

Distributed Dual Averaging for Convex Optimization under Communication Delays

Konstantinos I. Tsianos and Michael G. Rabbat

Abstract—In this paper we extend and analyze the distributed dual averaging algorithm [1] to handle communication delays and general stochastic consensus protocols. Assuming each network link experiences some fixed bounded delay, we show that distributed dual averaging converges and the error decays at a rate $O(T^{-0.5})$ where T is the number of iterations. This bound is an improvement over [1] by a logarithmic factor in T for networks of fixed size. Finally, we extend the algorithm to the case of using general non-averaging consensus protocols. We prove that the bias introduced in the optimization can be removed by a simple correction term that depends on the stationary distribution of the consensus matrix.

I. INTRODUCTION

In this paper we extend and analyze the distributed dual averaging algorithm [1]. We employ the fixed delay model introduced in [2] and show that distributed dual averaging still converges in the presence of finite and fixed communication delays. In addition, using a different bounding technique than [1], for a fixed network size, we improve on the convergence rate in terms of number of iterations by removing a logarithmic factor. Finally, we analyze the case where a general (non-averaging) consensus protocol is used. We explain and illustrate in simulation how the use of non-doubly stochastic consensus matrices biases the optimization. The issue is not however essential and we prove that a simple correction term removes the bias.

Over the last few years, the dramatic increase in available data has made imperative the use of parallel and distributed algorithms for solving large scale optimization and machine learning problems (see for example [3], [4]). Among the numerous possible choices, fully distributed algorithms combining some version of local optimization with a distributed consensus protocol are an appealing option [1], [4]–[7]. With such an approach, all computing nodes have the same role in the optimization procedure. We thus eliminate single points of failure and increase robustness. This is important in large scale systems where machines may go down during the computation. We also have increased flexibility at adding more computational resources. At the same time, these algorithms are simple to implement and avoid the bookkeeping needed for more intricate hierarchical algorithms.

The main focus of this paper is the analysis and extension of the distributed dual averaging algorithm. For practical application, it is important to know how the algorithm behaves

in the presence of communication delays. For example, in a network with 1 Gigabit per second ethernet, for a small machine learning problem we may need to send messages of size 1Mbyte per iteration which translates to a transmission delay of 8 milliseconds per message. For a modern processor using some fast local optimization routine (e.g., stochastic gradient descent [8]), 8 milliseconds is enough time to perform multiple iterations of computation, and the communication delay when exchanging information over the network is not negligible.

We employ a delay model introduced in [2] and show that under finite directed edge delays the algorithm will still converge to the optimum. We prove that despite the presence of delays, the error decays at a rate $O(T^{-0.5})$ where T is the number of iterations. We also show that the dependence of the error on the cumulative total edge delay b , is $O(b^{\frac{3}{2}})$. In addition, using a different bounding technique, we can improve on the rate given in [1] by a factor $O(\log T)$ if we keep the network size n fixed. Moreover, as explained in [9] for some network topologies it may not be possible to use a doubly stochastic consensus protocol. We thus generalize the algorithm for cases where a general non-averaging consensus protocol is used. The non-uniform stationary distribution of the consensus matrix causes an undesired bias in the objective function. This issue has been mentioned in previous work (e.g., [6]). Here we exhibit the effect in simulation. We prove however that the issue is not essential. If we know the stationary distribution of the consensus matrix, a simple re-weighting of the gradients removes the bias.

The rest of the paper is organized as follows. In Section II we briefly review the standard distributed dual averaging algorithm to keep the paper self-contained. Section III contains our analysis and extensions. After introducing the delay model, we describe the necessary modification and provide a complete convergence proof in the presence of delays using an arbitrary consensus matrix. Comments on the implications of our analysis and some illustrative simulations are included in Section IV. The paper concludes with a summary and discussion of future work in Section V.

II. DISTRIBUTED DUAL AVERAGING

To make the paper self-contained, we provide some necessary background on the distributed dual averaging algorithm. For more details consult [1]. Suppose we are given an undirected network $G = (V, E)$ of $|V| = n$ compute nodes. Each node i knows a convex function $f_i(x) : \mathbb{R}^d \rightarrow \mathbb{R}$. Our

K. I. Tsianos is a PhD candidate at the Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec H3A 2A7, Canada, konstantinos.tsianos@mail.mcgill.ca

M.G. Rabbat is an Assistant Professor at the Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec H3A 2A7, Canada, michael.rabbat@mcgill.ca

goal is to solve the following minimization problem:

$$\text{minimize } f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \quad (1)$$

$$\text{subject to } x \in \mathcal{X} \quad (2)$$

where \mathcal{X} is a convex set. We assume that each f_i is convex and L -Lipschitz continuous with respect to the same norm $\|\cdot\|$; i.e., $|f_i(x) - f_i(y)| \leq L\|x - y\|, \forall x, y \in \mathcal{X}$. As a consequence, for any $x \in \mathcal{X}$ and any subgradient $g_i \in \partial f_i(x)$ we have $\|g_i\|_* \leq L$ where $\|v\|_* = \sup_{\|u\|=1} \langle u, v \rangle$ is the dual norm.

Let us select a 1-strongly convex proximal function $\psi : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\psi(x) \geq 0$ and $\psi(0) = 0$. Also select a non-increasing sequence of positive step sizes $\{a(t)\}_{t=0}^\infty$ and a doubly stochastic matrix P respecting the structure of G in the sense that $P_{ij} > 0$ only if $i = j$ or $(i, j) \in E$. The distributed dual averaging algorithm repeats, for each node i in discrete steps t , the following updates:

$$z_i(t+1) = \sum_{j=1}^n P_{ij} z_j(t) + g_i(t) \quad (3)$$

$$x_i(t+1) = \Pi_{\mathcal{X}}^\psi(z_i(t+1), a(t)) \quad (4)$$

where the projection operator $\Pi_{\mathcal{X}}^\psi(\cdot, \cdot)$ is defined as

$$\Pi_{\mathcal{X}}^\psi(z, a) = \underset{x \in \mathcal{X}}{\operatorname{argmin}} \{ \langle z, x \rangle + \frac{1}{a} \psi(x) \}. \quad (5)$$

In (3),(4) x_i is the local estimate at node i and z_i is a dual variable maintaining an accumulated subgradient. To update z_i , at each iteration each node needs to collect the z -values of its neighbours, form a convex combination of the received information and add its local most recent subgradient $g_i(t)$. In [1] it is proven that using this algorithm, the local running average $\hat{x}_i(T) = \frac{1}{T} \sum_{t=1}^T x_i(t)$ converges to the optimum. Specifically, keeping track of the average cumulative gradient $\bar{z}(t) = \frac{1}{n} \sum_{i=1}^n z_i(t)$, the following basic theorem is proven.

Theorem 1 (Basic Result): Let the sequences $\{x_i(t)\}_{t=0}^\infty$ and $\{z_i(t)\}_{t=0}^\infty$ be generated by the updates (3),(4) using a non-increasing step size sequence $\{a(t)\}_{t=0}^\infty$. For any $x^* \in \mathcal{X}$ and for every node $i \in V$ we have:

$$\begin{aligned} f(\hat{x}_i(T)) - f(x^*) &\leq \frac{1}{T a(T)} \psi(x^*) + \frac{L^2}{2T} a(t-1) \\ &+ \frac{2L}{nT} \sum_{t=1}^T \sum_{j=1}^n a(t) \|\bar{z}(t) - z_j(t)\|_* \\ &+ \frac{L}{T} \sum_{t=1}^T a(t) \|\bar{z}(t) - z_i(t)\|_*. \quad (6) \end{aligned}$$

The first two terms in (6) are standard terms in subgradient optimization algorithms while the last two terms capture the network error due to the discrepancy between the local gradients and the true average gradient. By bounding the network error $\|\bar{z}(t) - z_i(t)\|_*$, we can derive convergence rates that depend on the network characteristics.

In the following section we prove that Theorem 1 holds also when there is delayed communication between the

compute nodes. We show this by employing the delay model introduced in [2]. We also prove that with a simple re-weighting of the local gradients $g_i(t)$ we can relax the assumption that P is doubly stochastic without biasing the optimization.

III. ANALYSIS WITH DELAYS

In this section we extend Theorem 1 to the case where each directed communication link in the network experiences a fixed amount of delay. We first briefly describe the delay model and then provide a convergence proof for distributed dual averaging.

A. Fixed Delay Model

Assume that for each directed link (i, j) of G every message from i is delayed by b_{ij} time units before arriving at j . We model this delay by adding b_{ij} delay nodes in the network acting as relays between i and j . In total we have $b = \sum_{(i,j) \in E} b_{ij}$ delay nodes. In [2] we describe how to construct a stochastic matrix Q in the augmented space of $n + b$ nodes starting from a doubly stochastic P . Matrix Q is responsible for communicating information between delay and compute nodes so that each compute node still forms a convex combination of the incoming messages. Matrix Q has a stationary distribution π which is not uniform and depends on both P and the edge delays. See [2] for an exact characterization of π .

B. Convergence with Fixed Delays and General Consensus Matrices

To model communication delays we introduce b delay nodes in the network G . We associate with each delay node a function $f_i(x) = 0, i = n+1, \dots, n+b$ so that the subgradients on the delay nodes $g_i(t), i > 0$ are zero as well. For the rest we also assume that the dual variables are initialized to zero i.e., $z_i(0) = 0$. To analyze distributed dual averaging with delays, we use Q as a transition matrix instead of P in equation (3). Matrix Q is not doubly stochastic and has a stationary distribution π which is not uniform. For reasons to be explained in the sequel, we introduce for each compute node $i \in V$ a weight $c_i = \frac{1}{\pi_i n}$ and replace update (3) of the original algorithm by

$$z_i(t+1) = \sum_{j=1}^{n+b} Q_{ij} z_j(t) + c_i g_i(t), \quad i = 1, \dots, n+b. \quad (7)$$

We begin by re-defining the auxiliary sequences:

$$\bar{z}(t) = \sum_{i=1}^{n+b} \pi_i z_i(t), \quad y(t) = \Pi_{\mathcal{X}}^\psi(\bar{z}(t), a). \quad (8)$$

The weighed average cumulative gradient \bar{z} evolves as follows:

$$\bar{z}(t+1) = \sum_{i=1}^{n+b} \pi_i z_i(t+1)$$

$$\begin{aligned}
&= \sum_{i=1}^{n+b} \pi_i \left(\sum_{j=1}^{n+b} Q_{ij} z_j(t) + c_i g_i(t) \right) \\
&= \sum_{j=1}^{n+b} z_j(t) \left(\sum_{i=1}^{n+b} \pi_i Q_{ij} \right) + \sum_{i=1}^{n+b} \pi_i c_i g_i(t). \quad (9)
\end{aligned}$$

Since π is a left eigenvector of Q , $\pi^T Q = \pi^T$ implying that $\sum_{i=1}^{n+b} \pi_i Q_{ij} = \pi^T Q_{:,j} = \pi_j$. Using this fact and noting that $\pi_i c_i = \frac{1}{n}$,

$$\bar{z}(t+1) = \sum_{j=1}^{n+b} z_j(t) \pi_j + \sum_{i=1}^{n+b} \frac{1}{n} g_i(t), \quad (10)$$

or finally

$$\bar{z}(t+1) = \bar{z}(t) + \frac{1}{n} \sum_{i=1}^n g_i(t) \quad (11)$$

since $g_i(t) = 0$ for $i > n$. Using the last recursion, with $z_i(0) = 0$, we rewrite (8) as

$$\bar{z}(t) = \frac{1}{n} \sum_{s=1}^{t-1} \sum_{i=1}^n g_i(s), \quad y(t) = \Pi_{\mathcal{X}}^{\psi} \left(\frac{1}{n} \sum_{s=1}^{t-1} \sum_{i=1}^n g_i(s), a \right). \quad (12)$$

Next, we state three lemmas which are proved in [1] and remain unaltered in our modified setup.

Lemma 1: Let $\{g(t)\}_{t=1}^{\infty} \subset \mathbb{R}^d$ be an arbitrary sequence of vectors, $\{a(t)\}_{t=0}^{\infty}$ be a non-increasing sequence and consider the sequence

$$x(t+1) = \Pi_{\mathcal{X}}^{\psi} \left(\sum_{s=1}^t g(s), a(t) \right). \quad (13)$$

For any $x^* \in \mathcal{X}$ we have

$$\sum_{t=1}^T \langle g(t), x(t) - x^* \rangle \leq \frac{1}{2} \sum_{t=1}^T a(t-1) \|g(t)\|_*^2 + \frac{1}{a(T)} \psi(x^*). \quad (14)$$

Lemma 2: Consider the sequences $\{x_i(t)\}_{t=1}^{\infty}$, $\{z_i(t)\}_{t=0}^{\infty}$ and $\{y_i(t)\}_{t=0}^{\infty}$ defined in (4), (7) and (8). For each $i = 1, \dots, n+b$ and any $x^* \in \mathcal{X}$ we have

$$\begin{aligned}
\sum_{t=1}^T f(x_i(t)) - f(x^*) &\leq \sum_{t=1}^T f(y(t)) - f(x^*) \\
&\quad + L \sum_{t=1}^T a(t) \|\bar{z}(t) - z_i(t)\|_* \quad (15)
\end{aligned}$$

and, with $\hat{y}(T) = \frac{1}{T} \sum_{t=1}^T y(t)$ and $\hat{x}_i(T) = \frac{1}{T} \sum_{t=1}^T x_i(t)$,

$$\begin{aligned}
f(\hat{x}_i(T)) - f(x^*) &\leq f(\hat{y}(T)) - f(x^*) \\
&\quad + \frac{L}{T} \sum_{t=1}^T a(t) \|\bar{z}(t) - z_i(t)\|_*. \quad (16)
\end{aligned}$$

Lemma 3: For an arbitrary pair $u, v \in \mathbb{R}^d$, we have

$$\|\Pi_{\mathcal{X}}^{\psi}(u, a) - \Pi_{\mathcal{X}}^{\psi}(v, a)\| \leq \|u - v\|_*. \quad (17)$$

At this point we have all we need to proceed with the convergence proof. Since $f(x)$ is convex, for any $x^* \in \mathcal{X}$ we have

$$f(\hat{x}_i(T)) - f(x^*) \leq \frac{1}{T} \sum_{t=1}^T [f(x_i(t)) - f(x^*)]. \quad (18)$$

Using Lemma 2 we obtain

$$\begin{aligned}
f(\hat{x}_i(T)) - f(x^*) &\leq \frac{1}{T} \sum_{t=1}^T [f(y(t)) - f(x^*)] \\
&\quad + \frac{L}{T} \sum_{t=1}^T a(t) \|\bar{z}(t) - z_i(t)\|_*. \quad (19)
\end{aligned}$$

To bound the first term in (19), we add and subtract $\sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} f_i(x_i(t))$ and observe that $\sum_{i=1}^n \frac{1}{n} f(x^*) = f(x^*)$ to get

$$\begin{aligned}
\sum_{t=1}^T [f(y(t)) - f(x^*)] &\leq \sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} [f_i(x_i(t)) - f(x^*)] \\
&\quad + \sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} |f_i(y(t)) - f_i(x_i(t))|. \quad (20)
\end{aligned}$$

Using convexity of each $f_i(x)$ with $g_i(t) \in \partial f_i(x_i(t))$,

$$\begin{aligned}
\sum_{t=1}^T f(y(t)) - f(x^*) &\leq \sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} \langle g_i(t), y(t) - x^* \rangle \\
&\quad + \sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} \langle g_i(t), x_i(t) - y(t) \rangle \\
&\quad + \sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} |f_i(y(t)) - f_i(x_i(t))|. \quad (21)
\end{aligned}$$

Focusing on the first term of (21) and recalling the definition (12) of $y(t)$ we have

$$\begin{aligned}
\sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} \langle g_i(t), y(t) - x^* \rangle &= \sum_{t=1}^T \left\langle \sum_{i=1}^n \frac{1}{n} g_i(t), y(t) - x^* \right\rangle \\
&= \sum_{t=1}^T \left\langle \sum_{i=1}^n \frac{1}{n} g_i(t), \Pi_{\mathcal{X}}^{\Psi} \left(\sum_{s=1}^{t-1} \sum_{i=1}^n \frac{1}{n} g_i(s), a(t) \right) - x^* \right\rangle. \quad (22)
\end{aligned}$$

With $\sum_{i=1}^n \frac{1}{n} g_i(s)$ playing the role of the arbitrary vector sequence, the last equation can be bounded using Lemma 1 after applying the Cauchy-Schwartz inequality and remembering that $\|g_i\|_* \leq L$:

$$\begin{aligned}
&\sum_{t=1}^T \sum_{i=1}^n \frac{1}{n} \langle g_i(t), y(t) - x^* \rangle \\
&\leq \frac{1}{2} \sum_{t=1}^T a(t-1) \left\| \sum_{i=1}^n \frac{1}{n} g_i(t) \right\|_*^2 + \frac{1}{a(T)} \Psi(x^*) \\
&\leq \frac{L^2}{2} \sum_{t=1}^T a(t-1) + \frac{1}{a(T)} \Psi(x^*). \quad (23)
\end{aligned}$$

For the last two terms in (21) we use L -Lipshitz continuity of $f(x)$ and Lemma 3 to get after some calculations that

$$\begin{aligned}
& \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^n \left(\langle g_i(t), x_i(t) - y(t) \rangle + |f_i(y(t)) - f_i(x_i(t))| \right) \\
& \leq \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^n \left(L \|y(t) - x_i(t)\| + |f_i(y(t)) - f_i(x_i(t))| \right) \\
& \leq \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^n \left(L \|y(t) - x_i(t)\| + L \|y(t) - x_i(t)\| \right) \\
& \leq \frac{2L}{n} \sum_{t=1}^T \sum_{i=1}^n \left\| \Pi_{\mathcal{X}}^{\Psi}(\bar{z}(t), a(t)) - \Pi_{\mathcal{X}}^{\Psi}(z_i(t), a(t)) \right\| \\
& \leq \frac{2L}{n} \sum_{t=1}^T \sum_{i=1}^n a(t) \|\bar{z}(t) - z_i(t)\|_*. \tag{24}
\end{aligned}$$

Going back to (19), we replace the bounds we derived for the first and last two terms to retrieve exactly the bound of Theorem 1 for the modified version of the algorithm:

$$\begin{aligned}
f(\hat{x}_i(T)) - f(x^*) & \leq \frac{L^2}{2T} \sum_{t=1}^T a(t-1) + \frac{1}{Ta(T)} \Psi(x^*) \\
& \quad + \frac{2L}{nT} \sum_{t=1}^T \sum_{i=1}^n a(t) \|\bar{z}(t) - z_i(t)\|_* \\
& \quad + \frac{L}{T} \sum_{t=1}^T a(t) \|\bar{z}(t) - z_i(t)\|_*. \tag{25}
\end{aligned}$$

Next we need to bound the network error $\|\bar{z}(t) - z_i(t)\|_*$. If we define for convenience $\Phi(t, s) = Q^{t-s+1}$ and back-substitute in the recursion (7) we can see that

$$z_i(t) = c_i \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} [\Phi(t-1, s)]_{ji} \cdot g_j(s-1) + c_i g_i(t-1). \tag{26}$$

Recalling the definition (8) for $\bar{z}(t)$, after some term rearrangements and using the facts that $\sum_{k=1}^{n+b} [\Phi(t-1, s)]_{jk} = 1$, $c_i = \frac{1}{\pi_i n}$ and $g_i(t) = 0, i > n$, we see that

$$\begin{aligned}
\bar{z}(t) - z_i(t) & = \sum_{k=1}^{n+b} \pi_k z_k(t) - z_i(t) \\
& = \sum_{k=1}^{n+b} \pi_k \left[c_k \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} [\Phi(t-1, s)]_{jk} \right. \\
& \quad \cdot g_j(s-1) + c_k g_k(t-1) \left. \right] \\
& \quad - c_i \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} [\Phi(t-1, s)]_{ji} \cdot g_j(s-1) - c_i g_i(t-1)
\end{aligned}$$

$$\begin{aligned}
& = \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} \frac{1}{n} g_j(s-1) \sum_{k=1}^{n+b} [\Phi(t-1, s)]_{jk} + \sum_{k=1}^{n+b} \frac{1}{n} g_k(t-1) \\
& \quad - \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} c_i [\Phi(t-1, s)]_{ji} \cdot g_j(s-1) - c_i g_i(t-1) \\
& = \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} \left(\frac{1}{n} - c_i [\Phi(t-1, s)]_{ij} \right) g_j(s-1) \\
& \quad + \frac{1}{n} \sum_{k=1}^{n+b} g_k(t-1) - c_i g_i(t-1) \\
& = c_i \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} \left(\pi_i - [\Phi(t-1, s)]_{ij} \right) g_j(s-1) \\
& \quad + \frac{1}{n} \sum_{k=1}^n [g_k(t-1) - c_i g_i(t-1)]. \tag{27}
\end{aligned}$$

Taking norms on both side and using the bound L on gradient magnitudes, we obtain

$$\begin{aligned}
& \|\bar{z}(t) - z_i(t)\|_* \\
& \leq c_i \sum_{s=1}^{t-1} \sum_{j=1}^{n+b} \left| \pi_i - [\Phi(t-1, s)]_{ij} \right| \cdot \|g_j(s-1)\|_* \\
& \quad + \frac{1}{n} \sum_{k=1}^n \|g_k(t-1) - c_i g_i(t-1)\|_* \\
& \leq L c_i \sum_{s=1}^{t-1} \left\| \pi - [\Phi(t-1, s)]_{i,:} \right\|_1 + (1 + c_i)L. \tag{28}
\end{aligned}$$

The last expression reduces to exactly the bound obtained in Theorem 2 in [1] if Q is doubly stochastic and there is no delays since in that case $\pi_i = \frac{1}{n}$ and $c_i = 1$.

Instead of using the bounding technique of [1], we provide a different bound here that is tighter in the number of iterations. From [2] we know that for all i

$$\left\| \pi - [\Phi(t-1, s)]_{i,:} \right\|_1 = 2 \left\| \pi - Q_{i,:}^{t-s+1} \right\|_{TV} \leq \sqrt{\frac{\lambda_2^{t-s+1}}{\pi_i}} \tag{29}$$

where $\|\cdot\|_{TV}$ denotes total variation distance and λ_2 is the second largest eigenvalue of the *lazy additive reversibilization* of Q (see also [2] and [10]). Using this result and applying the formula for a finite geometric sum (since $\lambda_2 < 1$), we bound the network error by:

$$\begin{aligned}
\|\bar{z}(t) - z_i(t)\|_* & \leq L c_i \sum_{s=1}^{t-1} \sqrt{\frac{\lambda_2^{t-s+1}}{\pi_i}} + (1 + c_i)L \\
& = \frac{L c_i}{\sqrt{\pi_i}} \sum_{s=1}^{t-1} \left(\sqrt{\lambda_2} \right)^{t-s+1} + (1 + c_i)L \\
& = \frac{L c_i}{\sqrt{\pi_i}} \sum_{s=2}^t \left(\sqrt{\lambda_2} \right)^s + (1 + c_i)L
\end{aligned}$$

$$\begin{aligned}
&= \frac{Lc_i}{\sqrt{\pi_i}} \frac{(\sqrt{\lambda_2})^2 - (\sqrt{\lambda_2})^{t+1}}{1 - \sqrt{\lambda_2}} + (1 + c_i)L \\
&\leq \frac{Lc_i}{\sqrt{\pi_i}} \frac{\lambda_2}{1 - \sqrt{\lambda_2}} + (1 + c_i)L \triangleq K_i. \quad (30)
\end{aligned}$$

This bound is tighter than the one obtained in [1] since for fixed n it is constant and does not increase logarithmically with time. This comes at the expense of a dependence $O(\sqrt{n})$ on the network size instead of $O(\log \sqrt{n})$ as shown in the next section. Replacing the network error in the main bound (25) we have shown the following.

Theorem 2: Let the sequences $\{x_i(t)\}_{t=0}^{\infty}$ and $\{z_i(t)\}_{t=0}^{\infty}$ be generated by the updates (7),(4) using a non-increasing step size sequence $\{a(t)\}_{t=0}^{\infty}$. The modified distributed dual averaging algorithm will converge to the optimum for any distribution of fixed edge delays and any stochastic communication protocol P .

IV. COMMENTS AND IMPLICATIONS OF THE ANALYSIS

The convergence proof of the previous section leads to some important conclusions depending on the situation in which we apply distributed dual averaging. We make comments about specific cases in this section.

A. Doubly stochastic P , fixed edge delays

In this case, Q is a stochastic matrix whose stationary distribution π assigns equal probabilities to all the compute nodes as is shown in [2]. We can derive a precise expression for the convergence rate by first recalling from [2] that $\pi_i \geq \frac{1}{n+b}$, $i \in V$. We use this fact to get

$$\begin{aligned}
K_i &\leq \frac{L(n+b)^{\frac{3}{2}}}{n} \frac{\lambda_2}{1 - \sqrt{\lambda_2}} + \left(1 + \frac{n+b}{n}\right)L \\
&= \underbrace{\left(\frac{(n+b)^{\frac{3}{2}}}{n} \frac{\lambda_2}{1 - \sqrt{\lambda_2}} + \left(1 + \frac{n+b}{n}\right)\right)}_K L \triangleq KL. \quad (31)
\end{aligned}$$

By replacing the bound KL from (31) in (25), using the fact that $\sum_{t=1}^T t^{-0.5} \leq 2\sqrt{T} - 1$ and selecting $a(t) = \frac{R}{L\sqrt{t}}$, after some algebraic manipulations we prove the following.

Theorem 3: Under the conditions of Theorem 1, assuming $\psi(x^*) \leq R^2$, using a step size sequence $a(t) = \frac{R}{L\sqrt{t}}$ and assuming that P is doubly stochastic and we have fixed edge delays,

$$f(\hat{x}_i(T)) - f(x^*) \leq (1 + 3K) \frac{2RL}{\sqrt{T}}. \quad (32)$$

The influence of the network topology is represented by λ_2 in K . Moreover, the effect of communication delays as well as the size of the network n are captured by the dominant term $\frac{(n+b)^{\frac{3}{2}}}{n}$ in K where b is the total amount of delay cumulatively on all the links. Figure 1 illustrates the effect of delays in a toy example. We create a random network topology of 10 nodes. Each node i holds a simple quadratic: $f_i(x) = (x - \mathbf{1}_i)^T(x - \mathbf{1}_i)$, $x \in \mathbb{R}^5$. For this problem we can compute easily the exact minimizer $x^* = 5.5 \cdot \mathbf{1}$ with $f(x^*) = 412.5$. The blue curve shows the progress of the

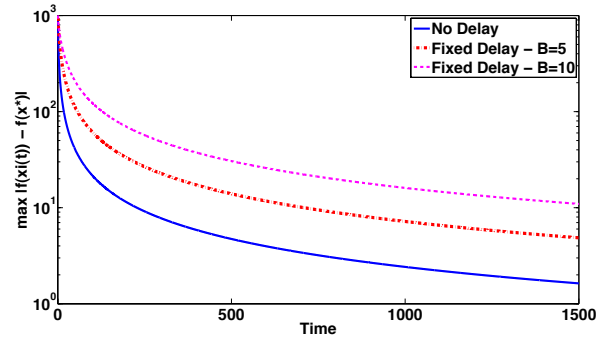


Fig. 1. Illustration of the effect of fixed edge delays on distributed dual averaging. Blue curve: Performance without delays. Red curve: Performance using a fixed delay up to $B = 5$ time steps per directed link. Purple curve: Performance using a fixed delay up to $B = 10$ time steps per directed link.

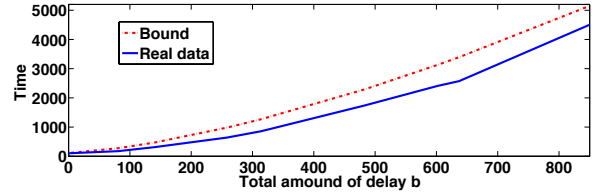


Fig. 2. Blue curve: Time it takes for a network of 10 nodes to reduce the objective function error $\max_i |f(x_i(t)) - f(x^*)|$ below 180 as we increase the total amount of delay b in the network. The theoretical bound (red curve) is in the right order of magnitude.

minimization with standard distributed dual averaging as the evolution of the maximum error $\max_i |f(x_i(t)) - f(x^*)|$. The red curve shows that the algorithm is slowed down when we inject a random fixed delay up to $B = 5$ on each directed link (i, j) . The purple curve allows for a maximum possible delay $B = 10$.

To verify that the dependence in the total amount of delay appearing in the bound (32) is in the right order of magnitude, in figure 2 we record the amount of time it takes to bring the optimization error below a threshold for varying amounts of delay. Specifically, in our problem with 10 nodes, we measure the time until $\max_i |f(x_i(t)) - f(x^*)| < 180$. We also plot the dominant term in our theoretical bound as $O\left(\frac{(n+b)^{\frac{3}{2}}}{n}\right) = 2\frac{(n+b)^{\frac{3}{2}}}{n} + 102$. The bound and simulation are well matched.

B. Not doubly stochastic P , no delays

In this case, $Q = P$ with a stationary distribution π that is not uniform. Going back to the definition (8) of the true average cumulative gradient $\bar{z}(t)$, we see that certain gradients get more weights than others which is equivalent to rescaling each component $f_i(x)$ of the objective function $f(x)$ by a factor π_i . As a result, we still converge but end up minimizing the biased objective $\tilde{f}(x) = \sum_{i=1}^n \pi_i f_i(x)$. This bias can be removed if we multiply each local gradient $g_i(t)$ by $c_i = \frac{1}{\pi_i n}$. This situation is illustrated in figure 3 where we minimize the same sum of ten quadratics as before. Instead of a consensus protocol with a uniform stationary distribution $0.1 \cdot \mathbf{1}$, we generate a stochastic matrix with stationary distribution

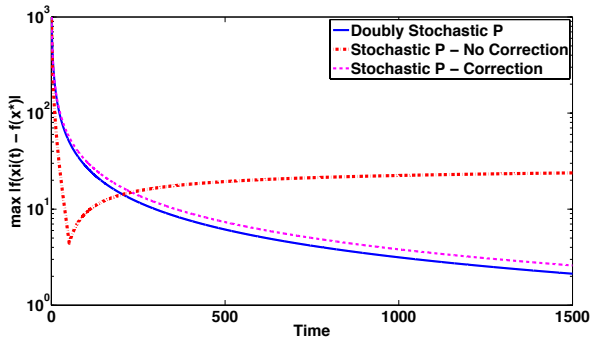


Fig. 3. Illustration of optimization bias with non-doubly stochastic matrices. The blue curve shows progress of distributed dual averaging with a doubly stochastic consensus matrix P . Choosing a random stochastic P that has a non-uniform stationary distribution we end up solving a biased problem with a different optimum shown by the red curve. Applying the suggested correction c_i in the gradient weights we remove the bias as shown in the purple curve.

$\pi = (0.06, 0.04, 0.11, 0.10, 0.09, 0.04, 0.10, 0.21, 0.14, 0.11)$ giving significant weight to node 8. As a result, the optimal value is biased to be $x_{biased}^* = 6.2847 \cdot \mathbf{1}$ instead of $x^* = 5.5 \cdot \mathbf{1}$ with $f(x_{biased}^*) = 443.286$ instead of the correct value $f(x^*) = 412.5$. By employing the suggested correction in the gradient weights, we remove the bias and solve the original problem as the purple curve shows.

As a last comment relating to the previous case, if P is doubly stochastic but we have delays, Q will not be doubly stochastic, but the c_i multipliers are not necessary since the stationary distribution assigns equal probabilities π_V to all compute nodes. Since those probabilities are not $\frac{1}{n}$ the side effect is a rescaling of the objective function by π_V which however does not change the location of the optimum x^* .

C. Not doubly stochastic P , fixed edge delays

In this case, we can still follow the procedure in [2] to construct a stochastic matrix Q to model the delays. However, the stationary distribution of Q will not assign equal probabilities to the compute nodes anymore and [2] does not provide a nice closed form expression for π . To use the multipliers c_i to remove the bias we require the knowledge of π . If the full matrix P is known in advance we can compute the stationary distribution π of P and the c_i s numerically.

V. SUMMARY AND FUTURE WORK

We analyze and extend distributed dual averaging [1]. For practical problems we expect to experience non-negligible communication delays which the original algorithm does not take into account. We employ the fixed communication delay model appearing in [2] and prove that the presence of fixed edge delays does not hurt the ability of the algorithm to converge and reduce the error at a rate $O(T^{-0.5})$ where T is the number of iterations. If we have a total of b units of delay cumulatively on the network edges, the rate of convergence is slower by a factor no more than $O(b^{\frac{3}{2}})$. As a byproduct of our analysis, we show how to remove a logarithmic factor

$O(\log T)$ from the convergence rate presented in [1] at the expense of an $O(\sqrt{n})$ dependence in the network size instead of $O(\log \sqrt{n})$. Finally, we investigate the subtle issue of involuntarily introducing a bias in the optimization if we do not use a doubly stochastic communication matrix. We show that the issue is not essential. By using any stochastic matrix P with a stationary distribution π , we can still achieve convergence and remove the bias if we re-weight each gradient $g_i(t)$ by a factor $c_i = \frac{1}{\pi_i n}$.

In the future we would like to extend our results to a more realistic delay model relaxing the assumption that the delay per edge is fixed. A random delay model is presented in [2] as well. In that model every transmission over a link (i, j) experiences a random delay that is assumed finite. It is possible to show that we can still achieve consensus with the random delay model, although not average consensus in general. In that model however, it is not clear what the stationary distribution is going to be. We would like to prove convergence of distributed dual averaging under the random delay model and also avoid introducing a bias in the optimization. In addition, besides dual averaging there is significant work in primal averaging algorithms such as [5], [6]. It would be interesting to obtain results similar to the ones presented here for those algorithms.

REFERENCES

- [1] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, 2011.
- [2] K. I. Tsianos and M. G. Rabbat, "Distributed consensus and optimization under communication delays," in *49th Allerton*, 2011.
- [3] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up Machine Learning, Parallel and Distributed Approaches*. Cambridge University Press, 2011.
- [4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [5] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, January 2009.
- [6] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2011.
- [7] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Control and Optimization*, vol. 20, no. 3, 2009.
- [8] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of the 19th International Conference on Computational Statistics*, Y. Lechevallier and G. Saporta, Eds., Paris, France, August 2010, pp. 177–187.
- [9] B. Gharesifard and J. Cortes, "When does a digraph admit a doubly stochastic adjacency matrix?" in *Proceedings of the American Control Conference*, Baltimore, Maryland, 2010, pp. 2440–2445.
- [10] J. A. Fill, "Eigenvalue bounds on convergence to stationarity for non reversible markov chains, with an application to the exclusion process," *The Annals of Applied Probability*, vol. 1, no. 1, pp. 62–87, 1991.