (to set \( \tau_k \) in (12)) and \( L \) for convergence. (See ahead Figure 1, left.) With respect to the stopping iteration and the optimal step-size \( \alpha \), all nodes need upper bounds on \( L, G, \mu(N) \), and \( R \) – the same as with D–NG. Again, \( \alpha^* \) that minimizes the constant \( \frac{2R^2}{\alpha} + 11\alpha^2LG^2 + \alpha G^2 \) over \( (0, 1/(2L)) \) can be easily found numerically, as \( \frac{2R^2}{\alpha} + 11\alpha^2LG^2 + \alpha G^2 \) is convex in \( \alpha \).

Finally, D–NG has a simpler structure than D–NC, and hence admits easier implementation; this is relevant, e.g., with the inexpensive sensor network mote processors.

**Algorithms in [6], [14].** We focus only on [14], while [6] is similar. Like D–NG, [14] does not require any global knowledge for convergence with the guaranteed rate \( \mathcal{O} \left( \frac{1}{(1-\mu(N))^2} \frac{\log k}{k^{1/2}} \right) \). The weight matrix \( W \) with [14] does not need to satisfy (1); however, [14] still needs \( \mu(N) < 1 \).
We are not aware if it is possible to set a symmetric $W$ with $\mu(N) < 1$ using a “less global” knowledge than what we require with D–NG (each node knowing its own and the neighbors’ degrees.) For stopping and optimal step-size, [14] requires all that D–NG requires, except an upper bound on $L$.

Comments on acquiring the global knowledge $\mu(N), L, G, R$. We comment how nodes can acquire $L, G, R$ and $\mu(N)$ before the run of an algorithm to solve (2). Consider first $L$ (see Assumption 2.) Suppose that each node knows a Lipschitz constant $L_i$ of its own cost function $f_i$. Then, $L$ can be taken as $L = \max_{i=1,...,N} L_i$. Thus, each node can compute $L$ if all nodes run a distributed algorithm for maximum computation, e.g., algorithm in equation (1) in [27]; all nodes get $L$ after $O(\text{Diam}) = O(N)$ per-node communicated scalars, where Diam is the network diameter. Likewise, a gradient bound $G$ (see Assumption 3) can be also taken as $G = \max_{i=1,...,N} G_i$, where $G_i$ is a gradient bound for the $f_i$. The quantity $\mu(N)$ (equal to the second largest eigenvalue of $W$) can be computed in a distributed way, e.g., by algorithm DECENTRALOI, proposed for a more general setting in [28], and adapted to the problem like the one we consider in [29], Subsection IV-A, p. 2519. With DECENTRALOI, node $i$ obtains $q^\mu_i$, the $i$-th coordinate of the $N \times 1$ eigenvector $q^\mu$ of $W$ that corresponds to $\mu(N)$, (up to $\epsilon$-accuracy) after $O\left(\frac{\log^2(N/\epsilon) \log N}{1-\mu(N)}\right)$ per-node communicated scalars [28]; then, node $i$ obtains $\mu(N)$ as: $\frac{\sum_{j \in \mathcal{N}} W_{ij} q^\mu_j}{q^\mu_i}$.

B. Comparisons: Convergence rate

Absence of global knowledge. When no global knowledge except the neighbors’ degrees is available, D–NG achieves $O\left(\frac{1}{(1-\mu(N))^{3+\epsilon}} \frac{\log k}{k} \right)$ and $O\left(\frac{1}{(1-\mu(N))^{3+\epsilon}} \frac{\log K}{K}\right)$. Without global knowledge of (at least upper bounds on) $\mu(N)$ and $L$, D–NC is not guaranteed to converge. Algorithm in [14] achieves $O\left(\frac{1}{(1-\mu(N))^{3/2}} \frac{\log k}{k}\right)$ and $O\left(\frac{1}{(1-\mu(N))^{3/2}} \frac{\log K}{K}\right)$. The latter suggests that, for larger $k$ and $K$ (higher accuracy), D–NG is better than [14]. On the other hand, for sufficiently small $k$ and $K$, and sufficiently large $1/(1-\mu(N))$, [14] may be better. (The optimality gap with [14] increases slower with $1/(1-\mu(N))$.) However, in our simulations with up to 400 nodes, D–NG had a smaller optimality gap than [14] for any value of $k$ and $K$. (See also Figure 1.)

Presence of global knowledge $L$ and $\mu(N)$. The algorithm D–NG achieves $O\left(\frac{1}{(1-\mu(N))^{3+\epsilon}} \frac{\log k}{k}\right)$ and $O\left(\frac{1}{(1-\mu(N))^{3+\epsilon}} \frac{\log K}{K}\right)$; D–NC is $O\left(\frac{1}{K^2}\right)$ and $O\left(\frac{1}{(1-\mu(N))^2} \frac{\log k}{K^{3/2}}\right)$. The algorithm in [14] is $O\left(\frac{1}{(1-\mu(N))^{3/2}} \frac{\log k}{K^{3/2}}\right)$ and $O\left(\frac{\log K}{K^{3/2}}\right)$. Thus, in terms of $k$, D–NC is better than D–NG. In terms of $K$, our
big-O results suggest that, for sufficiently small $\mathcal{K}$ (lower accuracy), D–NG may be better than D–NC. Likewise, [14] may be better than D–NG and D–NC for sufficiently small $\mathcal{K}$ (low accuracies.) However, for sufficiently large $\mathcal{K}$ (high accuracy), D–NC eventually becomes the best. In all simulations that we performed, with networks up to 400 nodes, D–NC becomes better than D–NG only at very high accuracies $\frac{f(x_i) - f^*}{L} (f^* \neq 0)$ of order $10^{-6} - 10^{-8}$ or smaller (or never becomes better,) and D–NG is better than [14] for all accuracies. Finally, consider $\mathcal{K}(N; \epsilon)$ – the number of communication rounds needed to reduce $\frac{1}{N}(f(x_i) - f^*)$ below $\epsilon$.

With D–NG, $\mathcal{K}(N; \epsilon) = O\left(\frac{1}{(1-\mu(N))\epsilon}\right)$; with D–NC, $\mathcal{K}(N; \epsilon) = O\left(\frac{1}{(1-\mu(N))\epsilon^{1/2}}\right)$; and with [14], $\mathcal{K}(N; \epsilon) = O\left(\frac{1}{(1-\mu(N))\epsilon^{3/4}}\right)$. Hence, with all algorithms, for a fixed $\epsilon$, $\mathcal{K}(N; \epsilon) \approx a(\epsilon)1/(1-\mu(N))$, i.e., it scales linearly. While D–NC guarantees the smallest (best) slope $a(\epsilon)$ when $\epsilon \to 0$, D–NG shows the best slope for practical accuracies in simulation (See Figure 2.)

The $\Omega(1/k^{2/3})$ lower bound on the worst-case optimality gaps for [6]. We now focus on the dependence of convergence rate on $k$ and $\mathcal{K}$ only (assuming a finite, fixed $1/(1-\mu(N))$.) It is important to note that the optimality gap $O(\log k/k^{1/2})$ and $O(\log \mathcal{K}/\mathcal{K}^{1/2})$ in [14] holds for the non-differentiable convex $f_i$'s that are Lipschitz continuous (have bounded gradients) – a wider class of functions than what we consider. To our knowledge, a detailed study of the algorithms in [6], [14] under Assumptions 2 and 3 does not exist. We thus demonstrate here that D–NG has a strictly better convergence rate in $k$ (and $\mathcal{K}$) than [6], when applied to the class of functions defined by Assumptions 2 and 3. (Thus, D–NC also has a better rate.) The method in [14] performs very similarly to [6] (or slightly worse) in our simulations.

We clarify mathematically the claim that we make. Fix a generic, connected network $\mathcal{G}$ with $N$ nodes, and fix a weight matrix $W$ that satisfies (1). Let $\mathcal{F} = \mathcal{F}(L,G)$ be the class of all convex functions $\phi : \mathbb{R}^d \to \mathbb{R}$ that have Lipschitz continuous derivative with constant $L$ and bounded gradient with bound $G$. Consider (2) with $f_i \in \mathcal{F}$, for all $i$; consider D–NG with the step-size $\alpha_k = \frac{c}{(k+1)^\tau}$, $k = 0, 1, ..., c \leq 1/(2L)$. Denote by: $\mathcal{E}^{D-NG}(k,R) = \sup_{f_i \in \mathcal{F}} \sup_{\|\pi(0) - x^*\| \leq R} \max_{i=1,...,N} \{f(x_i(k)) - f^*\}$ the optimality gap at the $k$-th iteration of D–NG for the worst functions $f_i \in \mathcal{F}$, and the worst initial condition (provided $\|\pi(0) - x^*\| \leq R$.) From Theorem 5, for any $k = 1, 2, ...$: $\mathcal{E}^{D-NG}(k,R) \leq C(N) \log k = O(\log k/k)$, where $C(N)$ is given in (46). Now, consider the algorithm in [6] with the step-size $\alpha_k = \frac{c}{(k+1)\tau}$, $k = 0, 1, ..., c \in [0, 1/(2L)]$, $\tau \geq 0$ are the degrees of freedom for the step-size choice. (We constrain $c \leq 1/(2L)$, similarly as with D–NG.) With this algorithm, $k = \mathcal{K}$. We can show that there
exists a network (the $N = 2$-node connected network), a weight matrix $W$ that satisfies (1), and the values for $R$, $L$ and $G$, such that, with [6]:

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

(61)

where $\mathcal{E} (k, R; \tau, c) = \sup_{f_i \in F} \sup_{(x^0) : \|x^0 - x^*\| \leq R} \max_{i=1, \ldots, N} \{ f(x_i(k)) - f^* \}$ is the worst-case optimality gap when the step-size $\alpha_k = \frac{c}{(k+1)^\tau}$ is used. We omit the proof due to the lack of space.

**VIII. Simulation examples**

This Section provides simulation examples on the logistic (Subsections VIII-A and VIII-C), and Huber loss (Subsection VIII-B). Simulations show that D–NG outperforms the existing algorithms in [6], [14], both in terms of the number of communication rounds and the number of iterations (per-node gradient evaluations). Further, D–NC is not competitive with D–NG for practical accuracies ($(f(x_i) - f^*)/f^* \approx 10^{-6} - 10^{-8}$ or coarser). Finally, by simulations in Subsection VIII-C, D–NG and D–NC confirm the linear scaling of the number of communication rounds for $\epsilon$-accuracy versus the inverse of the spectral gap.

**A. Distributed learning of the best linear classifier via the logistic loss**

**Optimization problem.** We consider distributed learning via the logistic loss; see, e.g., [5] for further details. Nodes minimize the logistic loss: $\sum_{i=1}^N f_i(x) = \sum_{i=1}^N \log \left( 1 + e^{-b_i(a_i^T x + x_0)} \right)$, where $x = (x_1^T, x_2^T)^T$, $a_i \in \mathbb{R}^2$ 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 of $f(x) = \sum_{i=1}^N f_i(x)$ is: $\nabla^2 f(x) = \sum_{i=1}^N \frac{e^{-b_i a_i^T x}}{(1 + e^{-b_i a_i^T x})^2} c_i c_i^T$, where $c_i = (b_i a_i^T, b_i)^T \in \mathbb{R}^3$. A Lipschitz constant $L$ should satisfy $\|\nabla^2 f(x)\| \leq NL$, $\forall x \in \mathbb{R}^d$. Note that $\nabla^2 f(x) \preceq \frac{1}{4} \sum_{i=1}^N c_i c_i^T$, because $\frac{e^{-b_i a_i^T y}}{(1 + e^{-b_i a_i^T y})^2} \leq 1/4$ for all $y$. We thus choose $L = \frac{1}{4N} \left\| \sum_{i=1}^N c_i c_i^T \right\| \approx 0.3053$, for this example. None of the $f_i$’s is strongly convex on $\mathbb{R}^d$; $f$ is not strongly convex on $\mathbb{R}^d$ either; there exists a sequence $x_n$ with $\lim_{n \to \infty} \|x_n\| \to \infty$, such that $\lim_{n \to \infty} \|\nabla^2 f(x_n)\| = 0$. However, for the numerical values that we considered, $f$ was strongly convex in a neighborhood of the solution $x^*$.

**Data.** We generate $a_i$ independently over $i$; each entry is drawn from the standard normal distribution. We generate the “true” vector $x^* = (y_1^T, y_0^T)^T$ by drawing its entries independently from the standard normal distribution. The class labels are generated as $b_i = \text{sign} \left( x_1^T a_i + x_0^* + \epsilon_i \right)$, where the $\epsilon_i$’s are drawn independently from a normal distribution with zero mean and variance 3.
Network. 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 \( \left( \text{number of links} \right) / N(N-1)/2 \) = 10%.

Algorithm parameters and metrics. With D–NG, we set the \( W_{ij} \)'s based on the neighbors’ degrees as mentioned in Section II. With D–NC, [6], and [14], we use the Metropolis weights (that require the same knowledge to be set.) The step-size \( \alpha_k \) is: \( \alpha_k = 1/(k+1) \), with D–NG; \( \alpha = 1/(2L) \), with D–NC; and \( 1/(k+1)^p \), with [6] and [14], with \( p \in \{1/3, 1/2, 1\} \). Note that we do not optimize the constant in the step-sizes with [6], [14] nor with D–NG – they operate without the prior knowledge of \( L, \mu(N) \). We simulate the relative average error across nodes

\[
\frac{1}{N^2} \sum_{i=1}^{N} (f(x_i) - f^*) , \quad f^* \neq 0,
\]

versus the total number of communications at all nodes (\( = NK \)).

Results: D–NG versus D–NC. Figure 1 (left) compares D–NG (solid, red curve) versus D–NC in terms of the number of communication rounds \( \mathcal{K} \). Solid, blue curve shows the performance of D–NC for \( \tau_k \) in (12) – the value that guarantees the convergence result in Theorem 8, and

\[
\alpha = 1/(2L) \quad \text{– the largest step-size for which we can guarantee convergence on the entire class of the } f_i \text{'s that obey Assumptions 2 and 3.}
\]

D–NG performs better than D–NC for the accuracies smaller than about \( 10^{-8} \); at higher accuracies, D–NC becomes better than D–NG. Further, Figure 1 (left) shows D–NC when we decrease \( \tau_k \) (see the legend in Figure 1, left). For sufficiently small values of \( \tau_k \), D–NC converges only to a solution neighborhood. Note that, to appropriately set \( \tau_k \), nodes need to know an upper bound on \( \mu(N) \), and so this global knowledge is critical for the convergence of D–NC. Finally, Figure 1 (left) shows the behavior of D–NC when we increase the step-size \( \alpha \). For a sufficiently large \( \alpha \), D–NC fails to converge; again, the knowledge of (an upper bound on) \( L \) is critical.

Results: D–NG and D–NC versus [6], [14]. Figure 1 (center) compares our algorithms D–NG and D–NC with [6], [14]. First, we can see that D–NG outperforms [6], [14]. For example, for the precision \( 10^{-3} \), D–NG reduces the number of communication rounds with respect to [6], [14] about 40 times – from about 12,800 transmissions to about 423,000 transmissions for [6] with \( \alpha_k = 1/(k+1)^p \) and \( p = 1/2 \). Note also that D–NG achieves a faster rate than [6], [14]; D–NG achieves about \( 1/K^2 \), while [6], [14] with \( p = 1/2 \) achieve about \( 1/K \). Finally, with both [6], [14], the step size choice \( p = 1/2 \) performs better than \( p = 1 \) or \( p = 1/3 \). D–NC performs better than [6], [14] for the accuracies \( 10^{-3} \) or finer, and worse for the lower accuracies.
B. Huber loss

We also consider the Huber loss cost functions, which finds applications in, e.g., distributed estimation in sensor networks [1]. Given a node $i$’s private, scalar value (sensor’s measurement) $\theta_i$, the $f_i : \mathbb{R} \rightarrow \mathbb{R}$ is the Huber loss $f_i(x) = \|x - \theta_i\|^2$ if $\|x - \theta_i\| \leq 1$, and $f_i(x) = \|x - \theta_i\| - 1/2$, else. The $f_i$’s obey Assumptions 2 and 3, and no $f_i$ is strongly convex, nor is $f(x) = \sum_{i=1}^{N} f_i(x)$.

The setup is the same as in the previous subsection, unless specified here otherwise. The network is a geometric graph with $N = 101$ nodes and the relative degree $\approx 10\%$. We set $\theta_i = 10i$, $i = 1, ..., N = 101$; the Lipschitz constant $L = 1$; the algorithms in [14], [6] use $p = 1/2$; with D–NC, we set $\tau_k$ as in (12) and $\alpha = 1/(2L)$. Figure 1 (right) compares D–NC, D–NG, [6], and [14]. We can see that D–NG significantly outperforms the other algorithms. Also, it is worth noting that here D–NG and D–NC do not intersect at a high accuracy like with the logistic loss example; rather, D–NG persists in being better. Note that this does not contradict the theory, as both D–NG and D–NC achieve (from Figure 1, right) about $O(1/K^2)$.

![Fig. 1](image)

**Left and center:** Logistic loss; **Right:** Huber loss. In the left Figure, $\gamma = 1 - \mu(N)$ is the spectral gap.

C. Logistic loss: Network scaling

We simulate the dependence of the number of communications $K(\mu; \epsilon)$ required for the $\epsilon$-accuracy versus the inverse of the spectral gap $1/(1 - \mu)$, in the presence of global knowledge before the algorithm run. Ignoring $\xi$-small and logarithmic factors, with both D–NG and D–NC, but also with distributed dual averaging in [14], theory predicts (at worse) linear asymptotic dependence $O(1/(1 - \mu))$ when $\mu \rightarrow 1^-$. We consider a geometric graph with $N = 7$ nodes. We set the weight matrix as $W = I - wL$, where $L$ is the unweighted Laplacian matrix, and $w > 0$ is the weight. We emulate the network scaling (deterioration of the spectral gap) by decreasing...
the weight $w$. We let $w \in \{0.0714; 0.0238; 0.0143; 0.0102; 0.0079\}$. Note that $\mu = 1 - w \lambda_2(\mathcal{L}) = 1 - 3w$, in this example. The step-sizes are: 1) D–NG: $\alpha_k = w/(k+1)$ (See Theorem 6); 2) dual averaging: $\alpha_k = w^{1/2}/(k + 1)^{1/2}$ (See Theorem 2 in [14]); 3) D–NC: $\alpha = 1/(2L)$.

We find $\mathcal{K}(\mu; \epsilon)$ – the smallest number of communication rounds at which the average relative error $\frac{1}{N \tau} \sum_{i=1}^{N} (f(x_i) - f^*)$ falls below $\epsilon$. Figure 2 plots $\mathcal{K}(\mu; \epsilon)$ versus $1/(1 - \mu)$ for D–NG, D–NC, and distributed dual averaging; Figure 2, left is for $\frac{1}{N \tau} \sum_{i=1}^{N} (f(x_i) - f^*) = 10^{-1.4}$, while Figure 2, center is for $\frac{1}{N \tau} \sum_{i=1}^{N} (f(x_i) - f^*) = 10^{-6}$. First, we can see that, with all three algorithms, the theoretical linear dependence is confirmed well in simulation. Second, D–NG is the best for both higher and lower precision. Lastly, the gain of D–NG over D–NC decreases with the increase of precision, as predicted by theory. (See Figure 2, right.) Namely, $\rho(\mu) := \mathcal{K}(\mu; \epsilon; D - NC)/\mathcal{K}(\mu; \epsilon; D - NG)$ is much smaller ($\rho(\mu)$ equals $2 - 3$) for the higher accuracy $\epsilon = 10^{-6}$ than for the lower accuracy $\epsilon = 10^{-1.4}$ ($\rho(\mu)$ equals $6 - 6.5$.) Theory predicts that, when $\epsilon \to 0$, D–NC eventually becomes better; this occurs only at very small $\epsilon$.

![Figure 2](image-url)

**Fig. 2.** Network scaling: Number of communication rounds $\mathcal{K}(\mu; \epsilon)$ needed to reduce the average relative error $\frac{1}{N \tau} \sum_{i=1}^{N} (f(x_i) - f^*)$ below $\epsilon$ versus the inverse spectral gap $1/(1 - \mu)$. **Left:** Low accuracy: $\epsilon = 10^{-1.4}$; **Center:** High accuracy: $\epsilon = 10^{-6}$; **Right:** Ratio $\rho(\mu) = \mathcal{K}(\mu; \epsilon; 10^{-6})/\mathcal{K}(\mu; \epsilon; 10^{-1.4})$. 

**IX. CONCLUSION**

We studied distributed optimization in networks where nodes minimize the sum of their individual cost functions $\sum_{i=1}^{N} f_i(x)$ subject to a global variable $x \in \mathbb{R}^d$. Existing work has proposed distributed gradient based algorithms to solve the above problem and studied their convergence rates, under a wide class of convex, non-differentiable $f_i$’s, with bounded gradients – for unconstrained problems (Lipschitz $f_i$’s over the constraint set – for constrained problems.) In this paper, we asked whether faster convergence rates (than the rates established in the literature) can be achieved on a more structured class of $f_i$’s – convex, with Lipschitz continuous gradient (with constant $L$) and bounded gradient (with constant $G$). Building from the centralized Nesterov
gradient, we answer affirmatively this question by proposing two distributed gradient algorithms. Our algorithm D–NG achieves the rate $O\left(\frac{1}{(1-\mu(N))^{1+\varepsilon}} \log \frac{K}{k}\right)$ and $O\left(\frac{1}{(1-\mu(N))^{1+\varepsilon}} \frac{\log k}{k}\right)$, when the knowledge of global parameters $L$ and $\mu(N)$ is not available before the algorithm run. The rate, for the optimized step size, improves to $O\left(\frac{1}{(1-\mu(N))^{2+\varepsilon}} \frac{\log K}{K}\right)$ and $O\left(\frac{1}{(1-\mu(N))^{2+\varepsilon}} \frac{\log k}{k}\right)$, when $L$ and $\mu(N)$ are available before the run. Our algorithm D–NC operates only if $L$ and $\mu(N)$ are available and achieves the rate $O\left(\frac{1}{(1-\mu(N))^2} \frac{1}{K^{2-\varepsilon}}\right)$ and $O\left(\frac{1}{k^{2}}\right)$. We also showed that (under a fixed $N$) our methods achieve strictly faster rates than the method in [6], and that, under a fixed accuracy $\epsilon$, the communication cost with our methods scales linearly with the inverse of the spectral gap $1/(1 - \mu(N))$. Finally, simulations with the logistic and Huber losses show that our D–NG algorithm outperforms the existing methods.

Acknowledgement. We thank an anonymous reviewer whose instructive comments led us to develop algorithm D–NC. We also thank the anonymous reviewers and the associate editor for several useful suggestions regarding the presentation and organization of the paper. We thank as well João F. C. Mota for pointing us to relevant references and for useful discussions.

References

[1] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in IPSN 2004, 3rd International Symposium on Information Processing in Sensor Networks, Berkeley, California, USA, April 2004, pp. 20 – 27.

[2] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, “On decentralized negotiation of optimal consensus,” Automatica, vol. 44, no. 4, pp. 1175–1179, 2008.

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

[4] G. Mateos, J. A. Bazerque, and G. B. Giannakis, “Distributed sparse linear regression,” IEEE Transactions on Signal Processing, vol. 58, no. 11, pp. 5262–5276, November 2010.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends in Machine Learning, Michael Jordan, Editor in Chief, vol. 3, no. 1, pp. 1–122, 2011.

[6] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” IEEE Transactions on Automatic Control, vol. 54, no. 1, pp. 48–61, January 2009.

[7] S. S. Ram, A. Nedic, and V. Veeravalli, “Asynchronous gossip algorithms for stochastic optimization,” in CDC ’09, 48th IEEE International Conference on Decision and Control, Shanghai, China, December 2009, pp. 3581 – 3586.

[8] A. Nedic, A. Ozdaglar, and A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” IEEE Transactions on Automatic Control, vol. 55, no. 4, pp. 922–938, April 2010.

[9] S. Ram, A. Nedic, and V. Veeravalli, “Distributed stochastic subgradient projection algorithms for convex optimization,” Journal of Optimization Theory and Applications, vol. 147, no. 3, pp. 516–545, 2011.
[10] I. Lobel, A. Ozdaglar, and D. Feijer, “Distributed multi-agent optimization with state-dependent communication,” Mathematical Programming, vol. 129, no. 2, pp. 255–284, 2011.
[11] I. Lobel and A. Ozdaglar, “Convergence analysis of distributed subgradient methods over random networks,” in 46th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, September 2008, pp. 353 – 360.
[12] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, “Subgradient methods and consensus algorithms for solving separable distributed control problems,” in CDC’08, 47th IEEE Conference on Decision and Control, Cancun, Mexico, December 2008, pp. 4185–4190.
[13] I. Matei and J. S. Baras, “Performance evaluation of the consensus-based distributed subgradient method under random communication topologies,” IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 4, pp. 754–771, 2011.
[14] J. Duchi, A. Agarwal, and M. Wainwright, “Dual averaging for distributed optimization: Convergence and network scaling,” IEEE Transactions on Automatic Control, vol. 57, no. 3, pp. 592–606, March 2012.
[15] K. Tsianos and M. Rabbat, “Distributed consensus and optimization under communication delays,” in to appear in proc. 49th Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, Sept. 2011.
[16] M. Zhu and S. Martínez, “On distributed convex optimization under inequality and equality constraints via primal-dual subgradient methods,” to appear in IEEE Transactions on Automatic Control, 2011, dOI: 10.1109/TAC.2011.2167817.
[17] Y. E. Nesterov, “A method for solving the convex programming problem with convergence rate $O(1/k^2)$,” Dokl. Akad. Nauk SSSR, vol. 269, pp. 543–547, 1983, (in Russian).
[18] D. Jakovetic, J. Xavier, and J. M. F. Moura, “Cooperative convex optimization in networked systems: Augmented Lagrangian algorithms with directed gossip communication,” IEEE Transactions on Signal Processing, vol. 59, no. 8, pp. 3889–3902, August 2011.
[19] J. Mota, J. Xavier, P. Aguiar, and M. Pueschel, “Basis pursuit in sensor networks,” in ICASSP ’11, IEEE International Conference on Acoustics, Speech, and Signal Processing, Prague, Czech Republic, May 2011, pp. 2916–2919.
[20] ———, “Distributed basis pursuit,” submitted for publication, July 2011, available at: http://arxiv.org/abs/1009.1128.
[21] U. V. Shanbhag, J. Koshal, and A. Nedic, “Multiuser optimization: distributed algorithms and error analysis,” SIAM Journal on Control and Optimization, vol. 21, no. 2, pp. 1046–1081, 2011.
[22] H. Terelius, U. Topcu, and R. M. Murray, “Decentralized multi-agent optimization via dual decomposition,” in 18th World Congress of the International Federation of Automatic Control (IFAC), Milano, Italy, August 2011, identifier: 10.3182/20110828-6-IT-1002.01959.
[23] 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.
[24] L. Vandenberghe, “Optimization methods for large-scale systems,” 2010, lecture notes, available at: http://www.ee.ucla.edu/~vandenbe/ee236c.html.
[25] D. Jakovetic, J. M. F. Moura, and J. Xavier, “Distributed nesterov-like gradient algorithms,” in submitted to CDC’12, 51st IEEE Conference on Decision and Control, March 2012.
[26] O. Devolder, F. Glineur, and Y. Nesterov, “First-order methods of smooth convex optimization with inexact oracle,” submitted to Mathematical Programming, 2011, available at: http://www.optimization-online.org/DB_FILE/2010/12/2865.pdf.
[27] G. Shi and K. H. Johansson, “Finite-time and asymptotic convergence of distributed averaging and maximizing algorithms,” 2012, available at: http://arxiv.org/pdf/1205.1733.pdf.
[28] D. Kempe and F. McSherry, “A decentralized algorithm for spectral analysis,” in 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, August 2004, pp. 561–568.
APPENDIX A

PROOF THAT CONDITION (1) HOLDS FOR THE WEIGHT CHOICE EXAMPLE IN SECTION II

Consider the weight choice: \( W_{ij} = \frac{1}{1 + 3 \max \{d_i, d_j\}} \) for \( \{i, j\} \in E \); \( W_{ij} = 0 \) for \( i \neq j \) and \( \{i, j\} \notin E \); and \( W_{ii} = 1 - \sum_{j \neq i} W_{ij} \). We prove that for such a weight choice, under Assumption 1 of the connected underlying network, property (1) holds.

We first prove the second condition in (1). Consider the associated weighted Laplacian \( L_w := I - W \), and the associated unweighted Laplacian \( L \) with \( L_{ij} = -1 \) if, and only if, \( i \neq j, W_{ij} > 0 \); \( L_{ij} = 0 \) if, and only if, \( i \neq j, W_{ij} = 0 \); and \( L_{ii} = -\sum_{j \neq i} L_{ij} \). Then, to prove the second condition in (1), we have, by the inequality of the spectral and the max-row-sum norms, and because \( |[L_w]_{ii}| = \sum_{j \neq i} |[L_w]_{ij}| \):

\[
\|L_w\| \leq \|L_w\|_1 = 2 \max_{i=1,\ldots,N} \sum_{j \neq i} |[L_w]_{ij}| \leq 2 \max_{i=1,\ldots,N} \frac{d_i}{1 + 3d_i} \leq 2/3.
\]

Thus, \( \lambda_1(W) = \lambda_2(\widetilde{W}) = 1 - \|L_w\| \geq 1 - 2/3 = 1/3 > 0 \), and the second condition in (1) holds.

We now prove the first condition in (1). Recall that \( \mu(N) = \|W - J\| = \|I - J - L_w\| \), and so \( \mu(N) = \max\{|1 - \lambda_2(L_w)|, |1 - \lambda_N(L_w)|\} \), which, because \( 0 \leq \lambda_2(L_w) \leq \lambda_N(L_w) = \|L_w\| < 1 \), equals \( \mu(N) = 1 - \lambda_2(L_w) \). Further:

\[
\lambda_2(L_w) = \min_{\{p|p^T1=1,p^T1=0\}} p^T L_w p = \min_{\{p|p^T1=1,p^T1=0\}} \sum_{\{i,j\} \in E} W_{ij} (p_i - p_j)^2 \\
\geq \min_{\{p|p^T1=1,p^T1=0\}} \sum_{\{i,j\} \in E} (p_i - p_j)^2 \frac{1}{1 + 3d_{\text{max}}} = \frac{\lambda_2(L)}{1 + 3d_{\text{max}}} > 0,
\]

because the second smallest eigenvalue of the unweighted Laplacian \( L \) on a connected network is positive. Thus, \( \mu(N) \leq 1 - \frac{1}{1+3d_{\text{max}}} \lambda_2(L) < 1 \).

APPENDIX B

PROOF OF THE RESULT IN REMARK BELOW THEOREM 5

We let Assumptions 1–3 hold, and we estimate the optimality gap \( \frac{1}{N}(f(x_i(k)) - f^*) \) at any node \( i \) when condition \( c \leq \frac{1}{2L} \) does not hold. Suppose \( c > \frac{1}{2L} \). Denote by \( k' = 2cL \). We will
show that:

\[
C'(N) = \frac{(k')^2 - 1}{k'} L \left( \left( \frac{3k' - 1}{3 - 1} \right)^2 4c^2 G^2 + R^2 \right) + \frac{2}{c} \left( 2(2k' + 1)^2 \left( \frac{3k' - 1}{3 - 1} \right)^2 4c^2 G^2 + 2R^2 \right) + 16c^2 C_{\text{cons}}^2 G^2 + c C_{\text{cons}} G^2. \tag{62}
\]

Progress equation (51) still holds if \( L' = \frac{N(k+1)}{c} \geq 2NL \), i.e., if \( k \geq k' := 2cL \). Telescoping (51) backwards from \( k > k' \) until \( k' \), and using \( \frac{(k+1)^2 - 1}{k+1} \geq k \):

\[
k \frac{N}{N} (f(\bar{x}(k)) - f^*) \\
\leq \frac{(k')^2 - 1}{k' N} (f(\bar{x}(k' - 1)) - f^*) + \frac{2}{c} \|\bar{v}(k' - 1) - x^*\|^2 + \frac{L}{N} \sum_{t=k'}^{k} \|\bar{y}(t-1)\|^2 \frac{(t+1)^2}{t} \\
\leq \frac{(k')^2 - 1}{k' N} (f(\bar{x}(k' - 1)) - f^*) + \frac{2}{c} (2\|\bar{v}(k' - 1)\|^2 + 2\|x^*\|^2) + \frac{L}{N} \sum_{t=1}^{k} \|\bar{y}(t-1)\|^2 \frac{(t+1)^2}{t}. \tag{63}
\]

Lemma 4 holds unchanged if \( c > 1/(2L) \), and so (Compare also to (49)):

\[
\frac{L}{N} \sum_{t=1}^{k} \|\bar{y}(t-1)\|^2 \frac{(t+1)^2}{t} \leq \frac{16c^2}{k} C_{\text{cons}}^2 G^2 \sum_{t=2}^{k} \frac{(t+2)^2}{t(t-1)^2}. \tag{64}
\]

We now upper bound \( \|\bar{v}(k' - 1)\| \), where we recall \( \bar{v}(k) = \frac{1}{\theta_k} \bar{y}(k) - \frac{1 - \theta_k}{\theta_k} \bar{x}(k) \), and \( \theta_k = \frac{2}{k+2} \). Then:

\[
\|\bar{v}(k-1)\| \leq \frac{k+1}{2} ||\bar{y}(k-1)|| + k||\bar{x}(k-1)|| \leq (2k+1)M_{k-1}, \tag{65}
\]

where \( M_{k-1} := \max \{ ||\bar{y}(k-1)||, ||\bar{x}(k-1)|| \} \). By (31), using \( \alpha_{k-1} = c/k \), using \( \beta_{k-1} \leq 1 \), and \( \|\sum_{i=1}^{N} \nabla f_i(y_i(k-1))\| \leq NG \):

\[
\|\bar{x}(k)\| \leq ||\bar{y}(k-1)|| + \frac{cG}{k} \leq M_{k-1} + \frac{cG}{k} \\
\|\bar{y}(k)\| \leq 2||\bar{x}(k)|| + ||\bar{x}(k-1)|| \leq 2||\bar{y}(k-1)|| + ||\bar{x}(k-1)|| + \frac{2cG}{k} \leq 3M_{k-1} + \frac{2cG}{k},
\]

and so:

\[
M_k \leq 3M_{k-1} + \frac{2cG}{k}, \quad k = 1, 2, \ldots, \quad M_0 = 0.
\]
By unwinding the above recursion from \( k = k' - 1 \) to \( k = 0 \), we get:

\[
M_{k' - 1} \leq \left( \frac{3^{k'} - 1}{3 - 1} \right)^2 \lambda cG.
\]

(66)

Further, combining the last equation with (65), and squaring the resulting inequality:

\[
\|\pi(k' - 1)\|^2 \leq (2k' + 1)^2 \left( \frac{3^{k'} - 1}{3 - 1} \right)^2 4\lambda^2 c \lambda G^2.
\]

(67)

Using the Lischitz continuity of \( f \) (with constant \( LN \)), and using \( \nabla f(x^*) = 0 \):

\[
f(\pi(k' - 1)) - f^* \leq \nabla f(x^*)^\top (\pi(k' - 1) - x^*) + \frac{LN}{2} \|\pi(k' - 1) - x^*\|^2
\]

\[
\leq (LN) (\|\pi(k' - 1)\|^2 + \|x^*\|^2)
\]

\[
\leq (LN) \left( \left( \frac{3^{k'} - 1}{3 - 1} \right)^2 4\lambda^2 c \lambda G^2 + \|x^*\|^2 \right),
\]

where we used (66) to upper bound \( M_{k' - 1} \). Finally, combining the last equation with (63), (64), (67), using \( \sum_{t=1}^{\infty} \frac{(t+1)^2}{t^3} \leq 1 \), for all \( k \), and repeating the same argument as in (53):

\[
\frac{1}{N} (f(x_i(k)) - f^*) \leq C'(N) \sum_{t=2}^{k} \frac{(t + 2)^2}{t(t - 1)^2}, \quad k > k',
\]

where, recalling \( \|x^*\| \leq R \) as \( x(0) = y(0) = 0 \), we obtain (62): Compare \( C(N) \) in (46) with \( C'(N) \) in (62): \( C'(N) \geq C(N) \): the first (nonnegative) summand \( \frac{(k')^2 - 1}{k' - 1} L \left( \left( \frac{3^{k'} - 1}{3 - 1} \right)^2 4\lambda^2 c \lambda G^2 + R^2 \right) \) in (62) does not appear in (46); also, the second summand \( \frac{2}{c} \left( 2(2k' + 1)^2 \left( \frac{3^{k'} - 1}{3 - 1} \right)^2 4\lambda^2 c \lambda G^2 + 2R^2 \right) \) is larger than the corresponding summand \( \frac{2R^2}{c} \) in (46).

**APPENDIX C**

**PROOF OF THEOREM 6**

Consider the sequence of the weight matrices \( W^{(1)}, \ldots, W^{(N)}, \ldots \), where \( W^{(N)} \) has the dimension \( N \times N \). Write simply \( W = W^{(N)} \), and \( \tilde{W} = W - J \), and recall that \( \mu(N) = \|\tilde{W}\| \). Suppose that \( \lambda_2(\tilde{W}) = \Omega(1) \), i.e., there exists sufficiently large \( N_0 \) and \( \underline{\lambda} \in (0, 1) \), such that \( \lambda_2(\tilde{W}) \geq \underline{\lambda} > 0 \), for all \( N \geq N_0 \).

We first show the scaling result for \( C_{cons} = C_{cons}(N) \) in (32). Then, we apply the scaling result for \( C_{cons} \) to the constant \( C'(N) \) in (62) to prove part (a) of the Theorem: \( C'(N) = O \left( \frac{1}{(1 - \mu(N))^{3+\varepsilon}} \right) \) as \( N \to \infty \); finally, we use the scaling for \( C_{cons} \) and the constant \( C(N) \) in Theorem 5 (equa-
tion (46)) to prove part (b) of the Theorem: \( C(N) = O\left(\frac{1}{(1 - \mu(N))^{1+\xi}}\right) \) as \( N \to \infty \).

**Scaling of \( C_{\text{cons}} \) in (32).** We will show:

\[
C_{\text{cons}} = O\left(\frac{1}{(1 - \mu(N))^{3/2+\xi}}\right) \quad \text{when} \quad N \to \infty, \tag{68}
\]

for arbitrarily small \( \xi > 0 \). We separately upper bound each of the terms in \( C_{\text{cons}} \):

\[
1/ \min \left\{ \sqrt{\lambda_2(\tilde{W})(1 - \lambda_2(\tilde{W}))}, \sqrt{\mu(N)(1 - \mu(N))} \right\}, 2B\left(\sqrt{\mu(N)}\right), \frac{2(2C_h+1)}{e} \frac{1}{(-\log \mu(N))}, \text{ and } \frac{2}{1 - \sqrt{\mu(N)}}.
\]

First consider:

\[
1/ \min \left\{ \sqrt{\lambda_2(\tilde{W})(1 - \lambda_2(\tilde{W}))}, \sqrt{\mu(N)(1 - \mu(N))} \right\}. \quad \text{We have } \lambda_2(\tilde{W})(1 - \lambda_2(\tilde{W})) \geq \lambda(1 - \mu(N)), \quad N \geq N_0; \quad \text{likewise, } \mu(N)(1 - \mu(N)) \geq \lambda(1 - \mu(N)), \quad N \geq N_0.
\]

Thus:

\[
1/ \min \left\{ \sqrt{\lambda_2(\tilde{W})(1 - \lambda_2(\tilde{W}))}, \sqrt{\mu(N)(1 - \mu(N))} \right\} \leq \frac{1}{\sqrt{\lambda}} \frac{1}{\sqrt{1 - \mu(N)}}, \quad N \geq N_0. \tag{69}
\]

Further, recall the definition of the function \( B : (0, 1) \to \mathbb{R} \):

\[
B(r) = \sup_{z \geq 1/2} \{z r^2 \log(1 + z)\}.
\]

For any \( \xi > 0 \), there exists a constant \( K_B(\xi) \in (0, \infty) \) such that: \( \log(1+z) \leq K_B(\xi)z^\xi, \forall z \geq 1/2 \). Thus:

\[
B(r) = \sup_{z \geq 1/2} \{z r^2 \log(1 + z)\} \leq K_B(\xi) \sup_{z \geq 1/2} \{z^{1+\xi}r^2\}
\]

\[
= K_B(\xi) e^{-(1+\xi)(1 + \xi)}(1 + \xi)^{(1+\xi)} \frac{1}{(-\log r)^{1+\xi}} \tag{70}
\]

Note from the above that \( B(r) < \infty \), for all \( r \in (0, 1) \).

Now, consider the term: \( \frac{2(2C_h+1)}{e} \frac{1}{(-\log \mu(N))}. \) Because the function \( \frac{1}{-\log z} \leq \frac{1}{1-z} \), for all \( z \in (0, 1) \), we have that:

\[
\frac{2(2C_h+1)}{e} \frac{1}{(-\log \mu(N))} \leq \frac{2(2C_h+1)}{e} \frac{1}{1 - \mu(N)}. \tag{71}
\]

Consider next the term \( \frac{2}{1 - \sqrt{\mu(N)}}. \) Because \( \frac{1}{1-z} \leq \frac{2}{1-z}, \) for all \( z \in (0, 1) \), we have:

\[
\frac{2}{1 - \sqrt{\mu(N)}} \leq \frac{4}{1 - \mu(N)}. \tag{72}
\]

Now, consider the term \( 2B\left(\sqrt{\mu(N)}\right). \) From (70), and because \( 1/(-\log \sqrt{z}) \leq 2/(1 - z), \)
Combining (69), (71), (72), and (73), we obtain that $C_{\text{cons}}(N) \leq M_{\text{cons}}(N)\frac{1}{(1-\mu(N))^{3/2+\xi}}$, for all $N \geq N_0$, where the constant $M_{\text{cons}}(\xi)$ is independent of $N$ and of the matrices $W^{(N)}$, $N \geq 1$. Thus, the result in (68).

**Proof of part (a) of Theorem 6.** Recall that we assumed that $L$, $G$, $R$, and $c$ do not depend on $N$, and also $k' = 2Lc$ does not depend on $N$. In (62), only $C_{\text{cons}}$, given by (32), depends on $N$. Consider $C'(N)$ in (62). We can see that $C'(N) = O(C_{\text{cons}}^2) = O(1/(1-\mu(N))^{3+2\xi}) = O(1/(1-\mu(N))^{3+\xi'})$, where $\xi' = 2\xi > 0$ is arbitrarily small.

**Proof of part (b) of Theorem 6.** Plugging $c = \Theta(1-\mu(N))$ in $C(N)$ in (46), we have $C(N) = O\left(\frac{1}{1-\mu(N)} + \left(\frac{1}{1-\mu(N)}\right)^{3+2\xi'}(1-\mu(N))^2 + \frac{1-\mu(N)}{(1-\mu(N))^{3/2+\xi'}}\right) = O(1/(1-\mu(N))^{1+\xi'})$, where $\xi' = 2\xi > 0$ is arbitrarily small. This completes the Proof of Theorem 6.

**APPENDIX D**

**PROOF OF THEOREM 9**

We will prove that, after $\mathcal{K}$ communication rounds, \( \frac{1}{N}(f(x_i) - f^*) \leq C(N)\frac{1}{k^{2-\xi'}} \), where $C(N) = O\left(\frac{1}{(1-\mu(N))^2}\right)$ when $N \to \infty$. We first make the dependence of the optimality gap on $\mathcal{K}$ more explicit. From (58), for arbitrary small $\xi > 0$, there exists a constant $C_0(\xi) \in (1, \infty)$ that depends only on $\xi$, such that: $3k \log 3 + 3(k+1) \log(k+1) \leq C_0(\xi)k^{1+\xi}$, $\forall k \geq 1$. From the last equation, for $\mathcal{K}$ in (58): $\mathcal{K} \leq C_0(\xi)\frac{1}{-\log \mu(N)}k^{1+\xi} \leq C_0(\xi)\frac{1}{1-\mu(N)}k^{1+\xi}$, where the last inequality uses $1/(\log z) \leq 1/(1-z)$, $z \in (0,1)$. Raising to power $1/(1+\xi)$, and using $C_0(\xi) \geq 1$, $1/(1+\xi) \geq 1-\xi$, $\xi \in (0,1)$, gives:

\[
\mathcal{K}^{1-\xi} \leq C_0(\xi)\frac{1}{k}.
\]

From the last equation:

\[
\frac{1}{k^2} \leq (C_0(\xi'/2))^2 \left(\frac{1}{1-\mu(N)}\right)^2 \frac{1}{\mathcal{K}^{2-\xi'}}.
\]
where we introduced $\xi' = \xi/2$. Applying the last equation to Theorem 9, we obtain that, after $K$ communication rounds, $\frac{1}{K} (f(x_i) - f^*)$ is, for any node $i$, upper bounded by:

$$
(C_0(\xi' / 2))^2 \frac{1}{K^{2-\xi'}} \left( \frac{1}{1 - \mu(N)} \right)^2 \left( \frac{2}{\alpha} R^2 + 11 \alpha^2 LG^2 + \alpha G^2 \right),
$$

(74)

for arbitrarily small $\xi' > 0$. Thus, after $K$ communication rounds, $\frac{1}{N} (f(x_i) - f^*) \leq C(N) \frac{1}{K^{2-\xi'}}$, where $C(N) = O \left( \frac{1}{(1-\mu(N))^2} \right)$ when $N \to \infty$. Thus, the result in Theorem 9.

**APPENDIX E**

**PROOF OF THE $\Omega(1/k^{2/3})$ LOWER BOUND ON THE WORST-CASE OPTIMALITY GAP FOR [6]**

We prove (61) by constructing a hard example of the functions $f_i$ and a hard initial condition $x(0)$.

**Network and functions $f_i$.** Consider $N = 2$ nodes, and set the $2 \times 2$ weight matrix $W$ as $W_{11} = W_{22} = 1 - w$, and $W_{12} = W_{21} = w$, and $w = 1/8$. The eigenvalue decomposition is $W = QAQ^T$, with $Q = [q_1, q_2]$, the columns $q_1 = \frac{1}{\sqrt{2}} (-1, 1)^T$, $q_2 = \frac{1}{\sqrt{2}} (1, 1)^T$, and $\Lambda$ is diagonal with the eigenvalues $\lambda_1 = \lambda_2 = 1 - 2w = 3/4$, $\lambda_2 = \lambda_2 = 1$.

We set $d = 2$. We denote the coordinates of the 2-dimensional variable $x$ by $x = (x^{(1)}, x^{(2)})^T$; for example, the estimate at iteration $k$ of node $i$ is then denoted by $x_i(k) = (x_i^{(1)}(k), x_i^{(2)}(k))^T$. The nodes solve problem (2), where the functions $f_i : \mathbb{R}^2 \to \mathbb{R}$ are, for $i = 1, 2$:

$$
f_i^\eta(x) = \begin{cases} 
\eta (x^{(1)} + (-1)^i)^2 + \frac{(x^{(2)} + (-1)^i)^2}{2} \\
\chi \left[ \eta (x^{(1)} + (-1)^i)^2 + (x^{(2)} + (-1)^i)^2 \right]^{1/2} - \frac{\chi^2}{2} 
\end{cases}
$$

if $\eta (x^{(1)} + (-1)^i)^2 + (x^{(2)} + (-1)^i)^2 \leq \chi^2$

else.

(75)

Here $\chi > 0$ is a constant that we set to $\chi = 6$; $\eta$ is a constant in $[0, 1]$, for which we subsequently design a particular value that makes problem (2) hard. The function $f_i^\eta$ is similar to the Huber loss; it is quadratic in the region

$$
\mathcal{R}_i = \left\{ x \in \mathbb{R}^2 : \eta (x^{(1)} + (-1)^i)^2 + (x^{(2)} + (-1)^i)^2 \leq \chi^2 \right\},
$$

(76)

and outside of this region it behaves as the norm $x \to \|x\|$. The solution to (2), with $f(y) = f_1^\eta(y) + f_2^\eta(y)$, is $x^* = (0, 0)^T$, and the corresponding optimal value is $f^* = \eta + 1$.

**Properties of the $f_i^\eta$’s.** We now show that the $f_i^\eta$’s are convex, have Lipschitz continuous gradient with constant $L = \sqrt{2}$, and bounded gradients $\|\nabla f_i^\eta(x)\| \leq 10$, for all $x$, $i = 1, 2$. Thus, the $f_i^\eta$’s in (75) belong to the class $\mathcal{F} = \mathcal{F}(L = \sqrt{2}, G = 10)$, for any $\eta \in [0, 1]$. 

October 2, 2012 DRAFT
To show that the function \( x \mapsto f_1^n(x) \) is convex, note that it can be represented as the following concatenation: \( x \mapsto y = (\sqrt{\eta}(x^{(1)} - 1), (x^{(2)} - 1)) = f_h(z) = f_1^n(x) \), where \( f_h : \mathbb{R}_+ \rightarrow \mathbb{R} \) is the Huber loss: 
\[
    f_h(z) = \frac{1}{2} z^2, \text{ if } \|z\| \leq \chi \text{, and } f_h(z) = \chi(\|z\| - \chi/2), \text{ else.}
\]
Hence, \( x \mapsto f_1^n(x) \) is a concatenation of an affine function, a convex function, and a convex non-decreasing function, and hence it is convex. Analogously, we can show that \( x \mapsto f_2^n(x) \) is convex.

We now show the Lipschitz continuity and the boundedness of the gradient of \( f_1^n \); the gradient equals:
\[
    \nabla f_1^n(x) = \left( \frac{\partial f_1^n}{\partial x(1)}(x), \frac{\partial f_1^n}{\partial x(2)}(x) \right)^\top = \begin{cases} 
(\eta(x^{(1)} - 1), (x^{(2)} - 1))^\top & \text{if } x \in \mathcal{R}_1 \\
\frac{\chi}{\sqrt{\eta}(x^{(1)} - 1)^2 + (x^{(2)} - 1)^2} \eta(x^{(1)} - 1), (x^{(2)} - 1))^\top & \text{else.}
\end{cases}
\]
(77)

The first coordinate of the gradient \( x \mapsto \frac{\partial f_1^n}{\partial x(1)}(x) \) can be expressed as the following concatenation of the functions: 
\[
    x \mapsto y = (x^{(1)} - 1, x^{(2)} - 1)^\top \mapsto z = (\sqrt{\eta}y^{(1)}, y^{(2)}) \mapsto w = \text{Proj}_{B_0, \chi}(z) \mapsto v = \frac{\partial f_1^n}{\partial x(1)}(x) = \sqrt{\eta}w^{(1)},
\]
where \( \text{Proj}_{B_0, \chi}(z) \) is the projection of \( z \) on the ball centered at zero with radius \( \chi \). All the functions \( \phi \) in the concatenation above are Lipschitz continuous with constant one, and so \( x \mapsto \frac{\partial f_1^n}{\partial x(1)} \) is also Lipschitz continuous with constant one. (Given the function \( \phi_m(\phi_{m-1}(\ldots(\phi_1(x)))) \), where \( \phi_1, \ldots, \phi_{m-1} \) are Lipschitz continuous of constant one, we have 
\[
    \|\phi_m(\phi_{m-1}(\ldots(\phi_1(x)))) - \phi_m(\phi_{m-1}(\ldots(\phi_1(y))))\| \leq \|\phi_{m-1}(\ldots(\phi_1(x))) - \phi_{m-1}(\ldots(\phi_1(y)))\| \leq \ldots \|x - y\|.
\]
Similarly, we can show that \( x \mapsto \frac{\partial f_1^n}{\partial x(2)} \) is Lipschitz continuous with constant one. This implies that the gradient \( x \mapsto \nabla f_1^n(x) \) is Lipschitz continuous with constant \( \sqrt{2} \). Also, 
\[
    \|\frac{\partial f_1^n}{\partial x(1)}(x)\| \leq \sqrt{\eta}\chi \leq 6, \text{ for all } x. \text{ (Recall the concatenation representation } x \mapsto y = (x^{(1)} - 1, x^{(2)} - 1)^\top \mapsto z = (\sqrt{\eta}y^{(1)}, y^{(2)}) \mapsto w = \text{Proj}_{B_0, \chi}(z) \mapsto v = \frac{\partial f_1^n}{\partial x(1)}(x) = \sqrt{\eta}w^{(1)}; \text{ then, for any } x \in \mathbb{R}^2, \|\frac{\partial f_1^n}{\partial x(1)}(x)\| \leq \sqrt{\eta}\|\text{Proj}_{B_0, \chi}(z)\|, \text{ for some } z \in \mathbb{R}^2, \text{ and so } \|\frac{\partial f_1^n}{\partial x(1)}(x)\| \leq \sqrt{\eta}\chi. \) Similarly, 
\[
    \|\frac{\partial f_1^n}{\partial x(2)}(x)\| \leq \chi \leq 6. \text{ Thus, for the gradient, we have: } \|\nabla f_1^n(x)\| \leq 6\sqrt{2} < 10, \text{ for all } x. \text{ We can analogously show that } \|\nabla f_2^n(x)\| \leq 6\sqrt{2} < 10, \text{ for all } x.
\]

The algorithm in [6]. Now, consider the algorithm in [6], and consider \( x_i(k) \)–the solution estimate at node \( i \) and time \( k \). Denote by \( x^l(k) = (x^l_1(k), x^l_2(k))^\top \)–the vector that stacks the \( l \)-th coordinate of the solution estimate of both nodes, \( l = 1, 2 \). Also, denote by \( d^l(k) = \left( \frac{\partial f_1(x^l_1(k))}{\partial x^l(1)}, \frac{\partial f_2(x^l_2(k))}{\partial x^l(1)} \right) \), \( l = 1, 2 \). Then, the update rule of [6] is, for the \( f_1^n, f_2^n \) in (75), and for \( l = 1, 2 \):
\[
    x^l(k) = Wx^l(k - 1) - \alpha_{k-1}d^l(k - 1), \; k = 1, 2, \ldots
\]
(78)
We set the specific initialization $x^1(0) = (1, 1)^\top$, and $x^\Pi(0) = (0, 0)^\top$.

**Performance evaluation of [6].** We show that, under the above initialization, $x_i(k)$ belongs to the region $\mathcal{R}_i$ in (76) (the region where the $f_i^0$ is quadratic), for all $k$, for both nodes $i = 1, 2$.

We first prove that, if $\|x^1(k)\| \leq 2\sqrt{2}$, and $\|x^\Pi\| \leq 2\sqrt{2}$, then $x_i(k) \in \mathcal{R}_i, i = 1, 2$. Consider node $1$’s estimate $x_1(k)$. If $\|x^1(k)\| \leq 2\sqrt{2}$, and $\|x^\Pi\| \leq 2\sqrt{2}$, then $\|x^1(k)\| \leq 2\sqrt{2}, l = 1, 2$, and:

$$\eta(x_1^{(1)}(k) - 1)^2 + (x_1^{(2)}(k) - 1)^2 \leq 2(2\sqrt{2} + 1)^2 < 2(\frac{3}{2} + 1)^2 < 32 < \chi^2 = 36,$$

which means $x_1(k) \in \mathcal{R}_1$. (Analogously, we can show $x_2(k) \in \mathcal{R}_2$.)

We next prove that $\|x^l(k)\| \leq 2\sqrt{2}, l = 1, 2$, for all $k$; we do this by induction. For $k = 0$, $\|x^l(0)\| \leq 2\sqrt{2}, l = 1, 2$. Now, suppose that, for some $k \geq 1$, $\|x^l(k - 1)\| \leq 2\sqrt{2}, l = 1, 2$. Then, the update equations (78) to get $x^1(k)$ and $x^\Pi(k)$, using the derivatives of the $f_i^\eta$’s in the quadratic region in (77), become:

$$x^1(k) = (W - \alpha_{k-1}I) x^1(k - 1) - \alpha_{k-1} \eta (-1, 1)^\top$$  \hspace{1cm} (79)$$

$$x^\Pi(k) = (W - \alpha_{k-1}I) x^\Pi(k - 1) - \alpha_{k-1} (-1, 1)^\top.$$  \hspace{1cm} (80)

From (79)–(80), using the sub-additive and sub-multiplicative properties of norms, and the step-size value $\alpha_{k-1} = c/(k^\tau)$:

$$\|x^1(k)\| \leq \left(1 - \frac{cn}{k^\tau}\right) \|x^1(k - 1)\| + \frac{cn}{k^\tau} \sqrt{2}$$

$$= \|x^1(k - 1)\| - \frac{\eta c}{k^\tau} \left(\|x^1(k - 1)\| - \sqrt{2}\right).$$

Now, we distinguish two cases: 1) $\|x^1(k)\| \in [0, \sqrt{2}]$; and 2) $\|x^1(k)\| \in (\sqrt{2}, 2\sqrt{2}]$. In case 1, from the last equation:

$$\|x^1(k)\| \leq \|x^1(k - 1)\| + \frac{\sqrt{2}\eta c}{k^\tau} \leq 2\sqrt{2},$$

where we used $0 \leq c \leq 1/(2\sqrt{2}) = 1/(2L)$ and $0 \leq \eta \leq 1$. In case 2:

$$\|x^1(k)\| < \|x^1(k - 1)\| \leq 2\sqrt{2}.$$

Thus, we have shown that $\|x^1(k)\| \leq 2\sqrt{2}$. Similarly, we can show that $\|x^\Pi(k)\| \leq 2\sqrt{2}$. Thus, by induction, $\|x^l(k)\| \leq 2\sqrt{2}, l = 1, 2$, for all $k$, and so $x_i(k) \in \mathcal{R}_i, i = 1, 2$, for all $k$. 

October 2, 2012
We now evaluate the sum of the nodes’ optimality gaps \( \sum_{i=1}^{2} (f(x_i(k)) - f^*) \), \( f(x) = f_1^0(x) + f_2^0(x) \). Because \( x_i(k) \in R_i \), \( i = 1, 2 \), verify, using (75), and \( f^* = 1 + \eta \), that:

\[
\sum_{i=1}^{2} (f(x_1(k)) - f^*) = \eta \|x_1(k)\|^2 + \|x_1(k)\|^2. \tag{81}
\]

Because \( x_i(k) \in R_i \), for all \( k \), the recursions (79)–(80) hold for all \( k = 0, 1, 2, \ldots \), and we can write the solution for \( x_1(k) \) and \( x_1(k) \). By unwinding the recursions in (79)–(80), and using \( x_1(0) = (1, 1)^T \), \( x_1(0) = (0, 0)^T \):

\[
x_1(k) = (W - \alpha_{k-1} \eta I)(W - \alpha_{k-1} \eta I) \ldots (W - \alpha_0 \eta I)(1, 1)^T
\]
\[
+ \eta(-1, 1)^T \left( \sum_{t=0}^{k-2} (\lambda_1 - \alpha_{k-1} \eta)(\lambda_1 - \alpha_{k-2} \eta) \ldots (\lambda_1 - \alpha_t \eta) \alpha_t + \alpha_{k-1} \right).
\]
\[
x_1(k) = (-1, 1)^T \left( \sum_{t=0}^{k-2} (\lambda_1 - \alpha_{k-1} \eta)(\lambda_1 - \alpha_{k-2} \eta) \ldots (\lambda_1 - \alpha_t \eta) \alpha_t + \alpha_{k-1} \right). \tag{82}
\]

We simplify the above using the eigenvalue decomposition \( W = Q \Lambda Q^T \). The matrix \( W - \alpha_{k-1} \eta I \) decomposes as \( W - \alpha_{k-1} \eta I = Q(\Lambda - \alpha_{k-1} \eta I)Q^T \), and, similarly, \( W - \alpha_{k-1} I = Q(\Lambda - \alpha_{k-1} I)Q^T \). Then, the products \( (W - \alpha_{k-1} \eta I)(W - \alpha_{k-2} \eta I) \ldots (W - \alpha_t \eta I) = Q(\Lambda - \alpha_{k-1} \eta I) \ldots (\Lambda - \alpha_t \eta I)Q^T \), and \( (W - \alpha_{k-1} I)(W - \alpha_{k-2} I) \ldots (W - \alpha_t I) = Q(\Lambda - \alpha_{k-1} I) \ldots (\Lambda - \alpha_t I)Q^T \). Using these decompositions, and the orthogonality: \( q_1^T (1, 1)^T = 0 \), and \( q_2^T (-1, 1)^T = 0 \), we obtain:

\[
x_1(k) = (1 - \alpha_{k-1} \eta)(1 - \alpha_{k-2} \eta) \ldots (1 - \alpha_0 \eta)(1, 1)^T
\]
\[
+ \eta(-1, 1)^T \left( \sum_{t=0}^{k-2} (\lambda_1 - \alpha_{k-1} \eta)(\lambda_1 - \alpha_{k-2} \eta) \ldots (\lambda_1 - \alpha_t \eta) \alpha_t + \alpha_{k-1} \right).
\]
\[
x_1(k) = (-1, 1)^T \left( \sum_{t=0}^{k-2} (\lambda_1 - \alpha_{k-1} \eta)(\lambda_1 - \alpha_{k-2} \eta) \ldots (\lambda_1 - \alpha_t \eta) \alpha_t + \alpha_{k-1} \right). \tag{83}
\]

We next upper bound the norm of \( x_1(k) \). Note that \( \lambda_1 - \alpha_{k-1} \eta = 3/4 - \frac{c}{k \tau} \geq 1/4 \), for all \( k, \tau, c \). Also, \( \lambda_1 - \alpha_{k-1} \eta \leq \lambda_1 = 3/4 \), for all \( k, \tau, c \). Similarly, we can show \( 1 - \alpha_{k-1} \eta \in [1/2, 1] \). (Note that the terms \( (1 - \alpha_{k-1} \eta)(1 - \alpha_0 \eta) \) and \( (\lambda_1 - \alpha_{k-1} \eta)(\lambda_1 - \alpha_t \eta) \) in (82) are then nonnegative, \( \forall t \).) Now, we can upper bound the norm of \( x_1(k) \) as:

\[
\|x_1(k)\| \geq (1 - \alpha_{k-1} \eta)(1 - \alpha_{k-2} \eta) \ldots (1 - \alpha_0 \eta) - \eta c \sum_{t=0}^{k-1} \lambda_1^{k-t-1} \frac{1}{(t + 1)^\tau}.
\]
Further, using the inequality \((1 - a_1)(1 - a_2)...(1 - a_n) \geq 1 - (a_1 + a_2 + ... + a_n), a_i \in [0, 1), \forall i,\) and \(\alpha_k = \frac{c}{(k+1)^\tau}:\)

\[
\|x^1(k)\| \geq \left(1 - \eta c \sum_{t=0}^{k-1} \frac{1}{(t+1)^\tau}\right) - \eta c C(\lambda_1) \frac{1}{k^{\frac{\tau}{2}}},
\]

(84)

where we used \(\sum_{t=0}^{k-1} \lambda_1^{k-t} \frac{1}{(t+1)^\tau} \leq C(\lambda_1) \frac{1}{k^{\frac{\tau}{2}}},\) with \(C(\lambda_1) \in (0, \infty)\) depending only on \(\lambda_1,\) and not depending on \(\tau.\) (The latter can be proved similarly to the proof of Lemma 4.) Further, we assume from now on that \(\eta \in [0, 1]\) and \(k = 1, 2, ...\) are such that:

\[
\left(1 - \eta c \sum_{t=0}^{k-1} \frac{1}{(t+1)^\tau}\right) - \eta c C(\lambda_1) \frac{1}{k^{\frac{\tau}{2}}} \geq 0.
\]

(85)

(Note that such a choice of \(\eta, k\) exists; e.g., \(\eta = 0,\) and \(k\) is arbitrary.) Then, we can square (84); using the inequality \((a - b)^2 \geq a^2 - 2ab, a, b \geq 0,\) we obtain:

\[
\|x^1(k)\|^2 \geq \left(1 - 2\eta c \sum_{t=0}^{k-1} \frac{1}{(t+1)^\tau}\right) - 2\eta c C(\lambda_1) \frac{1}{k^{\frac{\tau}{2}}} \left(1 - \eta c \sum_{t=0}^{k-1} \frac{1}{(t+1)^\tau}\right).
\]

Finally, using the inequality \(k^{1-\tau} - 1 \leq \sum_{t=0}^{k-1} \frac{1}{(t+1)^\tau} \leq k^{1-\tau},\) and multiplying the last equation by \(\eta:\)

\[
\eta\|x^1(k)\|^2 \geq \eta \left(1 - 2\eta c k^{1-\tau}\right) - 2\eta^2 c C(\lambda_1) \left(1 - \eta c (k^{1-\tau} - 1)\right) \frac{1}{k^{\frac{\tau}{2}}}
\]

\[
= \eta \left(1 - 2\eta c k^{1-\tau}\right) - 2\eta^2 c C(\lambda_1) \left(1 - \eta c k^{1-\tau}\right) \frac{1}{k^{\frac{\tau}{2}}} - 2\eta^3 c^2 C(\lambda_1) \frac{1}{k^{\frac{\tau}{2}}}.
\]

Now, we set \(\eta = \eta(k, c, \tau) = \frac{1}{4ck^{1-\tau}}.\) Then, plugging \(\eta(k, c, \tau)\) in the above equation:

\[
\eta\|x^1(k)\|^2 \geq \frac{1}{8 \, c \, k^{1-\tau}} - \frac{1}{8 \, c \, k^{2-\tau}} C(\lambda_1) \frac{3}{4} \frac{1}{32 \, c \, k^{3-2\tau}} C(\lambda_1)
\]

\[
= \frac{1}{8 \, c \, k^{1-\tau}} \left(1 - \frac{3 \, C(\lambda_1)}{4k} - \frac{C(\lambda_1)}{4k^{2-\tau}}\right)
\]

\[
\geq \frac{1}{8ck^{1-\tau}} \left(1 - C(\lambda_1) \frac{1}{k}\right).
\]

(86)

The relation (86) is valid if, only if, the condition (85) holds; we now check (85) for \(\eta = \frac{1}{4ck^{1-\tau}}:\)

\[
\left(1 - \eta c \sum_{t=0}^{k-1} \frac{1}{(t+1)^\tau}\right) - \eta c C(\lambda_1) \frac{1}{k^{\frac{\tau}{2}}} \geq \left(1 - \eta c k^{1-\tau}\right) - \eta c C(\lambda_1) \frac{1}{k^{\frac{\tau}{2}}} = 3/4 - C(\lambda_1)/(4k),
\]

and so (86) holds for all \(k \geq C(\lambda_1)/3.\)
We continue by upper bounding the norm of $x^{\Pi}(k)$. From (83):

$$
\|x^{\Pi}(k)\| \geq \alpha_{k-1} = \frac{c}{k^{\tau}},
$$

and so:

$$
\|x^{\Pi}(k)\|^2 \geq \frac{c^2}{k^{2\tau}}. \tag{87}
$$

Finally, combining (86) and (87), and using (81):

$$
\max_{i=1,2} \left( f(x_i(k)) - f^* \right) \geq \frac{1}{2} \sum_{i=1}^2 \left( f(x_i(k)) - f^* \right) \geq \frac{1}{16ck^{1-\tau}} \left( 1 - C(\lambda_1) \frac{1}{k} \right) + \frac{c^2}{2k^{2\tau}}.
$$

We have just proved that, for all $k \geq C(\lambda_1)/3$:

$$
\mathcal{E}(k, R = \sqrt{2}; \tau, c) \geq \frac{1}{16ck^{1-\tau}} \left( 1 - C(\lambda_1) \frac{1}{k} \right) + \frac{c^2}{2k^{2\tau}}, \forall \tau \geq 0, \forall c \in [0, 1/2\sqrt{2}].
$$

Further, for any $k > 2C(\lambda_1)$:

$$
\mathcal{E}(k, R = \sqrt{2}; \tau, c) \geq \frac{1}{32ck^{1-\tau}} + \frac{c^2}{2k^{2\tau}}, \forall \tau \geq 0, \forall c \in [0, 1/2\sqrt{2}].
$$

Finally, taking the infimum over $\tau \geq 0, c \in [0, 1/2\sqrt{2}]$:

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

The result above shows that there exists a class $\mathcal{F}$, namely, the $\mathcal{F}(L = \sqrt{2}, G = 10)$, such that the algorithm in [6] cannot achieve the worst case error $\mathcal{E}(k, R = \sqrt{2}; \tau, c)$ better than $\Omega \left( \frac{1}{k^{2/3}} \right)$, for none of the parameter choices $\tau \geq 0, c \in [0, 1/(2L)]$. In contrast, our algorithm D–NG achieves the worst case error $\mathcal{E}(k, R = \sqrt{2}) = O(\log k/k)$ for any class $\mathcal{F}(L, G)$.

**Demonstration by simulation.** We demonstrate by simulation the obtained lower bound on the worst-case optimality gap with [6]. Figure 1 plots the estimated decay rate $\zeta(\tau)$ of $\mathcal{E}(k, R = \sqrt{2}; \tau, c) \approx \frac{1}{k^{2/3}}$ versus $\tau$, for the constant $c = 1/(2\sqrt{2})$. All the remaining parameters are set as in the theoretical analysis above. For a fixed $k$ and $\tau$, we estimate $\mathcal{E}(k, R = \sqrt{2}; \tau, c)$ as $\mathcal{E}(k, R = \sqrt{2}; \tau, c) = \max_{i=1,2} \{ f^n(x_i(k)) - f^* \}$, where $\eta = 1/(4ck^{1-\tau})$, and $\max_{i=1,2} \{ f^n(x_i(k)) - f^* \}$ is the optimality gap with the algorithm in [6] after $k$ iterations. (The initialization is as explained above in the theoretical analysis, and with the step size $\alpha_t = c/(t+1)^\tau$, $t = 1, ..., k$.)
\( \tau \), we estimate the slope \( \zeta(\tau) \) as:

\[
\zeta(\tau) = \frac{\log_{10}(\mathcal{E}(k_2, R = \sqrt{2}; \tau, c)) - \log_{10}(\mathcal{E}(k_1, R = \sqrt{2}; \tau, c))}{\log_{10}(k_1) - \log_{10}(k_2)},
\]

where \( k_1 = 900 \) and \( k_2 = 1000 \). Figure 1 shows (modulo a possible small offset due to the imperfect estimation in (88) of the asymptotic slope) that, with [6], \( \mathcal{E}(k, R = \sqrt{2}; \tau, c) \) indeed cannot be better than \( \mathcal{E}(k, R = \sqrt{2}; \tau, c) \approx \frac{1}{k^{2/3}} \).

Remark. We provide here an intuitive explanation why D–NG achieves a faster rate under Assumptions 1–3 than the method in [6]. With [6], it can be shown that the global average \( \frac{1}{N} \sum_{i=1}^{N} \overline{x}_i(k) \) of the nodes’ estimates evolves according to a (centralized) ordinary gradient under inexact oracle: \( \overline{x}(k) = \overline{x}(k-1) - \frac{\alpha_{k-1}}{N} \sum_{i=1}^{N} \nabla f_i(x_i(k)) \). The “inexactness” constant is \( \delta_k = L \sum_{i=1}^{N} \| \overline{x}(k) - x_i(k) \|^2 \). If [6] uses the step-size \( \alpha_k = c/(k+1)^{\tau} \), \( k = 0, 1, ..., \tau \in [0, 1] \), it can be shown that \( \delta_k = O\left( \frac{1}{k^{2\tau}} \right) \), and hence, the faster the step-size decays (the larger \( \tau \)), the better (the smaller \( \delta_k \)). On the other hand, the “optimization” process becomes slower when \( \tau \) increases: the ordinary gradient’s optimality gap is \( O\left( \frac{1}{k^{1/\tau}} \right) \), (assuming \( \delta_k = 0 \).) Thus, there is a tradeoff between the disagreement (or \( \delta_k \)) and the “optimization process.” It turns out that this tradeoff gives the optimal \( \tau \) to be strictly positive, so that the algorithm [6] achieves \( O\left( \frac{1}{k^{1/\tau}} \right) \), hence strictly worse than D–NG’s rate \( O(\log k/k) \).