Distributed Dual Averaging in Networks

John C. Duchi[1] **Alekh Agarwal**[1] **Martin J. Wainwright**[1,2]
Department of Electrical Engineering and Computer Science[1] and Department of Statistics[2]
University of California, Berkeley
Berkeley, CA 94720-1776
{jduchi,alekh,wainwrig}@eecs.berkeley.edu

Abstract

The goal of decentralized optimization over a network is to optimize a global objective formed by a sum of local (possibly nonsmooth) convex functions using only local computation and communication. We develop and analyze distributed algorithms based on dual averaging of subgradients, and provide sharp bounds on their convergence rates as a function of the network size and topology. Our analysis clearly separates the convergence of the optimization algorithm itself from the effects of communication constraints arising from the network structure. We show that the number of iterations required by our algorithm scales inversely in the spectral gap of the network. The sharpness of this prediction is confirmed both by theoretical lower bounds and simulations for various networks.

1 Introduction

Network-structured optimization problems arise in a variety of application domains within the information sciences and engineering. A canonical example that arises in machine learning is the problem of minimizing a loss function averaged over a large dataset (e.g. [16, 17]). With terabytes of data, it is desirable (even necessary) to assign smaller subsets of the data to different processors, and the processors must communicate to find parameters that minimize the loss over the entire dataset. Problems such as multi-agent coordination, estimation problems in sensor networks, and packet routing also are all naturally cast as distributed convex minimization [1, 13, 24]. The seminal work of Tsitsiklis and colleagues [22, 1] analyzed algorithms for minimization of a smooth function f known to several agents while distributing processing of components of the parameter vector $x \in \mathbb{R}^n$. More recently, a few researchers have shifted focus to problems in which each processor locally has its own convex (potentially non-differentiable) objective function [18, 15, 21, 11].

In this paper, we provide a simple new subgradient algorithm for distributed constrained optimization of a convex function. We refer to it as a *dual averaging subgradient method*, since it is based on maintaining and forming weighted averages of subgradients throughout the network. This approach is essentially different from previously developed distributed subgradient methods [18, 15, 21, 11], and these differences facilitate our analysis of network scaling issues—how convergence rates depend on network size and topology. Indeed, the second main contribution of this paper is a careful analysis that demonstrates a close link between convergence of the algorithm and the underlying spectral properties of the network. The convergence rates for a different algorithm given by the papers [18, 15] grow exponentially in the number of nodes n in the network. Ram et al. [21] provide tighter analysis that yields convergence rates that scale cubically in the network size, but are independent of the network topology. Consequently, their analysis does not capture the intuition that distributed algorithms should converge faster on "well-connected" networks—expander graphs being a prime example—than on poorly connected networks (e.g., chains or cycles). Johansson et al. [11] analyze a low communication peer-to-peer protocol that attains rates dependent on network structure. However, in their algorithm only one node has a current parameter value, while all nodes in our algorithm maintain good estimates of the optimum at all times. This is important in online

or streaming problems where nodes are expected to act or answer queries in real-time. In additional comparison to previous work, our analysis yields network scaling terms that are often substantially sharper. Our development yields an algorithm with convergence rate that scales inversely in the spectral gap of the network. By exploiting known results on spectral gaps for graphs with n nodes, we show that our algorithm obtains an ϵ-optimal solution in $\mathcal{O}(n^2/\epsilon^2)$ iterations for a single cycle or path, $\mathcal{O}(n/\epsilon^2)$ iterations for a two-dimensional grid, and $\mathcal{O}(1/\epsilon^2)$ iterations for a bounded degree expander graph. Simulation results show excellent agreement with these theoretical predictions.

2 Problem set-up and algorithm

In this section, we provide a formal statement of the distributed minimization problem and a description of the distributed dual averaging algorithm.

Distributed minimization: We consider an optimization problem based on functions that are distributed over a network. More specifically, let $G = (V, E)$ be an undirected graph over the vertex set $V = \{1, 2, \ldots, n\}$ with edge set $E \subset V \times V$. Associated with each $i \in V$ is convex function $f_i : \mathbb{R}^d \to \mathbb{R}$, and our overarching goal is to solve the constrained optimization problem $\min_{x \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^n f_i(x)$, where $\mathcal{X}$ is a closed convex set. Each function f_i is convex and hence subdifferentiable, but need not be smooth. We assume without loss of generality that $0 \in \mathcal{X}$, since we can simply translate $\mathcal{X}$. Each node $i \in V$ is associated with a separate agent, and each agent i maintains its own parameter vector $x_i \in \mathbb{R}^d$. The graph G imposes communication constraints on the agents: in particular, agent i has local access to only the objective function f_i and can communicate directly only with its immediate neighbors $j \in N(i) := \{j \in V \mid (i, j) \in E\}$.

A concrete motivating example for these types of problems is the machine learning scenario described in Section 1. In this case, the set $\mathcal{X}$ is the parameter space of the learner. Each function f_i is the empirical loss over the subset of data assigned to processor i, and the average f is the empirical loss over the entire dataset. We use cluster computing as our model, so each processor is a node in the cluster and the graph G contains edges between processors connected with small latencies; this setup avoids communication bottlenecks of architectures with a centralized master node.

Dual averaging: Our algorithm is based on a dual averaging algorithm [20] for minimization of a (potentially nonsmooth) convex function f subject to the constraint that $x \in \mathcal{X}$. We begin by describing the standard version of the algorithm. The dual averaging scheme is based on a *proximal function* $\psi : \mathbb{R}^d \to \mathbb{R}$ assumed to be strongly convex with respect to a norm $\|\cdot\|$, more precisely, $\psi(y) \geq \psi(x) + \langle \nabla\psi(x), y - x\rangle + \frac{1}{2}\|x - y\|^2$ for all $x, y \in \mathcal{X}$. We assume w.l.o.g. that $\psi \geq 0$ on $\mathcal{X}$ and that $\psi(0) = 0$. Such proximal functions include the canonical quadratic $\psi(x) = \frac{1}{2}\|x\|_2^2$, which is strongly convex with respect to the ℓ_2-norm, and the negative entropy $\psi(x) = \sum_{j=1}^d x_i \log x_i - x_i$, which is strongly convex with respect to the ℓ_1-norm for x in the probability simplex.

We assume that each function f_i is *L-Lipschitz* with respect to the same norm $\|\cdot\|$—that is,

$$|f_i(x) - f_i(y)| \leq L\|x - y\| \qquad \text{for } x, y \in \mathcal{X}. \tag{1}$$

Many cost functions f_i satisfy this type of Lipschitz condition, for instance, convex functions on a compact domain $\mathcal{X}$ or any polyhedral function on an arbitrary domain [8]. The Lipschitz condition (1) implies that for any $x \in \mathcal{X}$ and any subgradient $g_i \in \partial f_i(x)$, we have $\|g_i\|_* \leq L$, where $\|\cdot\|_*$ denotes the *dual norm* to $\|\cdot\|$, defined by $\|v\|_* := \sup_{\|u\|=1} \langle v, u\rangle$.

The dual averaging algorithm generates a sequence of iterates $\{x(t), z(t)\}_{t=0}^\infty$ contained within $\mathcal{X} \times \mathbb{R}^d$. At time step t, the algorithm receives a subgradient $g(t) \in \partial f(x(t))$, and updates

$$z(t+1) = z(t) - g(t) \qquad \text{and} \qquad x(t+1) = \Pi_{\mathcal{X}}^{\psi}(-z(t+1), \alpha(t)). \tag{2}$$

Here $\{\alpha(t)\}_{t=0}^\infty$ is a non-increasing sequence of positive stepsizes and

$$\Pi_{\mathcal{X}}^{\psi}(z, \alpha) := \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \langle z, x\rangle + \frac{1}{\alpha}\psi(x) \right\} \tag{3}$$

is a type of projection. Intuitively, given the current iterate $(x(t), z(t))$, the next iterate $x(t+1)$ to chosen to minimize an averaged first-order approximation to the function f, while the proximal

function ψ and stepsize $\alpha(t) > 0$ enforce that the iterates $\{x(t)\}_{t=0}^{\infty}$ do not oscillate wildly. The algorithm is similar to the follow the perturbed/regularized leader algorithms developed in the context of online learning [12], though in this form the algorithm seems to be originally due to Nesterov [20]. In Section 4, we relate the above procedure to the distributed algorithm we now describe.

Distributed dual averaging: Here we consider a novel extension of dual averaging to the distributed setting. For all times t, each node $i \in V$ maintains a pair of vectors $(x_i(t), z_i(t)) \in \mathcal{X} \times \mathbb{R}^d$. At iteration t, node i computes a subgradient $g_i(t) \in \partial f_i(x_i(t))$ of the local function f_i and receives $\{z_j(t),\ j \in N(i)\}$ from its neighbors. Its update of the current estimate $x_i(t)$ is based on a weighted average of these parameters. To model the process, let $P \in \mathbb{R}^{n \times n}$ be a doubly stochastic symmetric matrix with $P_{ij} > 0$ only if $(i,j) \in E$ when $i \neq j$. Thus $\sum_{j=1}^{n} P_{ij} = \sum_{j \in N(i)} P_{ij} = 1$ for all $i \in V$ and $\sum_{i=1}^{n} P_{ij} = \sum_{i \in N(j)} P_{ij} = 1$ for all $j \in V$. Given a non-increasing sequence $\{\alpha(t)\}_{t=0}^{\infty}$ of positive stepsizes, each node $i \in V$ updates

$$z_i(t+1) = \sum_{j \in N(i)} P_{ji} z_j(t) - g_i(t), \quad \text{and} \quad x_i(t+1) = \Pi_{\mathcal{X}}^{\psi}(-z_i(t+1), \alpha(t)), \tag{4}$$

where the projection $\Pi_{\mathcal{X}}^{\psi}$ was defined in (3). In words, node i computes the new dual parameter $z_i(t+1)$ from a weighted average of its own subgradient $g_i(t)$ and the parameters $\{z_j(t), j \in N(i)\}$ in its neighborhood; it then computes the local iterate $x_i(t+1)$ by a proximal projection. We show convergence of the local sequence $\{x_i(t)\}_{t=1}^{\infty}$ to an optimum of the global objective via the *local average* $\widehat{x}_i(T) = \frac{1}{T}\sum_{t=1}^{T} x_i(t)$, which can evidently be computed in a decentralized manner.

3 Main results and consequences

We will now state the main results of this paper and illustrate some of their consequences. We give the proofs and a deeper investigation of related corollaries at length in the sections that follow.

Convergence of distributed dual averaging: We start with a result on the convergence of the distributed dual averaging algorithm that provides a decomposition of the error into an optimization term and the cost associated with network communication. In order to state this theorem, we define the averaged dual variable $\bar{z}(t) := \frac{1}{n}\sum_{i=1}^{n} z_i(t)$, and we recall the local time-average $\widehat{x}_i(T)$.

Theorem 1 (Basic convergence result)**.** *Given a sequence $\{x_i(t)\}_{t=0}^{\infty}$ and $\{z_i(t)\}_{t=0}^{\infty}$ generated by the updates (4) with step size sequence $\{\alpha(t)\}_{t=0}^{\infty}$, for each node $i \in V$ and any $x^* \in \mathcal{X}$, we have*

$$f(\widehat{x}_i(T)) - f(x^*) \leq \frac{1}{T\alpha(T)}\psi(x^*) + \frac{L^2}{2T}\sum_{t=1}^{T}\alpha(t-1) + \frac{3L}{T}\max_{j=1,\ldots,n}\sum_{t=1}^{T}\alpha(t)\left\|\bar{z}(t) - z_j(t)\right\|_*.$$

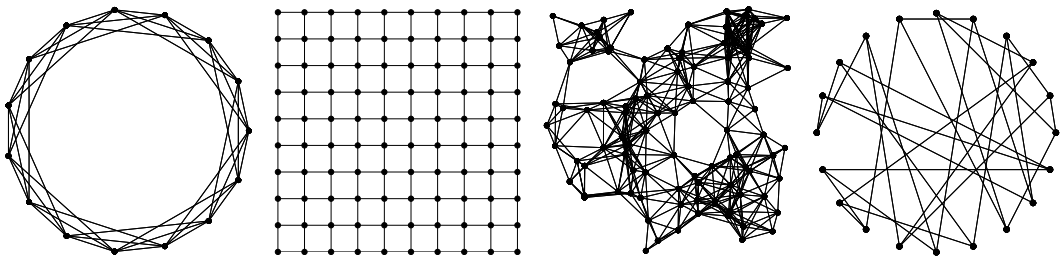
Theorem 1 guarantees that after T steps of the algorithm, every node $i \in V$ has access to a locally defined quantity $\widehat{x}_i(T)$ such that the difference $f(\widehat{x}_i(T)) - f(x^*)$ is upper bounded by a sum of three terms. The first two terms in the upper bound in the theorem are optimization error terms that are common to subgradient algorithms. The third term is the penalty incurred due to having different estimates at different nodes in the network, and it measures the deviation of each node's estimate of the average gradient from the true average gradient. Thus, roughly, Theorem 1 ensures that as long the bound on the deviation $\|\bar{z}(t) - z_i(t)\|_*$ is tight enough, for appropriately chosen $\alpha(t)$ (say $\alpha(t) \approx 1/\sqrt{t}$), the error of $\widehat{x}_i(T)$ is small uniformly across all nodes $i \in V$.

Convergence rates and network topology: We now turn to investigation of the effects of network topology on convergence rates. In this section,[1] we assume that the network topology is static and that communication occurs via a fixed doubly stochastic weight matrix P at every round. Since P is symmetric and stochastic, it has largest singular value $\sigma_1(P) = 1$. As the following result shows, the convergence of our algorithm is controlled by the *spectral gap* $\gamma(P) := 1 - \sigma_2(P)$ of P.

Theorem 2 (Rates based on spectral gap)**.** *Under the conditions and notation of Theorem 1, suppose moreover that $\psi(x^*) \leq R^2$. With step size choice $\alpha(t) = \frac{R\sqrt{1-\sigma_2(P)}}{4L\sqrt{t}}$, we have*

$$f(\widehat{x}_i(T)) - f(x^*) \leq 8\frac{RL}{\sqrt{T}} \cdot \frac{\log(T\sqrt{n})}{\sqrt{1-\sigma_2(P)}} \qquad \textit{for all } i \in V.$$

[1] We can weaken these conditions; see the long version of this paper for extensions to random P [4].



(a) (b) (c) (d)

Figure 1. (a) A 3-connected cycle. (b) 1-connected two-dimensional grid with non-toroidal boundary conditions. (c) A random geometric graph. (d) A random 3-regular expander graph.

This theorem establishes a tight connection between the convergence rate of distributed subgradient methods and the spectral properties of the underlying network. The inverse dependence on the spectral gap $1 - \sigma_2(P)$ is quite natural, since it is well-known to determine the rates of mixing in random walks on graphs [14], and the propagation of information in our algorithm is integrally tied to the random walk on the underlying graph with transition probabilities specified by P. Johansson et al. [11] establish rates for their Markov incremental gradient method (MIGD) of $\sqrt{n\Gamma_{ii}/T}$, where $\Gamma = (I - P + \mathbb{1}\mathbb{1}^\top/n)^{-1}$; performing an eigen-decomposition of the Γ matrix shows that $\sqrt{n\Gamma_{ii}}$ is always lower bounded by $1/\sqrt{1-\sigma_2(P)}$, our bound in Theorem 2.

Using Theorem 2, one can derive explicit convergence rates for several classes of interesting networks, and Figure 1 illustrates four graph topologies of interest. As a first example, the k-connected cycle in panel (a) is formed by placing n nodes on a circle and connecting each node to its k neighbors on the right and left. The grid (panel (b)) is obtained by connecting nodes to their k nearest neighbors in axis-aligned directions. In panel (c), we show a random geometric graph, constructed by placing nodes uniformly at random in $[0,1]^2$ and connecting any two nodes separated by a distance less than some radius $r > 0$. These graphs are often used to model the connectivity patterns of distibruted devices such as wireless sensor motes [7]. Finally, panel (d) shows an instance of a bounded degree expander, which belongs to a special class of sparse graphs that have very good mixing properties [3]. For many random graph models, a typical sample is an expander with high probability (e.g. random degree regular graphs [5]). In addition, there are several deterministic constructions of expanders that are degree regular (see Section 6.3 of Chung [3] for further details).

In order to state explicit convergence rates, we need to specify a particular choice of the matrix P that respects the graph structure. Let $A \in \mathbb{R}^{n\times n}$ be the symmetric adjacency matrix of the undirected graph G, satisfying $A_{ij} = 1$ when $(i,j) \in E$ and $A_{ij} = 0$ otherwise. For each node $i \in V$, let $\delta_i = |N(i)| = \sum_{j=1}^n A_{ij}$ denote the degree of node i and define the diagonal matrix $D = \mathrm{diag}\{\delta_1, \ldots, \delta_n\}$. Letting $\delta_{\max} = \max_{i\in V} \delta_i$ denote the maximum degree, we define

$$P_n(G) := I - \frac{1}{\delta_{\max}+1}\big(D - A\big), \tag{5}$$

which is symmetric and doubly stochastic by construction. The following result summarizes our conclusions for the choice (5) of stochastic matrix for different network topologies. We state the results in terms of optimization error achieved after T iterations and the number of iterations $T_G(\epsilon; n)$ required to achieve error ϵ for network type G with n nodes. (These are equivalent statements.)

Corollary 1. *Under the conditions of Theorem 2, using $P = P_n(G)$ gives the following rates.*

(a) k-connected paths and cycles: $f(\widehat{x}_i(T)) - f(x^*) = \mathcal{O}\big(\frac{RL}{\sqrt{T}}\,\frac{n\log(Tn)}{k}\big)$, $T(\epsilon; n) = \tilde{\mathcal{O}}(n^2/\epsilon^2)$.

(b) k-connected $\sqrt{n}\times\sqrt{n}$ grids: $f(\widehat{x}_i(T)) - f(x^*) = \mathcal{O}\big(\frac{RL}{\sqrt{T}}\,\frac{\sqrt{n}\log(Tn)}{k}\big)$, $T(\epsilon; n) = \tilde{\mathcal{O}}(n/\epsilon^2)$.

(c) Random geometric graphs with connectivity radius $r = \Omega(\sqrt{\log^{1+\epsilon} n/n})$ for any $\epsilon > 0$: $f(\widehat{x}_i(T)) - f(x^*) = \mathcal{O}\big(\frac{RL}{\sqrt{T}}\sqrt{\frac{n}{\log n}}\log(Tn)\big)$ *with high-probability,* $T(\epsilon; n) = \tilde{\mathcal{O}}(n/\epsilon^2)$.

(d) Expanders with bounded ratio of minimum to maximum node degree: $f(\widehat{x}_i(T)) - f(x^*) = \mathcal{O}\big(\frac{RL}{\sqrt{T}}\log(Tn)\big)$, $T(\epsilon; n) = \tilde{\mathcal{O}}(1/\epsilon^2)$.

By comparison, the results in the paper [11] give similar bounds for grids and cycles, but for d-dimensional grids we have $T(\epsilon; n) = \mathcal{O}(n^{2/d}/\epsilon^2)$ while MIGD achieves $T(\epsilon; n = \mathcal{O}(n/\epsilon^2)$; for expanders and the complete graph MIGD achieves $T(\epsilon; n) = \mathcal{O}(n/\epsilon^2)$. We provide the proof of Corollary 1 in Appendix A. Up to logarithmic factors, the optimization term in the convergence rate is always of the order $RL/\sqrt{T}$, while the remaining terms vary depending on the network topology.

In general, Theorem 2 implies that at most $T_G(\epsilon; n) = \mathcal{O}\big(\frac{1}{\epsilon^2} \cdot \frac{1}{1-\sigma_2(P_n(G))}\big)$ iterations are required to achieve an ϵ-accurate solution when using the matrix $P_n(G)$ defined in (5). It is interesting to ask whether this upper bound is actually tight. On one hand, it is known that even for centralized optimization algorithms, any subgradient method requires at least $\Omega\left(\frac{1}{\epsilon^2}\right)$ iterations to achieve ϵ-accuracy [19], so that the $1/\epsilon^2$ term is unavoidable. The next proposition addresses the complementary issue, namely whether the inverse spectral gap term is unavoidable for the dual averaging algorithm. For the quadratic proximal function $\psi(x) = \frac{1}{2}\|x\|_2^2$, the following result establishes a lower bound on the number of iterations in terms of graph topology and network structure:

Proposition 1. *Consider the dual averaging algorithm (4) with quadratic proximal function and communication matrix $P_n(G)$. For any graph G with n nodes, the number of iterations $T_G(c; n)$ required to achieve a fixed accuracy $c > 0$ is lower bounded as $T_G(c; n) = \Omega\big(\frac{1}{1-\sigma_2(P_n(G))}\big)$.*

The proof of this result, given in Appendix B, involves constructing a "hard" optimization problem and lower bounding the number of iterations required for our algorithm to solve it. In conjunction with Corollary 1, Proposition 1 implies that our predicted network scaling is sharp. Indeed, in Section 5, we show that the theoretical scalings from Corollary 1—namely, quadratic, linear, and constant in network size n—are well-matched in simulations of our algorithm.

4 Proof sketches

Setting up the analysis: Using techniques similar to some past work [18], we establish convergence via the two sequences $\bar{z}(t) := \frac{1}{n}\sum_{i=1}^n z_i(t)$ and $y(t) := \Pi_{\mathcal{X}}^{\psi}(-\bar{z}(t), \alpha)$. The average sum of gradients $\bar{z}(t)$ evolves in a very simple way: in particular, we have

$$\bar{z}(t+1) = \frac{1}{n}\sum_{i=1}^n\sum_{j=1}^n \big(P_{ji}(z_j(t) - \bar{z}(t))\big) + \bar{z}(t) - \frac{1}{n}\sum_{j=1}^n g_j(t) = \bar{z}(t) - \frac{1}{n}\sum_{j=1}^n g_j(t), \quad (6)$$

where the second equality follows from the double-stochasticity of P. The simple evolution (6) of the averaged dual sequence allows us to avoid difficulties with the non-linearity of projection that have been challenging in earlier work. Before proceeding with the proof of Theorem 1, we state a few useful results regarding the convergence of the standard dual averaging algorithm [20].

Lemma 2 (Nesterov)**.** *Let $\{g(t)\}_{t=1}^\infty \subset \mathbb{R}^d$ be an arbitrary sequence and $\{x(t)\}_{t=1}^\infty$ defined by the updates (2). For a non-increasing sequence $\{\alpha(t)\}_{t=0}^\infty$ of positive stepsizes and any $x^* \in \mathcal{X}$,*

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

Our second lemma allows us to restrict our analysis to the sequence $\{y(t)\}_{t=0}^\infty$ defined previously.

Lemma 3. *Consider sequences $\{x_i(t)\}_{t=1}^\infty$, $\{z_i(t)\}_{t=0}^\infty$, and $\{y(t)\}_{t=0}^\infty$ that evolve according to (4). Then for each $i \in V$ and any $x^* \in \mathcal{X}$, we have*

$$\sum_{t=1}^T f(x_i(t)) - f(x^*) \le \sum_{t=1}^T f(y(t)) - f(x^*) + L\sum_{t=1}^T \alpha(t)\|\bar{z}(t) - z_i(t)\|_*.$$

Now we give the proof of the first theorem.

Proof of Theorem 1: Our proof is based on analyzing the sequence $\{y(t)\}_{t=0}^\infty$. For any $x^* \in \mathcal{X}$,

$$\begin{aligned}\sum_{t=1}^T f(y(t)) - f(x^*) &= \sum_{t=1}^T \frac{1}{n}\sum_{i=1}^n f_i(x_i(t)) - f(x^*) + \sum_{t=1}^T \frac{1}{n}\sum_{i=1}^n [f_i(y(t)) - f_i(x_i(t))] \\ &\le \frac{1}{n}\sum_{t=1}^T\sum_{i=1}^n f_i(x_i(t)) - f(x^*) + \sum_{t=1}^T\sum_{i=1}^n \frac{L}{n}\|y(t) - x_i(t)\|, \quad (7)\end{aligned}$$

by the L-Lipschitz continuity of the f_i. Letting $g_i(t) \in \partial f_i(x_i(t))$ be a subgradient of f_i at $x_i(t)$,

$$\frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n} f_i(x_i(t)) - f_i(x^*) \leq \sum_{i=1}^{n} \langle g_i(t), y(t) - x^* \rangle + \sum_{i=1}^{n} \langle g_i(t), x_i(t) - y(t) \rangle . \tag{8}$$

By definition of $\bar{z}(t)$ and $y(t)$, we have $y(t) = \operatorname{argmin}_{x\in\mathcal{X}}\{\frac{1}{n}\sum_{s=1}^{t-1}\sum_{i=1}^{n}\langle g_i(s), x\rangle + \frac{1}{\alpha(t)}\psi(x)\}$. Thus, we see that the first term in the decomposition (8) can be written in the same way as the bound in Lemma 2, and as a consequence, we have the bound

$$\frac{1}{n}\sum_{t=1}^{T}\left\langle \sum_{i=1}^{n} g_i(t), y(t) - x^* \right\rangle \leq \frac{L^2}{2}\sum_{t=1}^{T}\alpha(t-1) + \frac{1}{\alpha(T)}\psi(x^*). \tag{9}$$

It remains to control the final two terms in the bounds (7) and (8). Since $\|g_i(t)\|_* \leq L$ by assumption, we use the α-Lipschitz continuity of the projection $\Pi_{\mathcal{X}}^{\psi}(\cdot, \alpha)$ [9, Theorem X.4.2.1] to see

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

Combining this bound with (7) and (9) yields the running sum bound

$$\sum_{t=1}^{T}\left[f(y(t)) - f(x^*)\right] \leq \frac{1}{\alpha(T)}\psi(x^*) + \frac{L^2}{2}\sum_{t=1}^{T}\alpha(t-1) + \frac{2L}{n}\sum_{t=1}^{T}\sum_{j=1}^{n}\alpha(t)\|\bar{z}(t) - z_j(t)\|_* . \tag{10}$$

Applying Lemma 3 to (10) gives that $\sum_{t=1}^{T}[f(x_i(t)) - f(x^*)]$ is upper bounded by

$$\frac{1}{\alpha(T)}\psi(x^*) + \frac{L^2}{2}\sum_{t=1}^{T}\alpha(t-1) + \frac{2L}{n}\sum_{t=1}^{T}\sum_{j=1}^{n}\alpha(t)\|\bar{z}(t) - z_j(t)\|_* + L\sum_{t=1}^{T}\alpha(t)\|\bar{z}(t) - z_i(t)\|_* .$$

Dividing both sides by T and using convexity of f yields the bound in Theorem 1.

Proof of Theorem 2: For this proof sketch, we adopt the following notational conventions. For an $n \times n$ matrix B, we call its singular values $\sigma_1(B) \geq \sigma_2(B) \geq \cdots \geq \sigma_n(B) \geq 0$. For a real symmetric B, we use $\lambda_1(B) \geq \lambda_2(B) \geq \ldots \geq \lambda_n(B)$ to denote the n real eigenvalues of B. We let $\Delta_n = \{x \in \mathbb{R}^n \mid x \succeq 0, \sum_{i=1}^{n} x_i = 1\}$ denote the n-dimensional probability simplex. We make frequent use of the following inequality [10]: for any positive integer $t = 1, 2, \ldots$ and any $x \in \Delta_n$,

$$\left\|P^t x - \mathbb{1}/n\right\|_{\mathrm{TV}} = \frac{1}{2}\left\|P^t x - \mathbb{1}/n\right\|_1 \leq \frac{1}{2}\sqrt{n}\left\|P^t x - \mathbb{1}/n\right\|_2 \leq \frac{1}{2}\sigma_2(P)^t\sqrt{n}. \tag{11}$$

We focus on controlling the network error term in Theorem 1, $\frac{L}{n}\sum_{t=1}^{T}\sum_{i=1}^{n}\alpha(t)\|\bar{z}(t) - z_i(t)\|_*$. Define the matrix $\Phi(t, s) = P^{t-s+1}$. Let $[\Phi(t, s)]_{ji}$ be entry j of column i of $\Phi(t, s)$. Then

$$z_i(t+1) = \sum_{j=1}^{n}[\Phi(t, s)]_{ji} z_j(s) - \sum_{r=s+1}^{t}\left(\sum_{j=1}^{n}[\Phi(t, r)]_{ji} g_j(r-1)\right) - g_i(t). \tag{12}$$

Clearly the above reduces to the standard update (4) when $s = t$. Since $\bar{z}(t)$ evolves simply as in (6), we assume w.l.o.g. that $z_i(0) = 0$ and use (12) to see

$$z_i(t) - \bar{z}(t) = \sum_{s=1}^{t-1}\sum_{j=1}^{n}(1/n - [\Phi(t-1, s)]_{ji}) g_j(s-1) + \left(\frac{1}{n}\sum_{j=1}^{n}(g_j(t-1) - g_i(t-1))\right). \tag{13}$$

We use the fact that $\|g_i(t)\|_* \leq L$ for all i and t and (13) to see that

$$\begin{aligned}
\|\bar{z}(t) - z_i(t)\|_* &= \left\| \sum_{s=1}^{t-1} \sum_{j=1}^{n} (1/n - [\Phi(t-1,s)]_{ji}) g_j(s-1) + \left(\frac{1}{n} \sum_{j=1}^{n} g_j(t-1) - g_i(t-1) \right) \right\|_* \\
&\leq \sum_{s=1}^{t-1} \sum_{j=1}^{n} \|g_j(s-1)\|_* \left|(1/n) - [\Phi(t-1,s)]_{ji}\right| + \frac{1}{n} \sum_{i=1}^{n} \|g_j(t-1) - g_i(t-1)\|_* \\
&\leq \sum_{s=1}^{t-1} L \left\|[\Phi(t-1,s)]_i - \mathbb{1}/n\right\|_1 + 2L. \qquad (14)
\end{aligned}$$

Now we break the sum in (14) into two terms separated by a cutoff point $\widehat{t}$. The first term consists of "throwaway" terms, that is, timesteps s for which the Markov chain with transition matrix P has not mixed, while the second consists of steps s for which $\|[\Phi(t-1,s)]_i - \mathbb{1}/n\|_1$ is small. Note that the indexing on $\Phi(t-1,s) = P^{t-s+1}$ implies that for small s, $\Phi(t-1,s)$ is close to uniform. From the inequality (11), we have $\|[\Phi(t,s)]_j - \mathbb{1}/n\|_1 \leq \sqrt{n}\sigma_2(P)^{t-s+1}$. Hence, if $t - s \geq \frac{\log \epsilon^{-1}}{\log \sigma_2(P)^{-1}} - 1$, then we are guaranteed $\|[\Phi(t,s)]_j - \mathbb{1}/n\|_1 \leq \sqrt{n}\epsilon$. Thus, by setting $\epsilon^{-1} = T\sqrt{n}$, for $t - s + 1 \geq \frac{\log(T\sqrt{n})}{\log \sigma_2(P)^{-1}}$, we have $\|[\Phi(t,s)]_j - \mathbb{1}/n\|_1 \leq \frac{1}{T}$. For larger s, we simply have $\|[\Phi(t,s)]_j - \mathbb{1}/n\|_1 \leq 2$. The above suggests that we split the sum at $\widehat{t} = \frac{\log T\sqrt{n}}{\log \sigma_2(P)^{-1}}$. Since $t - 1 - (t - \widehat{t}) = \widehat{t}$ and there are at most T steps in the summation,

$$\begin{aligned}
\|\bar{z}(t) - z_i(t)\|_* &\leq L \sum_{s=t-\widehat{t}}^{t-1} \|\Phi(t-1,s)e_i - \mathbb{1}/n\|_1 + L \sum_{s=1}^{t-1-\widehat{t}} \|\Phi(t-1,s)e_i - \mathbb{1}/n\|_1 + 2L \\
&\leq 2L \frac{\log(T\sqrt{n})}{\log \sigma_2(P)^{-1}} + 3L \leq 2L \frac{\log(T\sqrt{n})}{1 - \sigma_2(P)} + 3L. \qquad (15)
\end{aligned}$$

The last inequality follows from the concavity of $\log(\cdot)$, since $\log \sigma_2(P)^{-1} \geq 1 - \sigma_2(P)$.

Combining (15) with the running sum bound in (10) of the proof of the basic theorem, Theorem 1, we find that for $x^* \in \mathcal{X}$,

$$\sum_{t=1}^{T} f(y(t)) - f(x^*) \leq \frac{1}{\alpha(T)} \psi(x^*) + \frac{L^2}{2} \sum_{t=1}^{T} \alpha(t-1) + 6L^2 \sum_{t=1}^{T} \alpha(t) + 4L^2 \frac{\log(T\sqrt{n})}{1 - \sigma_2(P)} \sum_{t=1}^{T} \alpha(t).$$

Appealing to Lemma 3 allows us to obtain the same result on the sequence $x_i(t)$ with slightly worse constants. Since $\sum_{t=1}^{T} t^{-1/2} \leq 2\sqrt{T} - 1$, using the assumption that $\psi(x^*) \leq R^2$, bounding $f(\widehat{x}_i(T)) \leq \frac{1}{T} \sum_{t=1}^{T} f(x_i(t))$, and setting $\alpha(t)$ as in the theorem statement completes the proof.

5 Simulations

In this section, we report experimental results on the network scaling behavior of the distributed dual averaging algorithm as a function of the graph structure and number of processors n. These results illustrate the excellent agreement of the empirical behavior with our theoretical predictions. For all experiments reported here, we consider distributed minimization of a sum of hinge losses. We solve a synthetic classification problem, in which we are given n pairs of the form $(a_i, y_i) \in \mathbb{R}^d \times \{-1, +1\}$, where $a_i \in \mathbb{R}^d$ corresponds to a feature vector and $y_i \in \{-1, +1\}$ is the associated label. Given the shorthand notation $[c]_+ := \max\{0, c\}$, the hinge loss associated with a linear classifier based on x is given by $f_i(x) = [1 - y_i \langle a_i, x \rangle]_+$. The global objective is given by the sum $f(x) := \frac{1}{n} \sum_{i=1}^{n} [1 - y_i \langle a_i, x \rangle]_+$. Setting $L = \max_i \|a_i\|_2$, we note that f is L-Lipschitz and non-smooth at any point with $\langle a_i, x \rangle = y_i$. As is common, we impose a quadratic regularization, choosing $\mathcal{X} = \{x \in \mathbb{R}^d \mid \|x\|_2 \leq 5\}$. Then for a given graph size n, we form a random instance of this SVM classification problem. Although this is a specific ensemble of problems, we have observed qualitatively similar behavior for other problem classes. In all cases, we use the optimal setting of the step size α specified in Theorem 2 and Corollary 1.

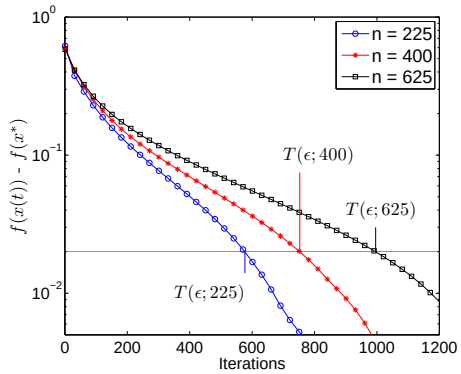
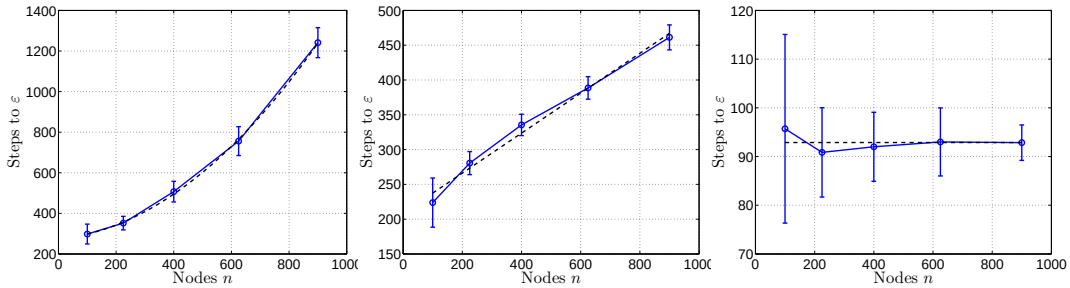


Figure 2. Plot of the function error versus the number of iterations for a grid graph. Each curve corresponds to a grid with a different number of nodes ($n \in \{225, 400, 600\}$). As expected, larger graphs require more iterations to reach a pre-specified tolerance $\epsilon > 0$, as defined by the iteration number $T(\epsilon; n)$. The network scaling problem is to determine how $T(\epsilon; n)$ scales as a function of n.



(a) (b) (c)

Figure 3. Each plot shows the number of iterations required to reach a fixed accuracy ϵ (vertical axis) versus the network size n (horizontal axis). Panels show the same plot for different graph topologies: (a) single cycle; (b) two-dimensional grid; and (c) bounded degree expander.

Figure 2 provides plots of the function error $\max_i[f(\widehat{x}_i(T) - f(x^*)]$ versus the number of iterations for grid graphs with a varying number of nodes $n \in \{225, 400, 625\}$. In addition to demonstrating convergence, these plots also show how the convergence time scales as a function of the graph size. We also experimented with the algorithm and stepsize suggested by previous analyses [21]; the resulting stepsize is so small that the method effectively jams and makes no progress.

In Figure 3, we compare the theoretical predictions of Corollary 1 with the actual behavior of dual subgradient averaging. Each panel shows the function $T_G(\epsilon; n)$ versus the graph size n for the fixed value $\epsilon = 0.1$; the three different panels correspond to different graph types: cycles (a), grids (b) and expanders (c). In the panels, each point on the solid blue curve is the average of 20 trials, and the bars show standard errors. For comparison, the dotted black line shows the theoretical prediction. Note that the agreement between the empirical behavior and theoretical predictions is excellent in all cases. In particular, panel (a) exhibits the quadratic scaling predicted for the cycle, panel (b) exhibits the the linear scaling expected for the grid, and panel (c) shows that expander graphs have the desirable property of having constant network scaling.

6 Conclusions

In this paper, we have developed and analyzed an efficient algorithm for distributed optimization based on dual averaging of subgradients. In addition to establishing convergence, we provided a careful analysis of the algorithm's network scaling. Our results show an inverse scaling in the spectral gap of the graph, and we showed that this prediction is tight in general via a matching lower bound. We have implemented our method, and our simulations show that these theoretical predictions provide a very accurate characterization of its behavior. In the extended version of this paper [4], we also show that it is possible to extend our algorithm and analysis to the cases in which communication is random and not fixed, the algorithm receives stochastic subgradient information, and for minimization of composite regularized objectives of the form $f(x) + \varphi(x)$.

Acknowledgements: JCD was supported by an NDSEG fellowship and Google. AA was supported by a Microsoft Research Fellowship. In addition, AA was partially supported by NSF grants DMS-0707060 and DMS-0830410. MJW and AA were partially supported by AFOSR-09NL184.

References

[1] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[3] F.R.K. Chung. *Spectral Graph Theory*. AMS, 1998.

[4] J. Duchi, A. Agarwal, and M. Wainwright. Dual averaging for distributed optimization: convergence analysis and network scaling. URL http://arxiv.org/abs/1005.2012, 2010.

[5] J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue of random regular graphs. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 587–598, New York, NY, USA, 1989. ACM.

[6] R. Gray. Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory*, 2(3):155–239, 2006.

[7] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[8] J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer, 1996.

[9] J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Springer, 1996.

[10] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[11] B. Johansson, M. Rabi, and M. Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3):1157–1170, 2009.

[12] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[13] V. Lesser, C. Ortiz, and M. Tambe, editors. *Distributed Sensor Networks: A Multiagent Perspective*, volume 9. Kluwer Academic Publishers, May 2003.

[14] D. Levin, Y. Peres, and E. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[15] I. Lobel and A. Ozdaglar. Distributed subgradient methods over random networks. Technical Report 2800, MIT LIDS, 2008.

[16] R. McDonald, K. Hall, and G. Mann. Distributed training strategies for the structured perceptron. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.

[17] A. Nedic and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.

[18] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54:48–61, 2009.

[19] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, 1983.

[20] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming A*, 120(1):261–283, 2009.

[21] S. Sundhar Ram, A. Nedic, and V. V. Veeravalli. Distributed subgradient projection algorithm for convex optimization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3653–3656, 2009.

[22] J. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[23] U. von Luxburg, A. Radl, and M. Hein. Hitting times, commute distances, and the spectral gap for large random geometric graphs. URL http://arxiv.org/abs/1003.1266, 2010.

[24] L. Xiao, S. Boyd, and S. J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

A Proof of Corollary 1

In order to prove the convergence rates, we relate our choice of P to a well-known quantity in spectral graph theory, specifically, the graph Laplacian [3]. Recall the definition of A as the symmetric adjacency matrix of the undirected graph G, the degree of node i as $\delta_i = |N(i)| = \sum_{j=1}^n A_{ij}$, and the diagonal matrix $D = \mathrm{diag}\{\delta_1, \ldots, \delta_n\}$. We assume that the graph is connected, so that $\delta_i \geq 1$ for all i and D is invertible. The *(normalized) graph Laplacian* is given by $\mathcal{L}(G) = I - D^{-1/2}AD^{-1/2}$, a matrix which is always symmetric, positive semidefinite, and satisfies $\mathcal{L}\mathbb{1} = 0$. Therefore, when the graph is degree-regular ($\delta_i = \delta$ for all $i \in V$), the standard random walk with self loops on G given by the matrix $P := I - \frac{\delta}{\delta+1}\mathcal{L}$ is doubly stochastic and is valid for our theory. For non-degree regular graphs, we make the minor modification as in (5) to obtain

$$P_n(G) := I - \frac{1}{\delta_{\max}+1}\big(D - A\big) = I - \frac{1}{\delta_{\max}+1}D^{1/2}\mathcal{L}D^{1/2}. \tag{16}$$

As noted earlier, $P_n(G)$ is doubly stochastic, and if the graph is δ-regular, then $P_n(G)$ is simply $(I + A)/(\delta + 1)$. The remainder of the corollary is based on bounding the spectral gap of $P_n(G)$.

Lemma 4. *The matrix P satisfies* $\sigma_2(P_n(G)) \leq \max\Big\{1 - \frac{\min_i \delta_i}{\delta_{\max}+1}\lambda_{n-1}(\mathcal{L}), \frac{\delta_{\max}}{\delta_{\max}+1}\lambda_1(\mathcal{L}) - 1\Big\}$.

Proof By a theorem of Ostrowski on congruent matrices (cf. Theorem 4.5.9, [10]), we have

$$\lambda_k(D^{1/2}\mathcal{L}D^{1/2}) \in \Big[\min_i \delta_i\lambda_k(\mathcal{L}), \max_i \delta_i\lambda_k(\mathcal{L})\Big]. \tag{17}$$

Since $\mathcal{L}\mathbb{1} = 0$, we have $\lambda_n(\mathcal{L}) = 0$, and so it suffices to focus on $\lambda_1(D^{1/2}\mathcal{L}D^{1/2})$ and $\lambda_{n-1}(D^{1/2}\mathcal{L}D^{1/2})$. From the definition (16), the eigenvalues of P are of the form $1 - (\delta_{\max} + 1)^{-1}\lambda_k(D^{1/2}\mathcal{L}D^{1/2})$. The bound (17) coupled with the fact that all the eigenvalues of $\mathcal{L}$ are non-negative implies that $\sigma_2(P) = \max_{k<n}\big\{\big|1 - (\delta_{\max} + 1)^{-1}\lambda_k(D^{1/2}\mathcal{L}D^{1/2})\big|\big\}$ is upper bounded by the larger of $1 - \frac{\delta_{\min}}{\delta_{\max}+1}\lambda_{n-1}(\mathcal{L})$ and $\frac{\delta_{\max}}{\delta_{\max}+1}\lambda_1(\mathcal{L}) - 1$. □

Computing the upper bound in Lemma 4 requires controlling $\lambda_{n-1}(\mathcal{L})$ and $\lambda_1(\mathcal{L})$. To circumvent this complication, we use the well-known idea of a lazy random walk [3, 14], in which we replace by P by $\frac{1}{2}(I + P)$. The resulting symmetric matrix has the same eigenstructure as P, moreover,

$$\sigma_2\Big(\frac{1}{2}(I + P)\Big) = \lambda_2\Big(I - \frac{1}{2(\delta_{\max}+1)}D^{1/2}\mathcal{L}D^{1/2}\Big) \leq 1 - \frac{\delta_{\min}}{2(\delta_{\max}+1)}\lambda_{n-1}(\mathcal{L}). \tag{18}$$

Consequently, it is sufficient to bound only $\lambda_{n-1}(\mathcal{L})$. The convergence rate implied by the lazy random walk through Theorem 2 is no worse than twice that of the original walk, which is insignificant for the analysis in this paper. The remainder of the proof involves exploiting results from spectral graph theory [3] in order to control the eigenvalues of the Laplacian.

Cycles and paths: Recall the regular k-connected cycle from Figure 1(a), constructed by placing the n nodes on a circle and connecting every node to k neighbors on the right and left. For this graph, the Laplacian $\mathcal{L}$ is a circulant matrix with diagonal entries 1 and off-diagonal non-zero entries $-1/2k$. Using known results on circulant matrices [2, 6], we find that $\lambda_{n-1}(\mathcal{L}) = \Theta\big(\frac{k^2}{n^2}\big)$ for $k = o(n)$. For the regular k-connected path, by computing Cheeger constants [3, Chapter 2] we find that if $k \leq \sqrt{n}$, then $\lambda_{n-1}(\mathcal{L}) = \Theta(k^2/n^2)$. Note also that for the k-connected path on n nodes, $\min_i \delta_i = k$ and $\delta_{\max} = 2k$. Thus, we can combine the above with Lemma 4 to see that for regular k-connected paths or cycles with $k \leq \sqrt{n}$, $\sigma_2(P) = 1 - \Theta\big(\frac{k^2}{n^2}\big)$. Substituting the bound into Theorem 2 yields the claim of Corollary 1(a).

Regular grids: Now consider the case of a $\sqrt{n}$-by-$\sqrt{n}$ grid, focusing in particular on regular k-connected grids, in which any node is joined to every node that is fewer than k horizontal or vertical edges away in an axis-aligned direction. In this case, we use results on Cartesian products of graphs [3, Section 2.6] to analyze the eigen-structure of the Laplacian. In particular, the $\sqrt{n}$-by-$\sqrt{n}$ k-connected grid is simply the Cartesian product of two regular k-connected paths of $\sqrt{n}$ nodes. The second smallest eigenvalue of a Cartesian product of graphs is half the minimum of second-smallest eigenvalues of the original graphs [3, Theorem 2.13]. Thus, based on the preceding discussion of k-connected cycles, we conclude that if $k \leq n^{1/4}$, then we have $\lambda_{n-1}(\mathcal{L}) = \Theta(k^2/n)$, and we use Lemma 4 and (18) to see that the result in Corollary 1(b) immediately follows.

Random geometric graphs: Using the proof of Lemma 10 from Boyd et al. [2], we see that for any $\epsilon > 0$, if $r = \sqrt{\log^{1+\epsilon} n/(n\pi)}$, then with probability at least $1 - 2/n^{c-1}$,

$$\min_i \delta_i \geq \log^{1+\epsilon} n - \sqrt{2}c \log n \quad \text{and} \quad \max_i \delta_i \leq \log^{1+\epsilon} n + \sqrt{2}c \log n. \tag{19}$$

Thus, letting $\mathcal{L}$ be the graph Laplacian of a random geometric graph, if we can bound $\lambda_{n-1}(\mathcal{L})$, (19) coupled with Lemma 4 will control the convergence rate of our algorithm.

Von Luxburg et al. [23] give concentration results on the second-smallest eigenvalue of a geometric graph. In particular, their Theorem 3 says that if $r = \omega(\sqrt{\log n/n})$, then with exceedingly high probability, $\lambda_{n-1}(\mathcal{L}) = \Omega(r)$. Using (19), we see that for $r = (\log^{1+\epsilon} n/n)^{1/2}$, the ratio $\frac{\min_i \delta_i}{\max_i \delta_i} = \Theta(1)$ and $\lambda_{n-1}(\mathcal{L}) = \Omega\big(\frac{\log^{1+\epsilon} n}{n}\big)$ with high probability. Combining this with Lemma 4 and (18), we have $\sigma_2(P) = 1 - \Omega\big(\frac{\log^{1+\epsilon} n}{n}\big)$, the desired result of Corollary 1(c).

Expanders: The constant spectral gap in expanders [3, Chapter 6] removes any penalty due to network communication (up to logarithmic factors) and hence yields Corollary 1(d).

B Proof of Proposition 1

The proof is based on construction of a set of objective functions f_i that force convergence to be slow by using the second eigenvector of the communication matrix P. Recall that $\mathbb{1} \in \mathbb{R}^n$ is the eigenvector of P corresponding to its largest eigenvalue (equal to 1). Let $v \in \mathbb{R}^n$ be the eigenvector of P corresponding to its second singular value, $\sigma_2(P)$. By using the lazy random walk (see Sec. A and (18)), we may assume without loss of generality that $\lambda_2(P) = \sigma_2(P)$. Let $w = \frac{v}{\|v\|_\infty}$ be a normalized version of the second eigenvector of P, and note that $\sum_{i=1}^n w_i = 0$. Without loss of generality, we assume that there is an index i for which $w_i = -1$ (otherwise we can flip signs in what follows); by re-indexing as needed, we can assume that $w_1 = -1$. We set $\mathcal{X} = [-1, 1] \subset \mathbb{R}$ and define the univariate functions $f_i(x) := (c + w_i)x$. The global problem is then to minimize

$$\frac{1}{n}\sum_{i=1}^n f_i(x) = \frac{1}{n}\sum_{i=1}^n (c + w_i)x = cx$$

for some constant $c > 0$ to be chosen. Note that each f_i is $c + 1$-Lipschitz. By construction, we see immediately that $x^* = -1$ is optimal for the global problem.

Consider the evolution of the $\{z(t)\}_{t=0}^\infty \subset \mathbb{R}^n$, as generated by the update (4). By construction, we have $g_i(t) = c + w_i$ for all t. Defining the vector $g = (c\mathbb{1} + w) \in \mathbb{R}^n$, we recall that $P\mathbb{1} = \mathbb{1}$ to see

$$\begin{aligned} z(t+1) &= Pz(t) - g = P^2 z(t-1) - Pg - g = \cdots = -\sum_{\tau=0}^{t} P^\tau g \\ &= -\sum_{\tau=0}^{t-1} P^\tau (w + c\mathbb{1}) = -\sum_{\tau=0}^{t-1} P^\tau w - ct\mathbb{1} = -\sum_{\tau=0}^{t-1} \sigma_2(P)^\tau w - ct\mathbb{1}. \end{aligned} \tag{20}$$

In order to establish a lower bound, it suffices to show that at least one node is far from the optimum after t steps, and we focus on node 1. Since $w_1 = -1$, the evolution (20) guarantees that

$$z_1(t+1) = \sum_{\tau=0}^{t-1} \sigma_2(P)^\tau - ct = \frac{1 - \sigma_2(P)^{t-1}}{1 - \sigma_2(P)} - ct. \tag{21}$$

Recalling that $\psi(x) = \frac{1}{2}x^2$ for this scalar setting, we have

$$x_i(t+1) = \operatorname*{argmin}_{x \in \mathcal{X}} \Big\{ -z_i(t+1)x + \frac{1}{2\alpha(t)}x^2 \Big\} = \operatorname*{argmin}_{x \in \mathcal{X}} \Big\{ (x - \alpha(t) z_i(t+1))^2 \Big\}$$

Hence $x_1(t)$ is the projection of $\alpha(t)z_1(t+1)$ onto $[-1, 1]$, and unless $z_1(t) < 0$ we have

$$f(x_1(t)) - f(-1) \geq c > 0.$$

If t is overly small, the relation (21) will guarantee that $z_1(t) \geq 0$, so that $x_1(t)$ is far from the optimum. If we choose $c \leq 1/3$, then a little calculation with (21) shows that we require $t = \Omega((1 - \sigma_2(P))^{-1})$ in order to drive $z_1(t)$ below zero.