DISTRIBUTED CONVEX OPTIMIZATION WITH LIMITED COMMUNICATIONS

Milind Rao[†] *Stefano Rini*[⋆] *Andrea Goldsmith*[†]

[†] Electrical Engineering, Stanford University, Stanford, CA
[⋆] National Chiao Tung University, Hsinchu, Taiwan
{milind,andreag}@stanford.edu, stefano@nctu.edu.tw

ABSTRACT

In this paper, a distributed convex optimization algorithm, termed *distributed coordinate dual averaging* (DCDA) algorithm, is proposed. The DCDA algorithm addresses the scenario of a large distributed optimization problem with limited communication among nodes in the network. Currently known distributed subgradient descent methods, such as the distributed dual averaging or the distributed alternating direction method of multipliers, assume that nodes can exchange messages of large cardinality. Such an assumption on the network communication capabilities is not valid in many scenarios of practical relevance. To address this setting, we propose the DCDA algorithm as a distributed convex optimization algorithm in which the communication between nodes in each round is restricted to a fixed number of dimensions. We bound the rate of convergence under different communication protocols and network architectures for this algorithm. We also consider the extensions to the cases of imperfect gradient knowledge and when transmitted messages are corrupted by additive noise or are quantized. Numerical simulations demonstrating the performance of DCDA in these different settings are also provided.

Index Terms— Distributed optimization, subgradient descent methods, convex analysis, wireless communications

1. INTRODUCTION

With the emergence in recent years of big data paradigms, decentralized optimization algorithms have received considerables interest in the literature. A distributed optimization problem of particular relevance is the one in which the global objective function is obtained as the sum of convex functions, each known at one of the nodes in the network. Originally considered by Tsitsiklis et. al. [1], this problem is broadly referred to as the consensus problem. A number of distributed subgradient descent methods have been proposed to solve the consensus problem such as distributed subgradient descent (DSG) [2, 3], distributed dual averaging (DDA) [4], accelerated Nesterov gradient descent [5] and distributed alternating direction method of multipliers (DADMM) [6, 7]. A DSG algorithm for the consensus problem is initially proposed in [2], building upon consensus algorithms for computing the exact averages of initial values at the agents [8], where each node updates its estimate using a linear combination of the estimates of its neighbors and the gradient of its local function. In the literature, a number of variations of this algorithm have been considered, such as continuous time extensions [9], networks with link failures [3], and quantized communication [10]. Liu et al [11] analyze an asynchronous distributed coordinate descent version where one coordinate of the optimal solution is communicated at each time instant. Inspired by Nesterov's dual averaging algorithm [12], Duchi et. al. [4] propose the DDA algorithm where each node maintains an estimate for a dual variable by averaging the estimates of its neighbors and adding the gradient. A proximal projection of the dual variable produces an optimization variable. The dual variable is updated similarly to the DSG algorithm, while the dual projection allows nonlinear constraints on the solution to incorporated. In [4], the authors also study the performance of the DDA algorithm in the presence of time varying networks, communication of gossip protocols, and stochastic gradients. The analysis of the DDA algorithm with delays in the communication network is performed in [13, 14]. The authors of [15] study the computation/communication trade-off for the DDA algorithm by considering the case in which communication is subject to a total cost constraint. Another popular class of algorithms to solve distributed constrained convex optimization problems are the DADMM algorithms proposed in [6], building upon the ADMM algorithm of [16]. The analysis of convergence for this algorithm is performed in [17], while the case of asynchronous communications is studied in [7]. In the above algorithms, the messages exchanged between nodes at each time instant is on the order of the dimension d of the optimization variable.

In this work, we introduce a decentralized optimization algorithm in which the dimension of the messages exchanged between nodes is restricted to dimension $m \leq d$ and yet guarantees convergence to the optimal value at all nodes. This algorithm is inspired by the DDA algorithm of [4] and is thus termed the *distributed coordinate dual averaging* (DCDA) algorithm. In the following, we derive the convergence of the DCDA algorithm for different communication protocols and network architectures. Additionally, we study the behavior of the algorithm in the scenario of a stochastic gradient, with noisy and quantized communication. We show an inverse relationship between number of iterations t, message dimension m such that t doubles when m is halved to achieve an optimality gap of ϵ.

Proofs are omitted for brevity: complete proofs are provided in an extended version of the manuscript available online [18].

2. PROBLEM FORMULATION

We study the distributed convex optimization problem in which the minimum of a function is to be computed when its factors are distributed across a network subject to communication constraints. More precisely, consider the n-nodes undirected graph $G = (V, E)$, $V = [1 : n]$, and $E \subset V \times V$ in which each node V_i is associated the function $f_i : \mathbb{R}^d \to \mathbb{R}$. We aim to minimize

This work was supported by the Texas Instruments Stanford Graduate Fellowship and the NSF-Center for the Science of Information. The work of S. Rini was funded by the Ministry of Science and Technology (MOST) under the grant # 107-2221-E-009-032-MY3.

function $f(x)$ obtained as

$$f(x) = \sum_{i=1}^{n} f_i(x), \tag{1}$$

for $x \in \mathcal{X}$ with $\mathcal{X}$ closed and convex. We assume that each $f_i(x)$ is convex and L-Lipschitz with respect to a norm $\|\cdot\|$, i.e. $|f_i(x) - f_i(y)| \leq L\|x-y\|, \;\; x, y \in \mathcal{X}$. The Lipschitz condition 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 $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. At each time instant $t \in \mathbb{N}$, the node V_i maintains an estimate $x_i(t)$ of the value x^* which attains the minimum of the function $f(x)$ in (1). The node V_i is able to communicate to the node V_j at the time instant t if the two nodes are connected by an edge E in G. Let A be the symmetric incidence matrix of G.[1] Some examples of a network are:

− **fully-connected network:** in which $A = \mathbf{1}\mathbf{1}^\intercal - \mathbf{I}$,

− **random network:** in which two nodes are connected with probability p,

− **ring network:** in which $A_{ij, i\neq j} = 1$ iff $|i-j| \mod n \leq l$ for some $l \in \mathbb{N}$.

Upon receiving the messages from its neighbors, each node V_i updates its estimate of $x_i(t)$. In the distributed optimization problem, the goal is to determine a set of communication strategies and estimate update rules such that each $x_i(t) \to x^*$.

In the following, given the time sequence $c(t) \in \mathbb{R}^n$, we will denote the time and space average as $\widehat{c}(t) = \frac{1}{t}\sum_{t'=1}^{t} x(t')$ and $\overline{c}(t) = \frac{1}{n}\sum_{i=1}^{n} c_i(t)$ respectively.

The DCDA algorithm: In the DCDA algorithm, each node V_i maintains both an estimate of the optimization variable, $x_i(t)$, and its dual variable, $z_i(t)$. At each time instant, both the primal variable and the dual variables are updated according to the message received from the neighboring nodes and the subgradient of the objective function f_i in the primal estimate x_i, $g_i(t)$. At each iteration, each node i broadcasts a subset of its d coordinates of the dual variable $z(t)$ to a subset of of its neighbors. For instance, node i broadcasts coordinate k to neighbors $N^k(i)$. The update of the dual variable is a component wise update

$$[z_i(t+1)]_k = \sum_{j \in N^k(i)} P_{ij}^k(t)[z_j(t)]_k + [g_i(t)]_k \quad \forall\, k, \tag{2}$$

where $P^k(t)$ is a doubly stochastic matrix and where $P_{ij}^k > 0$ if and only if $A_{ij} > 0$ and the node j is broadcasting the set of coordinates k to node i. In the following, we consider three different policies for the selection of the coordinate k broadcasted by the nodes:

− **static sharing scheme:** at each time instant, nodes transmit the same coordinates to their neighbors, corresponding to $P^k(t) = P^k \;\forall\, t$ for some fixed P^k.

− **round robin scheme:** in which the k^{th} coordinate is shared every π time instances, corresponding to $P^k(t) = P_\pi$ when $t = n\pi + k$ for some $n \in \mathbb{N}$, else $P^k(t) = \mathbf{I}$.

− **randomized scheme:** in which nodes randomly and uniformly select the coordinate to be transmitted in each time instant.

Note that the stated sharing scheme with $P^k = P$ corresponds to the DDA algorithm: this corresponds to the case when nodes broadcast their entire dual variable to their neighbors. Also note that, given a symmetric adjacency network $A^k(t)$ for coordinate k at time t, we can obtain the doubly stochastic matrix $P^k(t)$ as

$$D^k(t) = \text{diag}(A^k(t)\mathbf{1}), \;\; P^k(t) = \mathbf{I} - \frac{D^k(t) - A^k(t)}{\max_i D^k(t)_{ii} + 1}.$$

At each time instant t, the primal variable $x_i(t)$ is computed from $z_i(t)$ as:

$$x_i(t) = \Pi_{\psi, \alpha(t-1)}(z_i(t)) \tag{3}$$

The function $\Pi_{\psi,\alpha(t)}$ is a type of non-linear proximal projection and is used to stabilize estimates of the primal variable and ensure that optimization constraints are satisfied. It is defined as $\Pi_{\psi,\alpha(t)}(z) = \text{argmin}_x \langle x, z\rangle + \frac{1}{\alpha(t)}\psi(x)$. $\{\alpha(t)\}_{t=0}^{\infty}$ is a non-increasing sequence of positive step-sizes which typically scales as $1/\sqrt{t}$. Also, $\psi : \mathbb{R}^d \to \mathbb{R}$ is a *proximal function*, that is assumed to be 1-strongly convex with respect to norm $\|\cdot\|$. and positive defined. Examples of proximal functions include:

− **squared proximal function**: $\psi(x) = \frac{1}{2}\|x\|_2^2$ is 1-strongly convex with respect to the ℓ_2-norm.

− **entropic proximal function**: $\psi(x) = \sum_{k=1}^{d} x_i \log x_i - x_i$ is 1-strongly convex with respect to the ℓ_1-norm.

The performance of the DCDA algorithm is studied in terms of the convergence to zero of the term $f(\hat{x}_i(T)) - f(x^*)$. Finally, we consider three extensions of the DCDA algorithm:

− **stochastic gradient:** the objective function subgradient is not exactly known at each node,

− **noisy communication:** transmissions are corrupted by additive noise,

− **quantized communications:** in which transmissions are quantized before communication.

3. MAIN RESULTS

The main results of the paper consists of the characterization of the DCDA algorithm convergence rate for different coordinate selection policies and communication networks.

Theorem 1. *Let the sequences $\{x_i(t)\}_{t=0}^{\infty}$ and $\{z_i(t)\}_{t=0}^{\infty}$ be generated by the updates* (3) *and* (2) *with step size sequence $\{\alpha(t)\}_{t=0}^{\infty}$. Then for any $x^* \in \mathcal{X}$ and for each node $i \in V$, the DCDA algorithm is such that*

$$f(\hat{x}_i(T)) - f(x^*) \leq \frac{\psi(x^*)}{T\alpha(t)} + \frac{1}{T}\sum_{t=1}^{T}\alpha(t-1)\|\bar{g}(t)\|_*^2 \tag{4}$$

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

The result in Thm. 1 is substantially equivalent to the result, of [4, Thm. 1]. The first two terms in (4) are optimization error terms common to sub-gradient algorithms while the last two are penalties incurred due to having different estimates at different nodes in the network or the penalty from consensus. The result in Thm. 1 can be further developed for specific communication protocols.

Lemma 2. Static sharing scheme: *For the settings in Thm. 1, the DCDA algorithm under the static coordinate sharing scheme is such that*

$$f(\hat{x}_i(T)) - f(x^*) \leq \frac{\psi(x^*)}{T\alpha(T)} \tag{5}$$

$$+ \frac{L^2}{T}\sum_{t=1}^{T} 4\alpha(t-1)\left(\frac{2\min(d,n)\log\sqrt{n}dT}{1 - \max_k \sigma_2(P^k)} + 3\right),$$

[1] That is, A_{ij} is non-zero only if nodes i and j are neighbors.

where $\sigma_2(M)$ is the second largest eigenvalue of M.

Lem. 2 implies that for the choice of $\alpha(t) = C/\sqrt{t}$ for an appropriate C, the error scales as $L\sqrt{\frac{\min(d,n)\log(n^{1/2}dT)}{T(1-\max_k \sigma_2(P^k))}}$. The error scales as $T^{-1/2}$, which is a common factor generally observed when studying the convergence of stochastic communication schemes. The term $1/1-\sigma_2(P^k)$ determines how quickly nodes come to a consensus in coordinate k. When $P^k = P$, we do not obtain the factor $\min(d,n)$ retrieving the results of the DDA algorithm of [4].

Lemma 3. Round robin scheme: *For the settings in Thm. 1, the DCDA algorithm under the round robin $m-$coordinate sharing scheme is such that*

$$f(\hat{x}_i(T)) - f(x^*) \leq \frac{\psi(x^*)}{T\alpha(T)} + \frac{L^2}{T}\sum_{t=1}^{T}\alpha(t-1)\left(10 + \frac{12d\log 2\sqrt{n}T}{m(1-\sigma_2(P))}\right). \quad (6)$$

With an appropriate choice of the step size, the error in Lem. 3 scales as $L\sqrt{\frac{d\log(nT)}{mT(1-\sigma_2(P))}}$. Thus, we would need twice the amount of time to achieve a fixed error ϵ if we transmit half the number of coordinates m at each time instant.

Lemma 4. Randomized scheme: *For the settings in Thm. 1, the DCDA algorithm under the randomized coordinate sharing scheme is such that, with probability greater than $1-\delta$,*

$$f(\hat{x}_i(T)) - f(x^*) \leq \frac{\psi(x^*)}{T\alpha(T)} + \frac{L^2}{T}\sum_{t=1}^{T}\alpha(t-1)\left(10 + 18\frac{\min(d,n)\log Tdn^{1/3}/\delta}{1-\max_k \sigma_2(\mathbb{E}[P^k(t)^2])}\right). \quad (7)$$

The result in Lem. 4 is similar to the static coordinate sharing case with the expected doubly stochastic sharing matrix used. Consider the specific case where the nodes collectively share coordinate k with all other nodes with probability ρ. In this case, $P^k(t) = \frac{1}{n}\mathbf{1}\mathbf{1}^\intercal$ with probability ρ, else $P^k(t) = \mathbf{I}$. In this case, the error scales as $L\sqrt{\frac{\log Tdn/\delta}{\rho T}}$ with high probability. Similar to the round robin case, the analysis shows an inverse dependence between the number of coordinates shared and the time needed for convergence.

3.1. Variations of the DCDA algorithm

In this section we study three variations of the DCDA scheme as introduced in Sec. 2. First, we consider the case in which each node does not have access to the exact gradient of its local function but instead obtains a noisy estimate of this value. The DCDA algorithm for the stochastic gradient setting simply uses the stochastic gradient in place of the actual gradient. Convergence is studied under some mild assumptions on the noisy gradient value.

Assumption 1. *Assume $\mathcal{F}_t$ to be the σ-field that contains all information known by all nodes until time t and let $g'(t)$ be the stochastic gradient at time t. Further assume that:*
- *the stochastic gradient $g_i'(t)$ is an unbiased estimate of the actual gradient, i.e. $\mathbb{E}[g_i'(t)|\mathcal{F}_t] \in \partial f_i(x_i(t))$,*
- *the stochastic gradient is bounded. $\|g_i'(t)\|_* \leq L$, and*
- *the set $\mathcal{X}$ satisfies $\|x - x'\| \leq R\ \forall x, x' \in \mathcal{X}$.*

Lemma 5. Stochastic gradient DCDA algorithm: *For the settings in Thm. 1 and under the assumptions in Ass. 1, the stochastic gradient DCDA algorithm is such that, with probability $1-\delta$,*

$$f(\hat{x}_i(T)) - f(x^*) \leq (4) + LR\sqrt{\frac{1}{T}8\log\frac{1}{\delta}}. \quad (8)$$

From Lem. 5, we conclude that the scaling of the error of the stochastic gradient DCDA algorithm is the same as the DCDA algorithm.

Let us next consider the noisy communication scenario. More precisely, message $z_j(t)$ transmitted at time t from node j to node i suffers from additive noise $n_{ij}(t)$, i.e. node i observes $u_{ij}(t) = z_j(t) + n_{ij}(t)$

The DCDA algorithm for the noisy communication setting uses the noisy dual variable estimate $u_{ij}(t)$ instead of the actual value $z_j(t)$. Convergence is shown under some assumptions on the noise sequence and for the static sharing scheme.

Lemma 6. Noisy communication static staring scheme DCDA algorithm: *Consider the settings in Thm. 1 where the function is L-Lipschitz with respect to the ℓ_2-norm. Further assume that there exists R such that $\sup_{x,x'\in\mathcal{X}}||x-x'||_2 \leq R$. Under the assumptions that each $n_{ij}(t)$ has independent zero-mean sub-Gaussian components of power γ^2/d, the noisy communication static sharing DCDA algorithm is such that, with probability $1-\delta$,*

$$f(\hat{x}_i(T)) - f(x^*) \leq (5) + \gamma(R+2L)\sqrt{\frac{2\log\frac{3}{\delta}}{nT}} + \sum_{t=1}^{T}\alpha(t-1) \times \left(\frac{\gamma^2(1+\sqrt{8\log\frac{3}{\delta}})}{ndT} + \frac{3L}{T}\sqrt{\frac{2\gamma^2\log\frac{6Tnd}{\delta}}{(1-\max_k\sigma_2(P^k)^2)}}\right). \quad (9)$$

Finally, we consider the case in which the communication among nodes is quantized using infinite-level uniform quantization. At each time step, a node broadcasts the quantized scaled dual variable update

$$[u_i(t)]_k = \left\lfloor \frac{[z_i(t)]_k - [z_i(t-1)]_k}{s(t)} + \Delta_{ik}(t)\right\rfloor, \quad (10)$$

where $\Delta_{ik}(t) \overset{\text{iid}}{\sim} \mathcal{U}([-1/2,+1/2])$ is dither used to guarantee that the quantization noise is uniformly distributed in the interval $[-1/2,+1/2]$, while $s(t) > 0$ is a zooming sequence that converges to zero and is known a priori to all nodes. The dual update operation is replaced by:

$$[z_i(t+1)]_k = \left[z_i(t) + g_i(t) - g_i(t-1) + \sum_j P_{ij}^k(t)s(t)u_j(t)\right]_k.$$

Lemma 7. Quantized communication static coordinate scheme DCDA algorithm: *Consider the settings in Thm. 1 where the function is L-Lipschitz with respect to the ℓ_2-norm. Furthermore define*

$$\nu(t) = \max_k \sum_{r=0}^{t} s^2(r)\sigma_2(P^k)^{2(t-r+1)}. \quad (11)$$

Under these assumptions, the quantized communication static sharing DCDA is such that

$$f(\hat{x}_i(T)) - f(x^*) \leq (5) + R\sqrt{\widehat{s^2}(T)\frac{\log 1/\delta}{T}} + \sum_{t=1}^{T}\alpha(t-1) \times \left(\frac{2s(t)L + s^2(t)}{nT} + \frac{3L}{T}\sqrt{2\nu(t)\log(2Tnd/\delta)}\right). \quad (12)$$

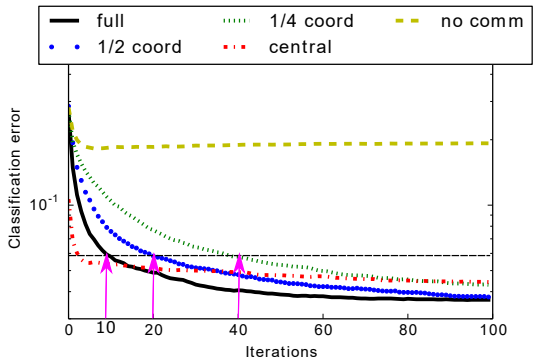

Fig. 1: Classification performance with distributed SVM

.

4. NUMERICAL SIMULATIONS

For numerical simulations, we consider the scenario in which the function $f_i(x)$ arises from evaluating a common loss function ℓ over a set of m local measurements $\{z_{ij}\}_{j=1}^m$: correspondingly we have

$$f(x) = \sum_{i=1}^{n} f_i(x) = \sum_{i=1}^{n}\sum_{j=1}^{m} \ell(x, z_{ij}). \tag{13}$$

Support Vector Machine (SVM): In the first case, we look at using support vector machines for classification. Each datapoint at a local node consists of a label l_{ij} uniformly drawn from $\{-1, 1\}$ and the data point $z_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(\mu_{l_{ij}}, \Sigma)$. The linear SVM algorithm finds a hyperplane that separates data drawn from the two distributions with the maximum margin

$$\ell(x, (z_{ij}, l_{ij})) = \frac{1}{2d}||x||_2^2 + C\sum_{j=1}^{m} \max\left(1 - l_{ij}x^T z_{ij}; 0\right).$$

In Fig. 1 we plot the simulation results for $X \in \mathbb{R}^{30}$, $n = 10$ and $m = 10$ and full network connectivity. At each instant in time, the nodes collectively sample a certain fraction Γ of their coordinates to share. For this scenario, we compare the performance for $\Gamma \in \{0, 1/2, 1/4, 1\}$. The performance for the centralized SVM algorithm is also plotted for comparison. As can be observed, without any communication, nodes reach a suboptimal solution. To reach 90% classification accuracy rate, nodes take twice as long if they share only half their coordinates.

Linear Regression: Next, we investigate the effect of noisy communication and stochastic gradient in the DCDA algorithm for the classic linear regression problem. We consider the case where $x \in \mathbb{R}^{30}$, $n = 10$ and $m = 20$ and a fully connected network. Note that local measurements consist of random normal measurement vectors a_{ij} and the measurement z_{ij}. For the noisy communication scenario, each node observes $z_{ij} = A_{ij}x + n_{ij}$,for $n_i = [n_{i1}, \ldots, n_{id}] \overset{\text{iid}}{\sim} \mathcal{N}(0, \Sigma)$ and $f_i(x) = \ell(x, (A_{ij}, z_{ij})) = \frac{1}{2}\|A_{i:}x - z_{i:}\|_2^2$ (with a slight abuse of notation). In the stochastic gradient case, nodes form mini-batches of size 4 as opposed to using all 20 data points for each iteration. As can be seen in Fig. 2, there is minimal loss in performance from using stochastic gradients or when noise is added to the coordinates being shared. This suggests that using less computationally expensive stochastic gradients, or quantization effects creating additive noise may not significantly alter performance.

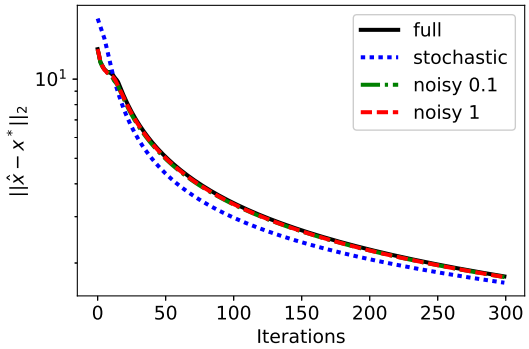

Fig. 2: Impact of stochastic gradients, noisy communications on performance of DCDA for linear regression

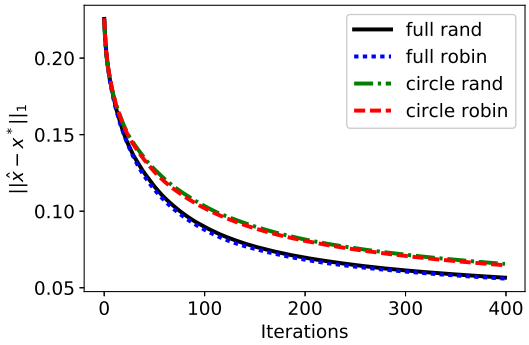

Fig. 3: Robust regression performance.

Robust Regression: Finally, we consider the robust regression problem in which each node observes $z_{ij} = A_{ij}x + (1 - b_{ij})o_{ij} + b_{ij}n_{ij}$ where x is in the unit simplex ($[x]_i \geq 0, \|x\|_1 = 1$) and b_{ij} is a binomial noise that modulates between a large outlier Gaussian noise o_{ij} or smaller additive Gaussian noise n_{ij}. The ℓ_1 penalty is used as the loss function or $f_i(x) = \ell(x, (A_{ij}, z_{ij})) = \|A_{i:}x - z_{i:}\|_1$. For this problem, we consider an entropic proximal function that ensures that the estimate is in the probability simplex and minimize the ℓ_1-norm. In the simulation, we compare the round robin scheme and the randomized scheme where the amount of communication is kept equal. Both these schemes are compared for the fully connected network as well as a circle topology where nodes are connected to the closest neighbor on each side. Estimate $x \in \mathbb{R}^{20}$ and $m, n = 10$. The performance is presented in Fig. 3. The performance of the fully connected network is better than the circle topology because more communication is taking place, allowing estimates to quickly travel through the network. No significant difference in performance between the randomized and round robin schemes is observed.

5. CONCLUSION

We considered the problem of distributed optimization with limited communication in which nodes collectively solve a convex optimization problem but have a limitation on the transmission among neighbors. We proposed a distributed coordinate dual averaging algorithm for this problem and analyzed its performance. We showed that the time required to achieve a fixed accuracy would double if the rate limitation in messages between nodes were halved. We also show convergence in the scenario of stochastic gradients, noisy and quantized communication.

A. REFERENCES

[1] J. N. Tsitsiklis, "Problems in decentralized decision making and computation." Massachusetts Institute of Tech Cambridge Lab for Information and Decision Systems, Tech. Rep., 1984.

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

[3] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, p. 1291, 2011.

[4] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2012.

[5] D. Jakovetić, J. M. Moura, and J. Xavier, "Distributed nesterov-like gradient algorithms," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5459–5464.

[6] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5445–5450.

[7] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *International Conference on Machine Learning (ICML)*, 2014, pp. 1701–1709.

[8] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[9] J. Wang and N. Elia, "A control perspective for centralized and distributed convex optimization," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 3800–3805.

[10] P. Yi and Y. Hong, "Quantized subgradient algorithm and data-rate analysis for distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 380–392, 2014.

[11] J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar, "An asynchronous parallel stochastic coordinate descent algorithm," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 285–322, 2015.

[12] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[13] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 873–881.

[14] K. I. Tsianos and M. G. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 1067–1072.

[15] K. Tsianos, S. Lawlor, and M. G. Rabbat, "Communication/computation tradeoffs in consensus-based distributed optimization," in *Advances in neural information processing systems*, 2012, pp. 1943–1951.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "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, 2011.

[17] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization." *IEEE Trans. Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[18] M. Rao, S. Rini, and A. Goldsmith, "Distributed convex optimization with limited communications," *arXiv preprint arXiv:1810.12457*, 2018.