52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Online Distributed Optimization via Dual Averaging

Saghar Hosseini, Airlie Chapman and Mehran Mesbahi

***Abstract*—This paper presents a regret analysis on a distributed online optimization problem computed over a network of agents. The goal is to distributively optimize a global objective function which can be decomposed into the summation of convex cost functions associated with each agent. Since the agents face uncertainties in the environment, their cost functions change at each time step. We extend a distributed algorithm based on dual subgradient averaging to the online setting. The proposed algorithm yields an upper bound on regret as a function of the underlying network topology, specifically its connectivity. The regret of an algorithm is the difference between the cost of the sequence of decisions generated by the algorithm and the performance of the best fixed decision in hindsight. A model for distributed sensor estimation is proposed and the corresponding simulation results are presented.**

***Index Terms*—Online Optimization, Distributed Algorithms, Distributed Estimation**

I. Introduction

An intrinsic aspect of the world around us is the existence of complex networks such as biological, chemical and social networks, and our technological world contains many designed networks such as the internet, power grids, and robotic networks. In recent years, the area of networked systems has received extensive attention and the manipulation and monitoring of such networks is of increasing interest. By increasing the size and complexity of these systems, decentralized control schemes are desired, reducing data transmission rates and ensuring robustness in the presence of local failures. In addition, when there is lack of access to centralized information, these control algorithms are essential to operate agents based on their locally available information.

Recently, there has grown an extensive literature on distributed convex optimization. Some of the recent distributed algorithms are based on the subgradient method [4], [5], [6], [7]. In these works the cost function is fixed while the topology of network is allowed to vary. However, uncertainties in the environment often have a significant impact on the cost functions making it difficult to set up a tractable optimization problem. One way to improve the robustness of the algorithm is via a stochastic framework [8], [9], [10], where the probability distribution of uncertain variables is known a priori. One such approach has been adopted by Duchi *et al.* [7] which examined this problem using stochastic subgradient method.

Despite its many successes, the stochastic optimization method fails at addressing the dynamic aspect of the problem. Online learning is an attempt to overcome this obstacle where the uncertainty in the system is demonstrated by an arbitrarily varying cost function. The cost function is unknown, even without a probabilistic assumption, at the time the relevant decision is made. This generalization of the framework has proved to have a significant impact on modern machine learning [11], [12], [13]. One standard metric to measure the performance of online algorithms is called *regret*. Regret measures the difference between the incurred cost and the cost of the best fixed decision in hindsight. Consequently, a good algorithm is one where the average regret approaches zero.

The distributed online optimization and its application in multi-agent systems has not been studied at large by the scientific community. Yan *et al.* in [14] introduced a decentralized online optimization based on dual averaging in which the agents are interacting over a weighted strongly connected directed graph. In addition, the regret bound of $O(\log T)$ is found for a strongly convex cost function and does not depend on the network topology. Considering an undirected chain graph and fixed-radius neighborhood information structure, Raginsky *et al.* [15] proposed an online algorithm for distributed optimization based on sequential updates proving a regret bound of $O(\sqrt{T})$.

The contribution of this paper is two folds; First, we discuss the extension of the work by Duchi *et al.* [7] on distributed optimization to the online setting, generalized to strongly connected network topology and provide regret bounds as a function of connectivity in the underlying network. These bounds highlight the link between the network topology and online algorithms and can thus be used for online design of high performance networks. Second, we demonstrate the applicability of the corresponding online distributed optimization for distributed sensor estimation. A comparison of network topologies illustrates some of the favorable network features in the online setting.

The organization of the paper is as follows. The paper commences with the introduction of notation, graphs, and regret. In §III, the optimization problem formulation is presented followed by the description of the algorithm and regret analysis in §IV. Then, in §V a distributed sensor estimation problem is solved based on the algorithm, and the simulation results are presented to reinforce the analysis. Finally, §VI provides our concluding remarks.

II. Background and Preliminaries

We provide a brief background on constructs that will be used in this paper.

The research of the authors was supported by the ONR grant N00014-12-1-1002 and AFOSR grant FA9550-12-1-0203-DEF. The authors are with the Department of Aeronautics and Astronautics, University of Washington, WA 98105. Emails: {saghar, airliec, mesbahi}@uw.edu.

For column vector $v \in \mathbb{R}^p$, v_i or $[v]_i$ denotes the ith element, e_i denotes the column vector which contains all zero entries except $[e_i]_i = 1$. The vector of all ones will be denoted by $\mathbf{1}$. For matrix $M \in \mathbb{R}^{p\times q}$, $[M]_{ij}$ denotes the element in its ith row and jth column. A doubly stochastic matrix P is a non-negative matrix with $\sum_{i=1}^n P_{ij} = 1$ and $\sum_{j=1}^n P_{ij} = 1$. For any positive integer n, the set $\{1, 2, ..., n\}$ is presented by $[n]$. The 2-norm is denoted by $||.||_2$.

A. Graphs

A succinct way to represent the interactions of agents in a network is through a graph. A weighted graph $\mathcal{G} = (V, E, W)$ is defined by a node set V with cardinality n, the number of nodes in the graph which represent the agents in the network, and an edge set E comprising of pairs of nodes which represent the agents interactions, i.e., agent i affects agent j's dynamics if there an edge from i to j, i.e., $(i, j) \in E$. In addition, a function $W : E \to R$ is given that associates a weight $w_{ji} \in W$ to every edge $(i, j) \in E$. A graph is undirected if $(i, j) \in E$ implies $(j, i) \in E$. The adjacency matrix $A(\mathcal{G})$ is a matrix representation of $\mathcal{G}$ with $[A(\mathcal{G})]_{ji} = w_{ji}$ for $(i, j) \in E$ and $[A(\mathcal{G})]_{ji} = 0$, otherwise. A graph $\mathcal{G}$ is strongly connected if between every pair of distinct vertices there exists a directed path. For a graph $\mathcal{G}$, we can define d_i as the summation of weights of incoming edges of node i. In particular, d_i is the weighted in-degree[1] of i and $d_{max} = \max_{i\in V} d_i$. When for every node $i \in V$ the weighted in-degree and out-degree are equal, the directed weighted graph $\mathcal{G}$ is called balanced. In addition, a graph $\mathcal{G}$ is balanced if and only if $\mathbf{1}$ is a left eigenvector of $L(\mathcal{G})$, where $L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G})$ is the graph Laplacian and $\Delta(\mathcal{G})$ is the diagonal matrix of d_i's. Based on the construction of weighted directed graph Laplacian, every graph $\mathcal{G}$ has a right eigenvector of $\mathbf{1}$ [3].

There are many families of graphs; the family of p-random graphs (Erdős–Rényi) are constructed by creating an edge $(i, j) \in E$ in the graph with probability p for all possible edges.

B. Regret

In online optimization, an online algorithm is used to generate a sequence of decisions $\{x(t)\}_{t=1}^T$. The number of iterations is denoted by T which is unknown to the online player. At iteration t, after committing to $x(t)$, a previously unknown convex cost function f_t is revealed, and a loss $f_t(x(t))$ is incurred. There are two main categories for the feedback available to the player. In the full information model, after revealing loss $f_t(x(t))$, all information about the function f_t is observable by the player. The other category is the bandit model, in which the player is only able to observe the loss $f_t(x(t))$ [12]. In this paper, the loss $f_t(x(t))$ and its gradient are available to the online player.

The goal of the online algorithm is to ensure that the difference between the total cost and the cost of the best fixed decision is small. In addition, the best fixed decision x^* is chosen with the benefit of hindsight. Formally, the difference between these two cost when running $t = 1, 2, ..., T$ iterations is called the regret of the online algorithm

$$R_T = \sum_{t=1}^{T} \left(f_t(x(t)) - f_t(x^*)\right). \tag{1}$$

Regret is a standard measure for the performance of learning algorithms. An algorithm performs well if its regret is sublinear as a function of T, i.e. $\lim_{T\to\infty} R_T/T = 0$. This implies that on the average, the algorithm performs as well as the best fixed strategy in hindsight independent of the adversary's moves. Further discussion on online algorithms and their regret can be found in [16], [17], [18].

III. Problem Statement

In this section a parallel decision processes for a distributed system is considered in which a large number of agents cooperatively optimize a global objective function. Let graph $\mathcal{G} = (V, E, W)$ represent the communication constraint on the system in which $V = [n]$. Each node $i \in V$ is an agent and an edge $(i, j) \in E$ indicates a communication link between agent i and j . Thus, the set of agents that are communicating with agent i is defined as the neighborhood set $N(i) = \{j \in V|(i, j) \in E\}$.

The global objective is to minimize

$$f_t(x) = \frac{1}{n}\sum_{i=1}^{n} f_{t,i}(x) \quad \text{subject to} \;\; x \in \chi, \tag{2}$$

where $f_{t,i}(x_i) : \mathbb{R}^d \to \mathbb{R}$ is a convex cost function associated with agent $i \in V$ and evolves over time steps in an unpredictable manner. In other words, at time step t, each agent i executes $x_i \in \chi$ based on the local information available to it and to its neighbors and then an "oracle" announces $f_t(x_i)$. The optimization variable $x_i \in \mathbb{R}^d$ belongs to a closed convex set $\chi \subseteq \mathbb{R}^d$ and represents the local decision made by agent i.

The general definition of regret is presented in (1) with the best fixed decision provided with the full knowledge of f_t in hindsight denoted as x^*. Two types of regret are introduced to analyze the performance of the system in this paper. The regret due to agent i's action is

$$R_T(x^*, x_i) = \sum_{t=1}^{T} \left(f_t(x_i(t)) - f_t(x^*)\right), \tag{3}$$

which is the cumulative penalty agent i pays because of its decisions on the global cost sequence $\{f_t\}$. The running average regret is another measure for the analysis of the algorithm

$$R_T(x^*, \hat{x}_i) = \sum_{t=1}^{T} \left(f_t(\hat{x}_i(t)) - f_t(x^*)\right), \tag{4}$$

where $\hat{x}_i(t) = \frac{1}{t}\sum_{s=1}^{t} x_i(s)$ is the running average of $x_i(t)$ by time t. In this case, the network pays its penalty on a running average vector $\hat{x}(t)$.

[1]Note that the weighted in-degree of a vertex v_i is defined as $d_i = \sum_{\{j|(j,i)\in E\}} w_{ij}$.

IV. Main Result

To solve the online optimization problem proposed in §III, we adapt Nesterov's dual averaging algorithm [19]. The algorithm sequentially updates the state variable $x_i(t)$ and a working variable $z_i(t)$ for each agent. The update itself is based on a provided local gradient of a cost function $\nabla f_{t,i}(x_i(t))$ denoted as $g_i(t)$. The centralized form of the algorithm appears as a gradient descent method followed by a projection step onto the constraint set χ, specifically,

$$z(t+1) = z(t) + g(t),$$

where $g(t) = \nabla f_t(x(t))$; then

$$x(t+1) = \prod_{\chi}^{\psi} \left(z(t+1), \alpha(t)\right),$$

with $\prod_{\chi}^{\psi}(\cdot)$ as the projection operator onto χ,

$$\Pi_{\chi}^{\psi}(z(t), \alpha(t)) = \arg\min_{x \in \chi} \left\{ \langle z(t), x \rangle + \frac{1}{\alpha(t)} \psi(x) \right\}, \quad (5)$$

where $\alpha(t)$ is a non-increasing sequence of positive functions and $\psi(x) : \chi \to \mathbb{R}$ is a proximal function. The standard dual averaging algorithm uses proximal function $\psi(x)$ to avoid wide oscillation in the projection step. Without loss of generality, ψ is assumed to be strongly convex with respect to $\|.\|$, $\psi \geq 0$, and $\psi(0) = 0$.

The distributed algorithm can be considered as an approximated gradient descent using $g_1(t), g_2(t), \ldots, g_n(t)$ in replacement of $g(t)$. The approximation is attained locally by an agent i via a convex combination of information provided by its neighbors $N(i)$. This can be represented compactly as a doubly stochastic matrix $P \in \mathbb{R}^{n \times n}$ which preserves the zero structure of the Laplacian matrix $L(\mathcal{G})$. It is clear that for all agents to have access to each cost function $f_{t,i}$ there must be a path from every agent i to every other agent. Consequently, a minimum requirement is that graph $\mathcal{G}$ must be strongly connected. The following proposition provides a method to construct a doubly stochastic matrix P of the required form from $L(\mathcal{G})$.

Proposition 1. *If $\mathcal{G}$ is balanced then $P = I - \frac{1}{\epsilon} L(\mathcal{G})$ is doubly stochastic, where $\epsilon \in (d_{\max}, \infty)$. If $\mathcal{G}$ is strongly connected then the matrix $P = I - \frac{1}{\epsilon} diag\,(v)\, L(\mathcal{G})$ is doubly stochastic, where $v^T L\,(\mathcal{G}) = 0$ with positive vector $v = [v_1, v_2, \ldots, v_n]^T$ and $\epsilon \in (\max_{i \in V}(v_i d_i), \infty)$.*

Proof: For a balanced $\mathcal{G}$ and $\epsilon \in (d_{\max}, \infty)$, $\frac{1}{\epsilon} L(\mathcal{G})$ preserves the sign structure of $L(\mathcal{G})$ with diagonal elements in $(0, 1)$ and off-diagonal elements in $[-1, 0]$. Consequently, $I - \frac{1}{\epsilon} L(\mathcal{G})$ is a non-negative matrix. As $\mathcal{G}$ is balanced, $L(\mathcal{G})\mathbf{1} = \mathbf{1}^T L(\mathcal{G}) = \mathbf{0}$, $\left(I - \frac{1}{\epsilon} L(\mathcal{G})\right) \mathbf{1} = \mathbf{1}$ and $\mathbf{1}^T \left(I - \frac{1}{\epsilon} L(\mathcal{G})\right) = \mathbf{1}^T$, and so $I - \frac{1}{\epsilon} L(\mathcal{G})$ is doubly stochastic.

A strongly connected graph $\mathcal{G}$ can always be reweighted to form a balanced graph $\widehat{\mathcal{G}}$ with Laplacian, $L(\widehat{\mathcal{G}}) = \text{diag}\,(v)\, L(\mathcal{G})$ [20]. The maximum degree of $\widehat{\mathcal{G}}$ is then $\max_{i \in V}(v_i d_i)$. Combined with the previous result the proposition follows. ∎

The Online Distributed Dual Averaging (ODD) algorithm is presented in Algorithm 1. The projection function used in this algorithm is defined as in (5).

Algorithm 1: Online Distributed Dual Averaging (ODD)

```
1 for t = 1 to T do
2     Adversary reveals f_t(t) = {f_{t,i}(t); for ∀i = 1, ..., n}
3     Compute subgradient g_i(t) ∈ ∂f_{t,i}(x_{t,i})
4     for Each Agent i do
5         z_i(t+1) = Σ_{j∈N(i)} P_{j,i} z_j(t) + g_i(t)
6         x_i(t+1) = Π_χ^ψ(z_i(t+1), α(t))
7         x̂_i(t+1) = 1/(t+1) Σ_{s=1}^{t+1} x_i(s)
8     end
9 end
```

Before proceeding to the main result we provide a few preliminary remarks and definitions. Any convex function $f_{t,i}$ on a compact domain is uniformly L-Lipschitz with respect to $\|.\|_2$.[2] In order to take advantage of the standard dual averaging algorithm properties in our regret analysis, the sequences of $\bar{z}(t)$ and $\bar{g}(t)$ are defined as

$$\bar{z}(t) = \frac{1}{n} \sum_{i=1}^{n} z_i(t), \quad \bar{g}(t) = \frac{1}{n} \sum_{i=1}^{n} g_i(t), \quad (6)$$

where these sequences are the average of dual variable and subgradient over all agents in ODD. Thus based on (6), the following update rule is introduced

$$\bar{z}(t+1) = \bar{z}(t) + \bar{g}(t). \quad (7)$$

The update in (7) is similar to the standard dual averaging algorithm where the primal variable is

$$y(t+1) = \Pi_{\chi}^{\psi}(\bar{z}(t+1), \alpha(t)). \quad (8)$$

The main result of this paper can now be stated as follows.

Theorem 2. *Given the sequence of $x_i(t)$ and $z_i(t)$ generated by line 5 and line 6 in* Algorithm 1 *for all $i \in [n]$ where $\psi(x^*) \leq R^2$ and $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T(x^*, x_i) \leq \left(\frac{R^2}{k} + kL^2 \left(\frac{6\sqrt{n}}{1 - \sigma_2(P)} + 13 \right) \right) \sqrt{T}. \quad (9)$$

Proof: The proof follows the general approach adopted in [7], extended to the online setting with an improved performance bound. Consider an arbitrary fixed decision $x^* \in \chi$ and a sequence $y(t)$ generated by (8) are given. From the L-Lipschitz continuity of $f_{t,i}$'s and the definition of regret in (3), the regret is bounded as

$$R_T(x^*, x_i) \leq \sum_{t=1}^{T} \left(f_t(y(t)) - f_t(x^*) + L\|x_i(t) - y(t)\| \right). \quad (10)$$

[2] Note that the L-Lipschitz continuity is defined as $|f_{t,i}(x) - f_{t,i}(y)| \leq L\|x - y\|_2$ for $\forall x, y \in \chi$

In addition, based on the convexity of $f_{t,i}$'s, we have

$$\sum_{t=1}^{T}\left(\frac{1}{n}\sum_{i=1}^{n} f_{t,i}(x_i(t)) - f_t(x^*)\right) \leq \sum_{t=1}^{T}\left(\frac{1}{n}\sum_{i=1}^{n}\langle g_i(t), x_i(t)-x^*\rangle\right) \tag{11}$$

where $g_i(t) \in \partial f_{t,i}(x_i(t))$ is the subgradient of $f_{t,i}$ at $x_i(t)$. we can now express the regret bound based on (11):[3]

$$R_T(x^*, x_i) \leq \sum_{t=1}^{T}(\frac{1}{n}\sum_{i=1}^{n}\langle g_i(t), x_i(t)-x^*\rangle + \frac{L}{n}\sum_{i=1}^{n}\|x_i(t)-y(t)\| + L\|x_i(t)-y(t)\|). \tag{12}$$

The first term on the right hand side of (12) is expanded as

$$\sum_{t=1}^{T}\left(\frac{1}{n}\sum_{i=1}^{n}\langle g_i(t), x_i(t)-x^*\rangle\right) = \sum_{t=1}^{T}(\frac{1}{n}\sum_{i=1}^{n}\langle g_i(t), x_i(t)-y(t)\rangle + \frac{1}{n}\sum_{i=1}^{n}\langle g_i(t), y(t)-x^*\rangle). \tag{13}$$

Now, we need to bound the terms on the right hand side of (13). The first term is bounded based on the convexity and L-Lipschitz conditions on $f_{t,i}$.[4] In other words,

$$\langle g_i(t), x_i(t)-y(t)\rangle \leq L\|x_i(t)-y(t)\|. \tag{14}$$

Since $x_i(t)$ and $y(t)$ are the projection of $z_i(t)$ and $\bar{z}(t)$ respectively, the Lipschitz continuity of $\Pi_{\mathcal{X}}^{\psi}(.,\alpha)$ presented in Lemma 4 imposes a bound on $\|x_i(t)-y(t)\|$:

$$\|x_i(t)-y(t)\| \leq \alpha(t)\|\bar{z}(t)-z_i(t)\|_*, \tag{15}$$

where $||.||_*$ is the dual norm.[5] Therefore, using the bound in Lemma 5 and noting that $\|g_i(t)\|_* \leq L$, we can write (13) as

$$\sum_{t=1}^{T}\left(\frac{1}{n}\sum_{i=1}^{n}\langle g_i(t), x_i(t)-x^*\rangle\right) \leq \frac{L}{n}\sum_{t=1}^{T}\sum_{i=1}^{n}\alpha(t)\|\bar{z}(t)-z_i(t)\|_* + \frac{L^2}{2}\sum_{t=2}^{T}\alpha(t-1) + \frac{1}{\alpha(T)}\psi(x^*). \tag{16}$$

[3]Note that we can rephrase the first term on the right hand side of (10) as $f_t(y(t)) - f_t(x^*) = \left(\frac{1}{n}\sum_{i=1}^{n} f_{t,i}(x_i(t)) - f_t(x^*)\right) + \left(\frac{1}{n}\sum_{i=1}^{n}[f_{t,i}(y(t)) - f_{t,i}(x_i(t))]\right)$. In addition, the L-Lipschitz continuity of of $f_{t,i}$'s implies that $f_{t,i}(y(t)) - f_{t,i}(x_i(t)) \leq L\|x_i(t)-y(t)\|$.

[4]Note that convexity implies $\langle g_i(t), y(t)-x_i(t)\rangle \leq f_{t,i}(y(t)) - f_{t,i}(x_i(t))$ and L-Lipschitz continuity implies $f_{t,i}(y(t)) - f_{t,i}(x_i(t)) \leq L\|x_i(t)-y(t)\|$.Therefore, $||g_i||_* \leq L$ and we can deduce (14).

[5]The dual norm of a vector x is defined as $||x||_* = \sup_{||y||=1}\langle x, y\rangle$.

Thus, (12), (15), and (16) imply

$$R_T(x^*, x_i) \leq \frac{L^2}{2}\sum_{t=2}^{T}\alpha(t-1) + \frac{1}{\alpha(T)}\psi(x^\star) + L\sum_{t=1}^{T}\alpha(t)\left(\|\bar{z}(t)-z_i(t)\|_* + \frac{2}{n}\sum_{i=1}^{n}\|\bar{z}(t)-z_i(t)\|_*\right). \tag{17}$$

Lemma 6 imposes an upper bound on the last term on the right hand side of (17). Thus, using (25) the regret is further bounded as

$$R_T(x^*, x_i) \leq \frac{L^2}{2}\sum_{t=1}^{T-1}\alpha(t) + \frac{1}{\alpha(T)}\psi(x^\star) + L\left(\frac{3\sqrt{n}L}{1-\sigma_2(P)} + 6L\right)\sum_{t=1}^{T}\alpha(t). \tag{18}$$

Since $\psi(x^\star) \leq R^2$ and $\alpha(t) = k/\sqrt{t}$, applying the integral test on the first and last terms in (18) leads to (9) and the theorem follows. ■

The result's significance is the validation of "good" performance through sublinear regret as well as highlighting the importance of the underlying topology through $\sigma_2(P)$. Further, when P is formed as proposed in Proposition 1, $1-\sigma_2(P)$ is proportional to $\lambda_2(\mathcal{G})$, where $\lambda_2(\mathcal{G})$ is the second smallest eigenvalue of the graph Laplacian $L(\mathcal{G})$ and a well known measure of network connectivity. Consequently, high network connectivity promotes good performance of the algorithm.

Now, the running average regret analysis can be stated exhibiting similar dependence on the parameters of the graph Laplacian.

Corollary 3. *Given the sequence of where $\hat{x}_i(t)$ is generated by line 7 in* Algorithm 1 *for all $i \in [n]$ where $\psi(x^*) \leq R^2$ and $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T(x^*, \hat{x}_i) = 2\left(\frac{R^2}{k} + kL^2\left(\frac{6\sqrt{n}}{1-\sigma_2(P)} + 13\right)\right)\sqrt{T}.$$

Proof: Since the cost function $f_t(x(t)$ is convex, $f_t(\hat{x}_i(t)) \leq \frac{1}{t}\sum_{s=1}^{t} f_t(x_i(s))$. Based on Theorem 2, we have

$$f_t(\hat{x}_i(t)) - f_t(x^*) \leq \frac{1}{t}R_t(x^*, x_i). \tag{19}$$

Thus, the running average regret is bounded by the regret $R_t(x^*, x_i)$ as

$$R_T(x^*, \hat{x}_i) \leq \sum_{t=1}^{T}\left(\frac{1}{t}R_t(x^*, x_i)\right). \tag{20}$$

The regret bound is given in (9) which implies[6]

$$R_T(x^*, \hat{x}_i) \leq \left(\frac{R^2}{k} + kL^2\left(\frac{6\sqrt{n}}{1-\sigma_2(P)} + 13\right)\right) \times \left(2\sqrt{T}-1\right). \tag{21}$$

The corollary now follows from (21). ■

[6]Note that $\frac{1}{\sqrt{t}}$ is a non increasing positive function and the integral test leads to $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\sqrt{T}-1$.

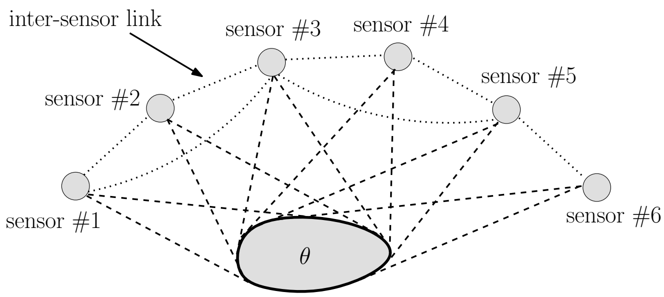

Figure 1. A graphical representation of a distributed sensor network.

V. Example - Distributed Sensor Network

In this section, we implement the online ODD algorithm in the context of a distributed sensor network.

Consider a distributed sensor network estimating at time k, a random vector $\theta_k \in \chi = \left\{\theta \in \mathbb{R}^d | \, \|\theta\|_2 \leq \theta_{\max}\right\}$, a closed convex set containing zero. The ith sensor observes θ_k at time k through an observation vector $z_{i,k}(\theta) : \mathbb{R}^d \to \mathbb{R}^{p_i}$, which is time-varying due to the sensor's susceptibility to unknown errors. The sensor is modeled as $h_i(\theta) = H_i\theta$ where $H_i \in \mathbb{R}^{p_i \times d}$ is the observation matrix of sensor i, and $\|H_i\|_1 \leq h_{\max}$, for all sensors i. Adopting a least square point of view, the objective is to find the argument $\bar{\theta}$ that minimizes the cost function

$$f_{i,k}(\bar{\theta}) \;=\; \frac{1}{2}\left\|z_{i,k}(\theta) - h_{i,k}(\bar{\theta})\right\|_2^2 = \frac{1}{2}\left\|z_{i,k}(\theta) - H_i\bar{\theta}\right\|_2^2,$$

a convex function with subgradient

$$\partial f_{i,k}(\bar{\theta}) = -H_i^T\left(z_{i,k}(\theta) - H_i\bar{\theta}\right).$$

Figure 1 graphically summarizes the problem setup. [7] The overall cost at time k is defined as, $f_k(\bar{\theta}) = \frac{1}{n}\sum_{i=1}^n f_{i,k}\left(\bar{\theta}\right)$, and over time as $f(\bar{\theta}) = \sum_{k=1}^T f_k(\bar{\theta})$.

Consider $f_k = f_{k+1}$ for all $k = 1, 2, \ldots, T$ and a noisy observation $z_{i,1}(\theta) = H_i\theta + v_i$, where v_i is independent white noise. The centralized optimal solution is

$$\theta^* = \left(\sum_{i=1}^n H_i^T H_i\right)^{-1}\left(\sum_{i=1}^n H_i^T z_{i,1}(\theta)\right);$$

for $\theta \in \mathbb{R}$ and $H_i = 1$ we notice that $\theta^* = \frac{1}{n}\sum_{i=1}^n z_i$. Consensus algorithms have consequently been used to solve this problem distributively [3]. Considered here is a more generalized family of $z_i(\theta)$ that are time varying. It is assumed that an estimate of θ is required in real-time, for example for an active tracking or response mission. The problem setup is consequently suitable for the proposed online optimization algorithm. The bounds presented in Theorem 2 follow after selecting $\psi(x) = \frac{1}{2}\|x\|_2^2$ and $\alpha(t)$ accordingly. To find the constants R and L featured in the result, we note that for $x \in \chi$ $\psi(x) \leq \frac{1}{2}\theta_{\max}^2$, and $R \leq \frac{1}{\sqrt{2}}\theta_{\max}$. Further, the function $f_{i,k}$ is Lipschitz as it is convex on a compact domain. Hence, from equation (9) as $R_T(x^*, x_i)/T \to 0$, "on average" the algorithm performs as well as best fixed case θ^*. For the case where $\theta_t = \theta_{t+1}$ for all $t = 1, 2, \ldots, T$, the optimal value is θ^*.

[7]Note that the interconnection topology between the network sensors is defined via a graph $\mathcal{G} = (V, E, W)$ where V corresponds to the n sensors, E the communication links between sensors and W the corresponding weights on the links.

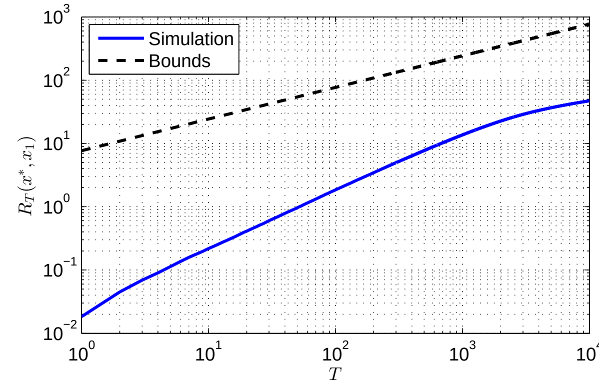

Figure 2. Accuracy of bounds in (9) where $\mathcal{G}$ is a 100 node random unweighted undirected graph with probability $p = 0.08$ and $\sigma_2(P(\mathcal{G})) = 0.0429$.

The ODD algorithm was applied on the described distributed sensor setup (without dual averaging) for 100 sensors. The objective is to estimate a scalar $\theta \in \left(-\frac{1}{2}, \frac{1}{2}\right)$ with a fixed $H_i \in \left(0, \frac{1}{4}\right)$ for each agent and $\sup_i |H_i| = \frac{1}{4}$. The observation for agent i at time t is $z_{t,i}(\theta) = a\theta + b$ for some $a \in (0, 1)$ and $b \in \left(-\frac{1}{4}, \frac{1}{4}\right)$. Thus, $d = 1$, $\chi = \left(-\frac{1}{2}, \frac{1}{2}\right)$, $h_{\max} = \frac{1}{4}$, $\theta_{\max} = \frac{1}{2}$, $R = \frac{1}{2\sqrt{2}}$ and $L = \frac{7}{32}$.

The ODD algorithm was applied to a random unweighted undirected graph sensor network with $k = \frac{1}{4}$ and edge probability $p = 0.08$. Figure 2 shows the good agreement of the estimated regret bound (9) and simulation results indicating that $R_T(x^*, x_1) = O(\sqrt{T})$.

The role of $\sigma_2(P)$, in equation (9), can be applied to the selection of effective sensor network topologies, as the algorithm's performance correlates inversely to the connectivity measure $\sigma_2(P(\mathcal{G}))$. Suitable candidates include those with $\sigma_2(P(\mathcal{G}))$ that scale well with n, such as random regular graphs and expander graphs [21].

VI. Conclusion

This paper studies the problem of decentralized decision making in a networked system. A fast online algorithm was presented in which the decisions are made locally and parallel in time. Our analysis provides a sublinear regret bound of $O(\sqrt{T})$. In particular, the online algorithm performance improves over time with respect to the best fixed decision performance in hindsight. In addition, extending the dual averaging method to an online setting the role of the underlying network topology in achieving good regret has been highlighted. Specifically, the regret bound improves with increasing connectivity of network.

The proposed algorithm was applied on a distributed sensor estimation problem. The suitability of this problem to an online approach was justified by the requirement of acquiring estimates in real-time coupled with sensors susceptibility to unknown errors. The simulation results showcased the benefits of the online learning scheme providing an estimation that on the average performed as well as the best case fixed solution in hindsight.

Future work of particular interest involves generalizing the results to time varying network topology, investigating favorable graph characteristics for the online framework improving the regret bound, and examining traditional problems in robotics and control systems using distributed online optimization.

VII. Appendix

The following two Lemma 4 and 5 can be found in [7], and so are presented here without proof.

Lemma 4. *For any $u, v \in \mathbb{R}^m$, and under the conditions stated for proximal function ψ and step size $\alpha(t)$, we have* $\|\Pi_{\chi}^{\psi}(u,\alpha) - \Pi_{\chi}^{\psi}(v,\alpha)\| \leq \alpha\|u-v\|_*$.

Lemma 5. *For any positive and non-increasing sequence $\alpha(t)$ and $x^* \in \chi$*
$\sum_{t=1}^{T} < \bar{g(t)}, x(t) - x^\star(t) > \leq \frac{1}{2}\sum_{t=1}^{T}\alpha(t-1)\|\bar{g(t)}\|_\star^2 + \frac{1}{\alpha(T)}\psi(x^\star)$,
where the sequence of $y(t)$ is generated by (8).

The following result presents a bound on $\|\bar{z}(t) - z_i(t)\|_*$ proportional to the error incurred in decentralized update of Algorithm 1.

Lemma 6. *For any sequences of $z_i(t)$ and $\bar{z}(t)$ generated by line 5 of Algorithm 1 and 7, respectively, we have* $\|\bar{z}(t) - z_i(t)\|_* \leq \frac{\sqrt{n}L}{1-\sigma_2(P)} + 2L$ *for all $i \in [n]$ and $t \in [T]$.*

Proof: If we write the update in line 5 after s steps for all $i \in [n]$ we have

$$z_i(t) = \sum_{j=1}^{n} P_{ji}^{s} z_j(t-s) + \sum_{k=t-s}^{t-2}\sum_{j=1}^{n} P_{ji}^{t-k-1} g_j(k) + g_i(t-1). \tag{22}$$

Since $\bar{z}(t)$ evolves as in (7), $\bar{z}(t) = \bar{z}(t-s) + \sum_{k=t-s}^{t-1}\bar{g}(k)$. By setting $s = t$ in (22) and assuming $z_i(0) = 0$, we get

$$\bar{z}(t) - z_i(t) = \sum_{k=0}^{t-2}\left(\sum_{j=1}^{n}\left(\frac{1}{n} - P_{ji}^{t-k-1}\right) g_j(k)\right) + \bar{g}(t-1) - g_i(t-1). \tag{23}$$

Since $\|g_i(t)\|_\star \leq L$, the dual norm of (23) is bounded as

$$\|\bar{z}(t) - z_i(t)\|_* \leq L\sum_{k=0}^{t-2}\|P^{t-k-1}e_i - \frac{\mathbf{1}}{n}\|_1 + 2L, \tag{24}$$

where e_i denotes the ith basis vector in $\mathbb{R}^n$ and $\mathbf{1} \in \mathbb{R}^n$ is an all ones vector. Therefore, (24) is further bounded by[8]

$$\|\bar{z}(t) - z_i(t)\|_* \leq \sqrt{n}L\sum_{k=1}^{t-1}\sigma_2(P)^k + 2L, \leq \frac{\sqrt{n}L}{1-\sigma_2(P)} + 2L. \tag{25}$$

■

A different bound on $\|\bar{z}(t) - z_i(t)\|_*$ is presented in [7].

[8]Note that a property of stochastic matrix P followed from Perron-Frobeniuos theory is introduced by Duchi *et al.* [7] as $\|P^t x - \frac{\mathbf{1}}{n}\|_1 \leq \sigma_2(P)^t\sqrt{n}$, where the vector x belongs to $\{x \in \mathbb{R}^n | x \geq 0, \sum_{i=1}^{n} x_i = 1\}$ and $\sigma_2(P)$ is the second largest singular value of P. In addition, as P is doubly stochastic then $\sigma_2(P) \leq 1$ [22].

References

[1] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[2] N. Ganguly, A. Deutsch, and A. Mukherjee, *Dynamics on and of complex networks: applications to biology, computer science, and the social sciences*. Boston: Birkhauser, 2009.

[3] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. NJ: Princeton University Press, 2010.

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

[5] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control,*, pp. 1291–1306, 2011.

[6] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, pp. 221–229, 2013.

[7] 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.

[8] S. Ram, A. Nedic, and V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM Journal on Optimization*, pp. 1–30, 2009.

[9] S. Sundhar Ram, A. Nedić, 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, 2010.

[10] A. Agarwal and J. Duchi, "Distributed delayed stochastic optimization," in *in Proc. 2012 IEEE 51st Annual Conference on Decision and Control*, 2012, pp. 5451–5452.

[11] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. International Conference on Maching Learning*, 2003, pp. 421–422.

[12] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, pp. 169–192, 2007.

[13] L. Xiao, "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization," *Journal of Machine Learning Research*, vol. 11, pp. 2543–2596, 2010.

[14] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, p. 1, 2012.

[15] M. Raginsky, N. Kiarashi, and R. Willett, "Decentralized Online Convex Programming with Local Information," in *Proc. American Control Conference*, 2011, pp. 5363–5369.

[16] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, pp. 107–194, 2012.

[17] S. Bubeck, "Introduction to Online Optimization," *Lecture Notes*, 2011.

[18] E. Hazan, "The Convex Optimization Approach to Regret Minimization," *Optimization for machine learning*, pp. 287–294, 2011.

[19] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, vol. 120, pp. 221–259, 2007.

[20] A. Chapman and M. Mesbahi, "Stability analysis of nonlinear networks via M-matrix theory: Beyond linear consensus," in *Proc. American Control Conference*, 2012, pp. 6626–6631.

[21] B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.

[22] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.