

Distributed online convex optimization on time-varying directed graphs

Mohammad Akbari Bahman Ghahesifard Tamás Linder

Abstract—This paper introduces a class of discrete-time distributed online optimization algorithms, with a group of agents whose communication topology is given by a uniformly strongly connected sequence of time-varying networks. At each time, a private locally Lipschitz strongly convex objective function is revealed to each agent. In the next time step, each agent updates its state using its own objective function and the information gathered from its immediate in-neighbors at that time. Under the assumption that the sequence of communication topologies is uniformly strongly connected, we design an algorithm, distributed over the sequence of time-varying topologies, which guarantees that the individual regret, the difference between the network cost incurred by the agent’s states estimation and the cost incurred by the best fixed choice, grows only sublinearly. This algorithm consists of a subgradient flow along with a push-sum step to adjust for the directed nature of the network topologies. We implement the proposed algorithm on a sensor network and the results show the proper performance of the algorithm.

I. INTRODUCTION

Many scenarios concerning the coordination of multi-agent systems can be modeled as optimization problems in which individual agents cooperatively try to minimize a common cost function. The main feature of any implementable coordination protocol is that the agents only use the information from their neighbouring agents, where the neighbourhood structure is cast as a graph, often directed and time-varying, to update their states. One well-studied class of such optimization problems is the so-called consensus-based distributed optimization problem, where the objective is for the aggregate states of the agents to converge to the set of minimizers of the common cost function [2], [3], [4]. The problem has a variety of applications including localization and robust estimation [5], formation control [6], and energy dispatch in power distribution networks [7], and has been extensively studied in recent years [2], [3], [8], [9], [10], [4].

Many practical scenarios of distributed optimization, however, are in highly dynamic environments, e.g., scheduling of renewable energy systems, where uncertainty plays a central role, and estimation using sensor networks, where the observations of each sensor change with time due to noise. Some of these issues can be addressed within the framework of online optimization, where the functions allocated to each agent possibly change with time, and this change is seen by the agents only in hindsight. In this sense, this is inherently different from distributed time-varying optimization, or dynamic consensus [11]. The objective is to bound the so-called

regret function, which measures the difference between the accumulated collective cost and the cost obtained by the best fixed decision, made by a hypothetical decision maker that knows the objective functions in advance.

There is a vast literature on online optimization, all of which we are unable to review here. This work builds on gradient-descent methods that have been used extensively for online convex optimization; see [12], [13] and [14] for a recent survey. In particular, it is well-known that gradient-descent protocols achieve regret bound of $\mathcal{O}(\sqrt{T})$ on convex functions, and $\mathcal{O}(\ln(T))$ on twice differentiable strongly convex functions, where T denotes the time horizon, see for example [13]. With the interest in decentralized architectures and motivated by the problem of distributed convex optimization, a distributed version of online optimization is proposed in [15], [16]. In [17] and [18], [19] consensus-based gradient-descent algorithms for distributed online optimization are proposed. In this setting, each agent aims at driving its individual average regret, which is the average over time of the regret function evaluated at this agent’s estimation for the choice that the whole network should make, to zero. Given that the agents do not have access to the local cost functions of other agents, these individual regrets are not computable. Nevertheless, the agents can use a consensus-based gradient-descent protocol to collaboratively achieve their objectives. A consensus-based dual averaging discrete-time protocol for online optimization on undirected networks is proposed in [17], and is extended in [20] to accommodate for time-varying weights, but on a fixed directed graph. In [18], [19], motivated by the saddle-point dynamics in [21], a discrete-time distributed online convex optimization algorithm on weight-balanced network topologies is introduced; in particular, the suggested protocol in [19] works on jointly connected weight-balanced digraphs. Other recent work includes [22], where under the assumption of doubly stochasticity, a gossip-based protocol is developed for distributed online convex optimization. In contrast, we develop an algorithm that achieves a sublinear regret over any sequence of uniformly strongly connected time-varying directed graphs. The idea behind our protocol is the push-sum algorithm, which was originally used for consensus [23], [24] on directed graphs with imbalanced nodes. In particular, some of our main results rely on an extension of this class of algorithms to the so-called perturbed push-sum protocol, which works on any uniformly strongly connected digraph and has recently been used for distributed convex optimization [25], [26]. In contrast, here we are interested in distributed *online* optimization.

Statement of contributions: The contributions of this paper are the following. We consider a group of agents communicating over a sequence of time-varying directed graphs. At each time instance, each agent uses the information about the

The authors are with the Department of Mathematics and Statistics at Queen’s University, Kingston, ON, Canada. {13mav1,gharesb,linder}@queensu.ca. Parts of this work were presented at the 2014 Allerton Conference on Communication, Control, and Computing. [1]

states of its neighboring agents and makes a decision about its next state. After that, the agent receives a locally Lipschitz strongly convex cost function and incurs a cost for its state estimation. Following the framework of [17] and [19], the regret for each individual agent at each time is defined as the difference between the network cost incurred by the agent's state estimation and the cost incurred by the best fixed choice, made by a decision maker that has access to the objective functions. Assuming that the individual cost functions are strongly convex on a compact neighbourhood of their minimizers and have bounded subgradients, we design a distributed discrete-time algorithm which achieves sublinear regret, logarithmic up to a square, i.e., $\mathcal{O}((\ln(T))^2)$, on any sequence of time-varying uniformly strongly connected digraphs. In this sense, and in contrast to the known consensus-based gradient-descent protocols for distributed online optimization, our proposed strategy does not rely on having weight-balanced or doubly stochastic network topologies, and accommodates time-varying directed graphs. The proposed algorithm can be thought of as an extension of the subgradient push-sum strategy, recently used for distributed convex optimization in [25], to online settings. Our proof strategy is to provide a sublinear network regret and then a sublinear bound on the difference between network and agent regret. For the special class of Ramanujan graphs, we make the dependency of our upper bound for the regret on the number of agents explicit and show that for a sufficiently large time, this upper bound grows linearly with the size of the network. Finally, we discuss an application of the proposed algorithm to a sensor network estimation problem, where a group of sensors with independent observations cooperatively and by communicating over a time-varying graph estimate a target.

Organization: Section II contains mathematical preliminaries on linear algebra, convex analysis, and graph theory. Section III introduces the distributed online convex optimization problem under study. In Section IV, we propose our distributed online discrete-time convex optimization algorithm which achieves sublinear regret. Section V contains our main contribution. We demonstrate the results by a simulation on a sensor network in Section VI. Section VII gathers our conclusions and ideas for future work.

II. MATHEMATICAL PRELIMINARIES

We start with some notational conventions that we use throughout the paper. Let \mathbb{R} , $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{> 0}$, \mathbb{Z} , $\mathbb{Z}_{> 0}$ denote the set of real, nonnegative real, positive real, integer, and positive integer numbers, respectively. We denote by $\|\cdot\|_2$ and $\|\cdot\|_1$ the Euclidean norm and 1-norm on \mathbb{R}^d , $d \in \mathbb{Z}_{> 0}$, respectively, and also denote by $\bar{B}(x, r) = \{y \in \mathbb{R}^d : \|y - x\|_2 \leq r\}$, the closed ball of radius r centered at $x \in \mathbb{R}^d$. We use the short-hand notation $\mathbf{1}_d = (1, \dots, 1)^\top \in \mathbb{R}^d$. We let I_d denote the identity matrix in $\mathbb{R}^{d \times d}$. For matrices $A \in \mathbb{R}^{d_1 \times d_2}$ and $B \in \mathbb{R}^{e_1 \times e_2}$, $d_1, d_2, e_1, e_2 \in \mathbb{Z}_{> 0}$, we let $A \otimes B$ denote their Kronecker product. We say matrix A is column stochastic (resp. row stochastic) if $\mathbf{1}_{d_1}^\top A = \mathbf{1}_{d_2}^\top$ (resp. $A \mathbf{1}_{d_2} = \mathbf{1}_{d_1}$). We also let $\sigma_i(A)$ denote the i th largest singular value of matrix A .

Convex analysis: A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex if for all $x, y \in \mathbb{R}^d$ and for all $\lambda \in [0, 1]$, we have

$$\lambda f(x) + (1 - \lambda)f(y) \geq f(\lambda x + (1 - \lambda)y).$$

Given a convex function f and $x \in \mathbb{R}^d$, we call $g_x \in \mathbb{R}^d$ a subgradient of f at x , if

$$f(y) - f(x) \geq g_x^\top (y - x),$$

for all $y \in \mathbb{R}^d$. It is well-known that the set of subgradients of a convex function is a nonempty, convex, compact for all $x \in \mathbb{R}^d$, see [27, Proposition 4.2.1]. We denote by $\partial f(x)$ the set of subgradients of f at x . We say $\partial f(x)$ is L -bounded if there exists $L \in \mathbb{R}_{\geq 0}$ such that $\|g_x\|_1 \leq L$ for all $g_x \in \partial f(x)$ and $x \in \mathbb{R}^d$. The function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is called Lipschitz, if for all $x, y \in \mathbb{R}^d$, $|f(x) - f(y)| \leq C\|x - y\|_2$ for some $C \in \mathbb{R}_{\geq 0}$. Note that a function with L -bounded subgradients is Lipschitz. The function f is μ -strongly convex, for some $\mu \in \mathbb{R}_{> 0}$, if for each $x \in \mathbb{R}^d$ and $g_x \in \partial f(x)$, we have

$$f(y) - f(x) \geq g_x^\top (y - x) + \frac{\mu}{2}\|y - x\|_2^2,$$

for all $y \in \mathbb{R}^d$. We let $\operatorname{argmin}(f)$ denote the set of minimizers of a convex function f in its domain. The convex function f is locally strongly convex if it is strongly convex on a compact set containing $\operatorname{argmin}(f)$. For $\beta \in [0, 1]$, a convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with $\operatorname{argmin}(f) \neq \emptyset$ is β -central on $Z \subset \mathbb{R}^n \setminus \operatorname{argmin}(f)$ if for each $x \in Z$, there exists $y \in \operatorname{argmin}(f)$ such that

$$-g_x^\top (y - x) \geq \beta \|g_x\|_2 \|y - x\|_2,$$

for all $g_x \in \partial f(x)$.

Graph theory: A weighted directed graph (or digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ consists of a vertex set \mathcal{V} , an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and an adjacency matrix $\mathbf{A} \in \mathbb{R}_{\geq 0}^{n \times n}$ with $a_{ij} > 0$ iff $(v_i, v_j) \in \mathcal{E}$. We assume each agent has a self-loop, so $a_{ii} > 0$ for all $i \in \mathcal{V}$. A path is a sequence of distinct vertices connected by edges. The graph \mathcal{G} is strongly connected if there is a path between any pair of distinct vertices. We define in-neighbors and out-neighbors of node v_i , respectively, as $N_i^{\text{in}} = \{v_j | (v_j, v_i) \in \mathcal{E}\} \cup \{v_i\}$ and $N_i^{\text{out}} = \{v_j | (v_i, v_j) \in \mathcal{E}\} \cup \{v_i\}$. The in- and out-degree of v_i are, respectively, $d_i^{\text{in}} = |N_i^{\text{in}}|$ and $d_i^{\text{out}} = |N_i^{\text{out}}|$. A regular (undirected) graph is a graph where every vertex has the same number of neighbours. A regular graph with vertices of degree d is a d -regular graph. A Ramanujan graph is a d -regular graph satisfying $\sigma_2(\mathbf{A}) \leq 2\sqrt{d-1}$, where $\mathbf{A} = [a_{ij}]$ is the unweighted adjacency matrix of the graph, i.e., $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise, see [28].

III. PROBLEM STATEMENT

We begin with describing the problem of online convex optimization. Suppose we have a sequence of convex cost functions $\{f^1, f^2, \dots, f^T\}$, where $f^t : \mathbb{R}^d \rightarrow \mathbb{R}$ for each $t \in \{1, \dots, T\}$ ($T \in \mathbb{Z}_{> 0}$ is the time horizon). At each time step $t \in \{1, \dots, T\}$, a decision maker chooses an action $z(t) \in \mathbb{R}^d$ and after committing to this decision, a convex cost function $f^t : \mathbb{R}^d \rightarrow \mathbb{R}$ is revealed and the decision maker is faced with a loss of $f^t(z(t))$. In this scenario, due to lack of access to the cost functions before the decision is made, the decision does not necessarily correspond to the minimizers and

the decision maker faces a so-called *regret*. Regret is defined as the difference between the accumulated cost over time and the cost incurred by the best fixed decision, when all the functions are known in advance, see [12], [13]. Formally, the regret is

$$R(T) = \sum_{t=1}^T f^t(z(t)) - \sum_{t=1}^T f^t(z^*),$$

where

$$z^* \in \operatorname{argmin}_{z \in \mathbb{R}^d} \sum_{t=1}^T f^t(z).$$

Throughout the paper, we assume that the minimizer set is nonempty. The objective here is to design a strategy for the decision maker so that it achieves a regret that is sublinear in T , i.e., $\limsup_{T \rightarrow \infty} \frac{R(T)}{T} = 0$, which guarantees that the average regret over time goes to zero.

Let us now review the setup for a distributed version of the online optimization problem [17], [19]. Consider a group of agents communicating with each other over a time-varying network, modeled by a directed graph at each time step, with properties that will be described shortly. At each time step $t \in \{1, 2, \dots, T\}$, an agent $i \in \mathcal{V} = \{1, \dots, n\}$ chooses its state $z_i(t) \in \mathbb{R}^d$. After this, a locally strongly convex cost function $f_i^t : \mathbb{R}^d \rightarrow \mathbb{R}$ is revealed, and the agent incurs the cost $f_i^t(z_i(t))$; in fact, agent i will not necessarily see the whole function, but can see its value and compute its subgradient at $z_i(t)$. In this scenario, at each time t , the whole network aims to minimize the cost function

$$f^t(z) = \sum_{i=1}^n f_i^t(z),$$

which is distributed among agents and is revealed when agents have chosen their states. Therefore, each agent guesses its state based on what it thinks the whole network would choose.

The regret of agent $j \in \mathcal{V}$, see [17], [18], is now defined as

$$R^j(T) := \sum_{t=1}^T \sum_{i=1}^n f_i^t(z_j(t)) - \sum_{t=1}^T \sum_{i=1}^n f_i^t(z^*); \quad (1)$$

where

$$z^* \in \operatorname{argmin}_{z \in \mathbb{R}^d} \sum_{t=1}^T \sum_{i=1}^n f_i^t(z). \quad (2)$$

Note that, since f^t is locally strongly convex, z^* is unique. This *individual* regret function for agent j computes the difference between the network cost incurred by the agent's states estimation and the cost incurred by the best fixed choice, when all functions are known in advance.

It is essential to note that, at each time, each agent has only access to the value of its own (past) cost functions, and their subgradients, and has only partial information about the other agents' states. Therefore, agents cannot compute their own regret. However, at each time step, agents have access to a communication network over which they can share information. In particular, at time $t \in \{1, 2, \dots, T\}$, agent $i \in \mathcal{V}$ receives information about the states of its in-neighbour via a time-varying directed graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), A(t))$. We assume that the sequence $\mathcal{G}(t)$, $t \in \{1, \dots, T\}$ is uniformly strongly connected (or B-strongly-connected), which means it

is strongly connected in a period of time, see [25]; specifically, there exists $B \in \mathbb{Z}_{>0}$ such that for each $k \in \mathbb{Z}_{>0}$, the digraph with vertices \mathcal{V} and edge set $\mathcal{E}_B(k) = \bigcup_{t=kB+1}^{(k+1)B} \mathcal{E}(t)$ is strongly connected. Finally, similar to most distributed optimization algorithms, we assume throughout that every node knows its out-degree.

The main objective of this paper is to design a distributed algorithm over the prescribed time-varying network topology which allows the agents to asymptotically drive the *average* individual regret over time to zero, even though limited information is available to the agents. More specifically, the distributed algorithm must have the property that the individual regret is upper bounded sublinearly of time T .

IV. DISTRIBUTED ONLINE SUBGRADIENT PUSH-SUM ALGORITHM

In this section, we introduce a distributed online subgradient push-sum algorithm motivated by [25], [23], which allows the agents to have a sublinear average individual regret. To this end, let us consider a group of agents $\mathcal{V} = \{1, \dots, n\}$ with the communication topology prescribed by a sequence of B-strongly-connected time-varying digraph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), A(t))$ as before. The distributed online subgradient push-sum algorithm is a discrete-time dynamical system, which is described next. We assume that at each time $t \in \{1, \dots, T\}$, each agent has four states: $x_i(t) \in \mathbb{R}^d$, $y_i(t) \in \mathbb{R}$, $w_i(t) \in \mathbb{R}^d$ and $z_i(t) \in \mathbb{R}^d$, which the agent computes locally. Here, $z_i(t)$ is the agent's primary state which incurs the cost $f_i^t(z_i(t))$. The parameters $x_i(t)$ and $w_i(t)$ are used to estimate $z_i(t)$ by using other agents' states and properties of cost function f_i^t . Finally, $y_i(t)$ is a scalar used to determine the influence of the agent's neighbours on its states over a directed graph.

We are now in a position to introduce our *distributed online subgradient push-sum algorithm*. At each iteration $t \in \{1, \dots, T\}$, the agent $i \in \mathcal{V}$ computes its next time state values by

$$w_i(t+1) = \sum_{j \in N_i^{\text{in}}(t)} \frac{x_j(t)}{d_j^{\text{out}}(t)},$$

$$y_i(t+1) = \sum_{j \in N_i^{\text{in}}(t)} \frac{y_j(t)}{d_j^{\text{out}}(t)},$$

$$z_i(t+1) = \frac{w_i(t+1)}{y_i(t+1)},$$

$$x_i(t+1) = w_i(t+1) - \alpha(t+1)g_i^{t+1}(z_i(t+1)),$$

where $g_i^{t+1}(z_i(t+1))$ is the subgradient of the function f_i^{t+1} at $z_i(t+1)$ and $\alpha : \mathbb{Z}_{>0} \rightarrow \mathbb{R}_{>0}$ is the *learning rate*. Throughout the rest of this paper, for simplicity, we write $g_i(t+1)$ instead of $g_i^{t+1}(z_i(t+1))$. We set the initial value $y_i(0) = 1$ for all $i \in \mathcal{V}$, and $x_i(0) \in \mathbb{R}^d$ where $i \in \mathcal{V}$. Note that f_i^t is available only after agent i selects the state $z_i(t)$.

We now briefly describe how each agent computes its state values. At each time t , all in-neighbor agents $j \in N_i^{\text{in}}(t)$ of agent i share $\frac{x_j(t)}{d_j^{\text{out}}(t)}$ and $\frac{y_j(t)}{d_j^{\text{out}}(t)}$ with this agent; hence i can compute $w_i(t+1)$, $y_i(t+1)$, $z_i(t+1)$, $x_i(t+1)$ using this information.

It is useful to represent the discrete-time dynamical system described above in matrix form. To this end, let us define the matrix $A(t) = [a_{ij}(t)]_{n \times n}$ and $\mathbf{A}(t) = A(t) \otimes \mathbf{I}_d$, where

$$a_{ij}(t) = \begin{cases} 1/d_j^{\text{out}}(t) & \text{whenever } j \in N_i^{\text{in}}(t) \\ 0 & \text{otherwise.} \end{cases}$$

The algorithm described above can now be written as

$$\begin{aligned} w(t+1) &= \mathbf{A}(t)x(t), \\ y(t+1) &= A(t)y(t), \\ z_i(t+1) &= \frac{w_i(t+1)}{y_i(t+1)}, \quad \text{for all } i \in \mathcal{V}, \\ x(t+1) &= w(t+1) - \alpha(t+1)g(t+1), \end{aligned} \quad (3)$$

where $w(t) = (w_1^\top(t), \dots, w_n^\top(t))^\top$, $x(t) = (x_1^\top(t), \dots, x_n^\top(t))^\top$, $y(t) = (y_1(t), \dots, y_n(t))^\top$, and $g(t) = (g_1^\top(t), \dots, g_n^\top(t))^\top$.

V. MAIN RESULT

In this section, we show how the distributed online subgradient push-sum algorithm (3) can be used to bound the individual regret defined in (1). Before stating our main result, we specify the sequence of cost functions $\{f_1^t, f_2^t, \dots, f_n^t\}_{t=1}^T$ that we consider throughout this paper.

Assumption 5.1: $\{f_1^t, f_2^t, \dots, f_n^t\}_{t=1}^T$ is a sequence of convex functions with nonempty set of minimizers, where for each $i \in \{1, \dots, n\}$, the function f_i^t :

- (i) has L_i -bounded subgradients, where $L_i \in \mathbb{R}_{>0}$, and
- (ii) is μ -strongly convex on $\bar{B}(0, H(\frac{\mu K_1}{2L}))$ for some $\mu \in \mathbb{R}_{>0}$, $K_1 \in \mathbb{R}_{>0}$ independent of T , and $L = \sum_{i=1}^n L_i$, where $H(\cdot)$ is defined in equation (19) in the Appendix, and $\cup_{i=1}^n \cup_{t=1}^T \arg \min f_i^t \subset \bar{B}(0, K_1/2)$.

The following theorem is the main result of this paper.

Theorem 5.2: (Sublinear agent's regret bound): Consider a group of agents $\mathcal{V} = \{1, \dots, n\}$ over a sequence of B-strongly connected graphs, where $T, n \in \mathbb{Z}_{>0}$. Let $\{f_1^t, f_2^t, \dots, f_n^t\}_{t=1}^T$ be a sequence of convex cost functions that satisfies Assumption 5.1. Suppose that the learning rate is given by $\alpha(t) = \frac{1}{\mu t}$ and that the agents use (3) to generate the sequence $\{z(t) = (z_1(t), z_2(t), \dots, z_n(t))\}_{t=1}^T$. Then for each agent $j \in \mathcal{V}$, we have

$$\mathbf{R}^j(T) \leq C_1 + C_2(1 + \ln(T)) + C_3(1 + \ln(T))^2, \quad (4)$$

where

$$\begin{aligned} C_1 &= \frac{8L}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ &\quad + \frac{8\mu n}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 \|\bar{x}(0) - z^*\|_2 \\ &\quad + \frac{16nL_j}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1, \\ C_2 &= \frac{8L^2}{\mu\delta(1-\lambda)} + \frac{8L}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 + \frac{16nL_j}{\delta(1-\lambda)} \frac{L}{\mu} \\ &\quad + \frac{8n}{\delta(1-\lambda)} L \|\bar{x}(0) - z^*\|_2 + \frac{L^2}{2n\mu}, \\ C_3 &= \frac{8}{\delta(1-\lambda)} \frac{L^2}{\mu}, \end{aligned}$$

z^* is defined in (2), $\delta \in \mathbb{R}_{>0}$ and $\lambda \in \mathbb{R}_{>0}$ depend on the network topology, c.f. Lemma 5.5,

$$\begin{aligned} L &= \sum_{i=1}^n L_i, \quad \text{and} \\ \bar{x}(t) &= \frac{1}{n} (x_1(t) + x_2(t) + \dots + x_n(t)) \\ &= \frac{1}{n} (\mathbf{1}_n \otimes \mathbf{I}_d)^\top [x_1^\top(t), x_2^\top(t), \dots, x_n^\top(t)]^\top. \end{aligned} \quad (5)$$

Before we prove this result, we make a few remarks on the comparison of our results with previous works. The distributed online subgradient push-sum algorithm (3) and the result presented by Theorem 5.2 do not rely on fixed graph topologies, or on the fact that the underlying network is weight-balanced. In this sense, this result is more general than the existing results in the literature [18], [17], [19]. On the other hand, the bound obtained is of order $(\ln(T))^2$, rather than $\ln(T)$, which is slightly worse than the known regret bounds in the centralized scenarios, or the known cases on weight-balanced directed graphs. This may be due to the estimates that we have used for some of our upper bounds, or can be due to the nature of the distributed online subgradient push-sum algorithm.

A. Proof of the main result

In order to prove this result, we need to define a so-called *network regret*. The network regret of agents (for the sequence of functions $\{f_i^t\}_{t=1}^T$) using $\{z_i(t)\}_{t=1}^T, i \in \mathcal{V}$ with respect to a best fixed offline decision $z^* \in \mathbb{R}^d$ is defined as

$$\mathbf{R}(T) := \sum_{t=1}^T \sum_{i=1}^n f_i^t(z_i(t)) - \sum_{t=1}^T \sum_{i=1}^n f_i^t(z^*), \quad (6)$$

where

$$z^* \in \underset{z \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{t=1}^T \sum_{i=1}^n f_i^t(z).$$

This regret function captures the difference between the collective accumulated cost over time and the cost resulting from the best offline fixed choice, selected by assuming that the information about the cost functions is available in advance.

Theorem 5.3: (Sublinear network regret bound): Consider a group of agents $\mathcal{V} = \{1, \dots, n\}$ over a sequence

of B-strongly connected graphs, where $T, n \in \mathbb{Z}_{>0}$. Let $\{f_1^t, f_2^t, \dots, f_n^t\}_{t=1}^T$ be a sequence of convex cost functions that satisfies Assumption 5.1. Then the sequence $\{z(t) = (z_1(t), z_2(t), \dots, z_n(t))\}_{t=1}^T$ generated by (3) with the learning rate $\alpha(t) = \frac{1}{\mu t}$ satisfies the network regret bound

$$R(T) \leq \tilde{C}_1 + \tilde{C}_2(1 + \ln(T)) + \tilde{C}_3(1 + \ln(T))^2,$$

where

$$\begin{aligned} \tilde{C}_1 &= \frac{8L}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ &\quad + \frac{8\mu n}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 \|\bar{x}(0) - z^*\|_2, \\ \tilde{C}_2 &= \frac{8L^2}{\mu\delta(1-\lambda)} + \frac{8L}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 \\ &\quad + \frac{8nL}{\delta(1-\lambda)} \|\bar{x}(0) - z^*\|_2 + \frac{L^2}{2n\mu}, \\ \tilde{C}_3 &= \frac{8}{\delta(1-\lambda)} \frac{L^2}{\mu}, \end{aligned}$$

z^* is defined in (2), $\delta \in \mathbb{R}_{>0}$ and $\lambda \in \mathbb{R}_{>0}$ depend on the network topology, and L and $\bar{x}(0)$ are given by (5).

The proof relies on a sequence of results, which we present next. Throughout the rest of this section, we adopt the notation introduced in Theorem 5.2.

Lemma 5.4: Let $\{f_1^t, f_2^t, \dots, f_n^t\}_{t=1}^T$ be a sequence of convex cost functions that satisfies Assumption 5.1. Then the sequence $\{z(t)\}_{t=1}^T$ generated by (3) with the learning rate $\alpha(t)$, over a sequence of B-strongly connected graphs, satisfies

$$\begin{aligned} R(T) &\leq \sum_{t=1}^T \sum_{i=1}^n L_i \|z_i(t) - \bar{x}(t-1)\|_2 + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ &\quad + \frac{n}{2} \sum_{t=1}^{T-1} \|\bar{x}(t) - z^*\|_2^2 \left(\frac{1}{\alpha(t+1)} - \frac{1}{\alpha(t)} - \mu \right) \\ &\quad - \sum_{t=1}^T \sum_{i=1}^n \mu (z_i(t) - \bar{x}(t-1))^\top (\bar{x}(t-1) - z^*) \\ &\quad + \frac{L^2}{2n} \sum_{t=1}^T \alpha(t), \end{aligned}$$

where L and $\bar{x}(t)$ are given by (5).

Proof: Using Theorem A.2 of the Appendix and Assumption 5.1, for any given initial condition and any agent $i \in \mathcal{V}$, we have that $z_i(t)$ stays in $\bar{B}(0, H(\frac{\mu K_1}{2L}))$ for all $t \in \{1, 2, \dots, T\}$, where the modulus of strong convexity of f is μ . Using (6) and since $\{\{f_i^t\}_{i=1}^n\}_{t=1}^T$ is a sequence of μ -strongly convex functions, we have that

$$\begin{aligned} R(T) &= \sum_{t=1}^T \sum_{i=1}^n (f_i^t(z_i(t)) - f_i^t(z^*)) \\ &\leq \sum_{t=1}^T \sum_{i=1}^n (g_i(t)^\top (z_i(t) - z^*) - \frac{\mu}{2} \|z_i(t) - z^*\|_2^2), \end{aligned}$$

By adding and subtracting $\bar{x}(t-1)$, we obtain

$$\begin{aligned} R(T) &\leq \sum_{t=1}^T \sum_{i=1}^n (g_i(t)^\top (z_i(t) - \bar{x}(t-1) + \bar{x}(t-1) - z^*) \\ &\quad - \frac{\mu}{2} \|z_i(t) - \bar{x}(t-1) + \bar{x}(t-1) - z^*\|_2^2), \\ &= \sum_{t=1}^T \sum_{i=1}^n \left(g_i(t)^\top (z_i(t) - \bar{x}(t-1)) \right. \\ &\quad + g_i(t)^\top (\bar{x}(t-1) - z^*) \\ &\quad \left. - \frac{\mu}{2} (\|z_i(t) - \bar{x}(t-1)\|_2^2 + \|\bar{x}(t-1) - z^*\|_2^2 \right. \\ &\quad \left. + 2(z_i(t) - \bar{x}(t-1))^\top (\bar{x}(t-1) - z^*)) \right) \end{aligned} \quad (7)$$

Using (3), we have

$$x(t) = \mathbf{A}(t-1)x(t-1) - \alpha(t)g(t),$$

for all $t \in \{1, \dots, T\}$. Multiplying the equation by $\frac{1}{n}(\mathbf{1}_n \otimes \mathbf{I}_d)^\top$ and using the fact that $\mathbf{A}(t-1)$ is column stochastic, we obtain

$$\bar{x}(t) = \bar{x}(t-1) - \frac{\alpha(t)}{n} \sum_{i=1}^n g_i(t), \quad (8)$$

where $\bar{x}(t)$ is given by (5). Subtracting z^* and taking the norm square, we get

$$\begin{aligned} \|\bar{x}(t) - z^*\|_2^2 &= \|\bar{x}(t-1) - z^*\|_2^2 + \frac{\alpha^2(t)}{n^2} \left\| \sum_{i=1}^n g_i(t) \right\|_2^2 \\ &\quad - \frac{2\alpha(t)}{n} \left[\sum_{i=1}^n g_i(t) \right]^\top (\bar{x}(t-1) - z^*). \end{aligned}$$

As a result, since $\|g_i(t)\|_2 \leq L_i$, we have

$$\begin{aligned} \left[\sum_{i=1}^n g_i(t) \right]^\top (\bar{x}(t-1) - z^*) &\leq \frac{n}{2\alpha(t)} (\|\bar{x}(t-1) - z^*\|_2^2 \\ &\quad - \|\bar{x}(t) - z^*\|_2^2) + \frac{\alpha(t)}{2n} L^2, \end{aligned}$$

where $L = \sum_{i=1}^n L_i$. Using this, we have

$$\begin{aligned} &\sum_{t=1}^T \left[\sum_{i=1}^n g_i(t) \right]^\top (\bar{x}(t-1) - z^*) \\ &\leq \sum_{t=1}^T \frac{n}{2\alpha(t)} (\|\bar{x}(t-1) - z^*\|_2^2 - \|\bar{x}(t) - z^*\|_2^2) \\ &\quad + \frac{L^2}{2n} \sum_{t=1}^T \alpha(t) \\ &\leq \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ &\quad + \frac{n}{2} \sum_{t=1}^{T-1} \|\bar{x}(t) - z^*\|_2^2 \left(\frac{1}{\alpha(t+1)} - \frac{1}{\alpha(t)} \right) + \frac{L^2}{2n} \sum_{t=1}^T \alpha(t). \end{aligned} \quad (9)$$

The proof then follows immediately using (7) and (9), along with the fact that $g_i(\cdot)$ is L_i -bounded over \mathbb{R}^d . ■

We also recall the following result from [26], without stating its proof.

Lemma 5.5: ([26, Corollary 1]): Consider the sequences $\{z_i(t)\}_{t=1}^T$, for all $i \in \mathcal{V}$, generated by (3) on a sequence of B-strongly-connected digraphs. Then we have

$$\begin{aligned} & \|z_i(t+1) - \bar{x}(t)\|_2 \\ & \leq \frac{8}{\delta} \left(\lambda^t \sum_{i=1}^n \|x_i(0)\|_1 + \sum_{s=1}^t \lambda^{t-s} \sum_{i=1}^n \|\alpha(s)g_i(s)\|_1 \right), \end{aligned} \quad (10)$$

where δ and $\lambda \in \mathbb{R}_{>0}$ satisfy

$$\delta \geq \frac{1}{n^{nB}} \quad \text{and} \quad \lambda \leq (1 - \frac{1}{n^{nB}})^{1/(nB)}.$$

Additionally, if each of the graphs $\mathcal{G}(t)$ is regular, then $\delta = 1$ and

$$\lambda \leq \min \left\{ (1 - \frac{1}{4n^3})^{1/(B)}, \max_{t \in \{1, \dots, T\}} \sigma_2(A(t)) \right\}.$$

The constant δ measures the imbalance of the network and λ is a measure of connectivity, see [25] for more details. We state a corollary of this result, which plays a key role in the proof of our main result.

Corollary 5.6: Under the assumption of Theorem 5.3, where the learning rate is chosen as $\alpha(t) = \frac{1}{\mu t}$, we have

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^n L_i \|z_i(t) - \bar{x}(t-1)\|_2 \\ & \leq \frac{8L}{\delta(1-\lambda)} \left(\sum_{i=1}^n \|x_i(0)\|_1 + \frac{L}{\mu} (1 + \ln(T)) \right). \end{aligned} \quad (11)$$

The proof follows immediately from the fact that

$$\sum_{i=1}^n \|g_i(s)\|_1 \leq L \quad \text{and} \quad \sum_{t=1}^T \alpha(t) \leq \frac{1}{\mu} (1 + \ln(T)).$$

The final stepping stone in the proof of Theorem 5.3 is stated next.

Lemma 5.7: Under the assumption of Theorem 5.3, where the learning rate is chosen as $\alpha(t) = \frac{1}{\mu t}$, we have

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^n -\mu(z_i(t) - \bar{x}(t-1))(\bar{x}(t-1) - z^*) \\ & \leq \mu \frac{8n}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 \|\bar{x}(0) - z^*\|_2 \\ & \quad + \frac{8}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 L (1 + \ln(T)) \\ & \quad + \frac{8n}{\delta} L \|\bar{x}(0) - z^*\|_2 \frac{1 + \ln(T)}{1 - \lambda} \\ & \quad + \frac{8}{\delta(1-\lambda)} \frac{L^2}{\mu} (1 + \ln(T))^2. \end{aligned}$$

Proof: Using the Cauchy-Schwarz inequality, we have

$$\begin{aligned} & - \sum_{t=1}^T \sum_{i=1}^n \mu(z_i(t) - \bar{x}(t-1))^\top (\bar{x}(t-1) - z^*) \\ & \leq \sum_{t=1}^T \sum_{i=1}^n \mu \|z_i(t) - \bar{x}(t-1)\|_2 \|\bar{x}(t-1) - z^*\|_2. \end{aligned}$$

Let $X = \sum_{t=1}^T \sum_{i=1}^n \mu \|z_i(t) - \bar{x}(t-1)\|_2 \|\bar{x}(t-1) - z^*\|_2$. From equation (8), we can write

$$\|\bar{x}(t-1) - z^*\|_2 \leq \|\bar{x}(0) - z^*\|_2 + \left\| \sum_{s=1}^{t-1} \frac{\alpha(s)}{n} \sum_{i=1}^n g_i(s) \right\|_2. \quad (12)$$

Using (10) and (12), we can write

$$\begin{aligned} X & \leq \sum_{t=1}^T \sum_{i=1}^n \mu \frac{8}{\delta} \left(\lambda^{t-1} \sum_{j=1}^n \|x_j(0)\|_1 \right. \\ & \quad \left. + \sum_{s=1}^{t-1} \lambda^{t-1-s} \sum_{j=1}^n \|\alpha(s)g_j(s)\|_1 \right) \\ & \quad \times \left(\|\bar{x}(0) - z^*\|_2 + \left\| \sum_{s=1}^{t-1} \frac{\alpha(s)}{n} \sum_{j=1}^n g_j(s) \right\|_2 \right) \\ & \leq \sum_{t=1}^T \sum_{i=1}^n \mu \frac{8}{\delta} \left(\lambda^{t-1} \sum_{j=1}^n \|x_j(0)\|_1 \|\bar{x}(0) - z^*\|_2 \right. \\ & \quad \left. + \lambda^{t-1} \sum_{j=1}^n \|x_j(0)\|_1 \left(\sum_{s=1}^{t-1} \frac{\alpha(s)}{n} L \right) \right. \\ & \quad \left. + L \|\bar{x}(0) - z^*\|_2 \sum_{s=1}^{t-1} \lambda^{t-1-s} \alpha(s) \right. \\ & \quad \left. + \frac{L^2}{n} \sum_{s=1}^{t-1} \lambda^{t-1-s} \alpha(s) \sum_{s=1}^{t-1} \alpha(s) \right). \end{aligned} \quad (13)$$

In the last inequality we used the subgradient bound. Letting $\alpha(s) = \frac{1}{\mu s}$, we have

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^n \mu \frac{8}{\delta} \lambda^{t-1} \sum_{j=1}^n \|x_j(0)\|_1 \|\bar{x}(0) - z^*\|_2 \\ & = \mu \frac{8n}{\delta} \sum_{j=1}^n \|x_j(0)\|_1 \|\bar{x}(0) - z^*\|_2 \sum_{t=1}^T \lambda^{t-1} \\ & \leq \mu \frac{8n}{\delta(1-\lambda)} \sum_{j=1}^n \|x_j(0)\|_1 \|\bar{x}(0) - z^*\|_2, \end{aligned} \quad (14)$$

where we used the fact that $\sum_{t=1}^T \lambda^{t-1} \leq \frac{1}{1-\lambda}$. We also have that

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^n \mu \frac{8}{\delta n} \sum_{j=1}^n \|x_j(0)\|_1 L \lambda^{t-1} \sum_{s=1}^{t-1} \frac{1}{\mu s} \\ & = \frac{8}{\delta} \sum_{j=1}^n \|x_j(0)\|_1 L \sum_{t=1}^T \lambda^{t-1} \sum_{s=1}^{t-1} \frac{1}{s} \\ & \leq \frac{8}{\delta} \sum_{j=1}^n \|x_j(0)\|_1 L \sum_{t=1}^T \lambda^{t-1} (1 + \ln(t)) \\ & \leq \frac{8}{\delta(1-\lambda)} \sum_{j=1}^n \|x_j(0)\|_1 L (1 + \ln(T)), \end{aligned} \quad (15)$$

where we used the fact that $\sum_{t=1}^T \lambda^{t-1}(1+\ln(t)) \leq \frac{(1+\ln(T))}{1-\lambda}$. Also, we have that

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^n \mu \frac{8}{\delta} L \|\bar{x}(0) - z^*\|_2 \sum_{s=1}^{t-1} \lambda^{t-1-s} \frac{1}{\mu s} \\ &= n \frac{8}{\delta} L \|\bar{x}(0) - z^*\|_2 \sum_{t=1}^T \sum_{s=1}^{t-1} \frac{\lambda^{t-1-s}}{s} \\ &\leq n \frac{8}{\delta} L \|\bar{x}(0) - z^*\|_2 \frac{1+\ln(T)}{1-\lambda}. \end{aligned} \quad (16)$$

Finally,

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^n \mu \frac{8}{\delta} \frac{L^2}{n} \sum_{s=1}^{t-1} \left(\frac{\lambda^{t-1-s}}{\mu s} \right) \sum_{s=1}^{t-1} \frac{1}{\mu s} \\ &\leq \frac{8}{\delta} \frac{L^2}{\mu} \sum_{t=1}^T (1+\ln(t)) \sum_{s=1}^{t-1} \frac{\lambda^{t-1-s}}{s} \\ &\leq \frac{8}{\delta(1-\lambda)} \frac{L^2}{\mu} (1+\ln(T))^2. \end{aligned} \quad (17)$$

In (16) and (17), by rearranging the summation, we have $\sum_{t=1}^T \sum_{s=1}^{t-1} \frac{\lambda^{t-1-s}}{s} \leq \frac{1+\ln(T)}{1-\lambda}$. Using (14)-(17) in (13) then yields the result. \blacksquare

We are now in a position to prove Theorem 5.3.

Proof of Theorem 5.3: Using Lemma 5.4 and the assumption that the learning rate is chosen as $\alpha(t) = \frac{1}{\mu t}$, we have that

$$\begin{aligned} \mathbf{R}(T) &\leq \sum_{t=1}^T \sum_{i=1}^n L_i \|z_i(t) - \bar{x}(t-1)\|_2 + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ &\quad - \sum_{t=1}^T \sum_{i=1}^n \mu (z_i(t) - \bar{x}(t-1))^\top (\bar{x}(t-1) - z^*) \\ &\quad + \frac{L^2}{2n} \sum_{t=1}^T \alpha(t), \end{aligned}$$

where we have used the fact that

$$\frac{n}{2} \sum_{t=1}^{T-1} \|\bar{x}_t - z^*\|_2^2 \left(\frac{1}{\alpha(t+1)} - \frac{1}{\alpha(t)} - \mu \right) = 0.$$

Using Corollary 5.6 and Lemma 5.7, we have

$$\begin{aligned} \mathbf{R}(T) &\leq \frac{8L}{\delta(1-\lambda)} \left(\sum_{i=1}^n \|x_i(0)\|_1 + \frac{L}{\mu} (1+\ln(T)) \right) \\ &\quad + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ &\quad + \mu \frac{8n}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 \|\bar{x}(0) - z^*\|_2 \\ &\quad + \frac{8}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 L (1+\ln(T)) \\ &\quad + \frac{8n}{\delta(1-\lambda)} L \|\bar{x}(0) - z^*\|_2 (1+\ln(T)) \\ &\quad + \frac{8}{\delta(1-\lambda)} \frac{L^2}{\mu} (1+\ln(T))^2 + \frac{L^2}{2n\mu} (1+\ln(T)). \end{aligned}$$

The proof then follows from rearranging the right-hand side. \blacksquare

In order to establish the proof of Theorem 5.2, using the previous result about the network regret, we provide an upper bound on the individual regrets.

Proposition 5.8: Let $\{f_1^t, f_2^t, \dots, f_n^t\}_{t=1}^T$ be a sequence of convex cost functions that satisfies Assumption 5.1. Suppose that the learning rate is chosen as $\alpha(t) = \frac{1}{\mu t}$, and the agents use (3), over a sequence of B -strongly connected graphs, to generate their states. Then for agent $j \in \mathcal{V}$, we have

$$\mathbf{R}^j(T) - \mathbf{R}(T) \leq \frac{16nL_j}{\delta(1-\lambda)} \left(\sum_{i=1}^n \|x_i(0)\|_1 + \frac{L}{\mu} (1+\ln(T)) \right).$$

Proof: First, note that

$$\begin{aligned} \mathbf{R}^j(T) - \mathbf{R}(T) &= \sum_{t=1}^T \sum_{i=1}^n f_i^t(z_j(t)) - \sum_{t=1}^T \sum_{i=1}^n f_i^t(z_i(t)) \\ &= \sum_{t=1}^T \sum_{i=1}^n (f_i^t(z_j(t)) - f_i^t(z_i(t))) \\ &\leq \sum_{t=1}^T \sum_{i=1}^n g_j(t)^\top (z_j(t) - z_i(t)) \\ &\leq \sum_{t=1}^T \sum_{i=1}^n L_j \|z_j(t) - z_i(t)\|_2. \end{aligned}$$

the last inequality follows from the convexity of cost functions and boundedness of subgradients. We also have that

$$\begin{aligned} & \|z_j(t+1) - z_i(t+1)\|_2^2 \\ &= \|z_j(t+1) - \bar{x}(t)\|_2^2 + \|z_i(t+1) - \bar{x}(t)\|_2^2 \\ &\quad - 2(z_j(t+1) - \bar{x}(t))^\top (z_i(t+1) - \bar{x}(t)) \\ &\leq \|z_j(t+1) - \bar{x}(t)\|_2^2 + \|z_i(t+1) - \bar{x}(t)\|_2^2 \\ &\quad + 2\|z_j(t+1) - \bar{x}(t)\|_2 \|z_i(t+1) - \bar{x}(t)\|_2 \\ &\leq 4 \left[\frac{8}{\delta} \left(\lambda^t \sum_{i=1}^n \|x_i(0)\|_1 + \sum_{s=1}^t \lambda^{t-s} \sum_{i=1}^n \|\alpha(s)g_i(s)\|_1 \right) \right]^2, \end{aligned}$$

where we used Cauchy-Schwarz inequality and the last inequality follows from Lemma 5.5. As a result

$$\|z_j(t) - z_i(t)\|_2 \leq \frac{16}{\delta} \left(\lambda^{t-1} \sum_{i=1}^n \|x_i(0)\|_1 + \sum_{s=1}^{t-1} \lambda^{t-1-s} \sum_{i=1}^n \|\alpha(s)g_i(s)\|_1 \right)$$

Now by choosing $\alpha(t) = \frac{1}{\mu t}$, we have

$$\begin{aligned} \sum_{t=1}^T \sum_{i=1}^n L_j \|z_j(t) - z_i(t)\|_2 &\leq \sum_{t=1}^T n L_j \left(\frac{16}{\delta} \lambda^{t-1} \sum_{i=1}^n \|x_i(0)\|_1 \right. \\ &\quad \left. + \sum_{s=1}^{t-1} \lambda^{t-1-s} \frac{L}{\mu s} \right) \\ &\leq \frac{16nL_j}{\delta(1-\lambda)} \left(\sum_{i=1}^n \|x_i(0)\|_1 \right. \\ &\quad \left. + \frac{L}{\mu} (1+\ln(T)) \right), \end{aligned}$$

which establishes the result. \blacksquare

Proof of Theorem 5.2: The proof of Theorem 5.2 follows by using the network regret bound in Theorem 5.3 and the bound on the difference between the network regret and the individual regret, obtained in Proposition 5.8. ■

It is worth noting that one can proceed with the proof of Theorem 5.2 if the learning rate is instead given by $\alpha(t) = \frac{C}{t}$ where $C \geq 1/\mu$ is a constant.

B. Dependency of the upper bound on the number of agents for Ramanujan graphs

It is fruitful to make the dependency on number of agents of the upper bound provided in Theorem 5.2 explicit, at least for some special cases. Motivated by the second statement of Lemma 5.5, let us consider the class of regular (undirected) graphs and in particular, the subclass of Ramanujan graphs.

Proposition 5.9: Suppose that $\{\mathcal{G}(t)\}_{t=1}^T$ is a B-strongly connected sequence of Ramanujan d -regular graphs, $d \geq 3$, of order n . Under the conditions of Theorem 5.2, we have

$$\begin{aligned} R^j(T) \leq & c_1 \frac{dn^2}{d-2\sqrt{d-1}} + c_2 \frac{dn}{d-2\sqrt{d-1}} (1 + \ln(T)) \\ & + c_3 \frac{d}{d-2\sqrt{d-1}} (1 + \ln(T))^2 \end{aligned}$$

for some constants $c_1, c_2, c_3 \in \mathbb{R}_{\geq 0}$.

Proof: Suppose $\mathcal{G}(t)$ is a Ramanujan d -regular graph with the unweighted adjacency matrix $A(t)$. Then, using [28, Definition 2.2], we have that $\sigma_2(A(t)) \leq 2\sqrt{d-1}$. We hence obtain $\lambda \leq \sigma_2(A(t)) \leq \frac{2\sqrt{d-1}}{d}$, where $A(t) = \frac{1}{d}A(t)$. Consider now the distributed online subgradient push-sum algorithm (3), with $A(t)$ as described. We also have $\delta = 1$ for regular graphs. Using Theorem 5.2, in particular (4), we have that

$$\begin{aligned} C_1 = & \frac{8L}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ & + \frac{8\mu n}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 \|\bar{x}(0) - z^*\|_2 \\ & + \frac{16nL_j}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 \end{aligned}$$

Now, using $\delta = 1$, $d \geq 3$, and $\lambda \leq \frac{2\sqrt{d-1}}{d}$, we have that

$$\begin{aligned} C_1 \leq & \frac{8Ld}{d-2\sqrt{d-1}} \sum_{i=1}^n \|x_i(0)\|_1 + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ & + \frac{8\mu nd}{d-2\sqrt{d-1}} \sum_{i=1}^n \|x_i(0)\|_1 \|\bar{x}(0) - z^*\|_2 \\ & + \frac{16nL_j}{d-2\sqrt{d-1}} \sum_{i=1}^n \|x_i(0)\|_1 \end{aligned}$$

Finally, using $\sum_{i=1}^n \|x_i(0)\|_1 \leq n \max_{i \in \mathcal{V}} \|x_i(0)\|_1$, we conclude that

$$\begin{aligned} C_1 \leq & \frac{8Lnd}{d-2\sqrt{d-1}} \max_{i \in \mathcal{V}} \|x_i(0)\|_1 + \frac{n}{2\alpha(1)} \|\bar{x}(0) - z^*\|_2^2 \\ & + \frac{8\mu n^2 d}{d-2\sqrt{d-1}} \max_{i \in \mathcal{V}} \|x_i(0)\|_1 \|\bar{x}(0) - z^*\|_2 \\ & + \frac{16n^2 d L_j}{d-2\sqrt{d-1}} \max_{i \in \mathcal{V}} \|x_i(0)\|_1 \\ \leq & c_1 \frac{n^2 d}{d-2\sqrt{d-1}}, \end{aligned}$$

where

$$\begin{aligned} c_1 = & \max_{i \in \mathcal{V}} \|x_i(0)\|_1 \left(\frac{8L}{n} + 8\mu \|\bar{x}(0) - z^*\|_2 \right. \\ & \left. + 16L_j \right) + \frac{\|\bar{x}(0) - z^*\|_2^2}{2\mu n}. \end{aligned}$$

Similarly, we have

$$\begin{aligned} C_2 = & \frac{8L^2}{\mu\delta(1-\lambda)} + \frac{8L}{\delta(1-\lambda)} \sum_{i=1}^n \|x_i(0)\|_1 + \frac{16nL_j}{\delta(1-\lambda)} \frac{L}{\mu} \\ & + \frac{8n}{\delta(1-\lambda)} L \|\bar{x}(0) - z^*\|_2 + \frac{L^2}{2n\mu} \end{aligned}$$

Using $\delta = 1$ and $\lambda \leq \frac{2\sqrt{d-1}}{d}$, we have that

$$\begin{aligned} C_2 \leq & \frac{8dL^2}{\mu(d-2\sqrt{d-1})} + \frac{8Ld}{d-2\sqrt{d-1}} \sum_{i=1}^n \|x_i(0)\|_1 \\ & + \frac{16ndL_j}{d-2\sqrt{d-1}} \frac{L}{\mu} \\ & + \frac{8nd}{d-2\sqrt{d-1}} L \|\bar{x}(0) - z^*\|_2 + \frac{L^2}{2n\mu}, \end{aligned}$$

Hence, using $\sum_{i=1}^n \|x_i(0)\|_1 \leq n \max_{i \in \mathcal{V}} \|x_i(0)\|_1$, we conclude that

$$\begin{aligned} C_2 \leq & \frac{8dL^2}{\mu(d-2\sqrt{d-1})} + \frac{8Lnd}{d-2\sqrt{d-1}} \max_{i \in \mathcal{V}} \|x_i(0)\|_1 \\ & + \frac{16ndL_j}{d-2\sqrt{d-1}} \frac{L}{\mu} \\ & + \frac{8nd}{d-2\sqrt{d-1}} L \|\bar{x}(0) - z^*\|_2 + \frac{L^2}{2n\mu}, \\ \leq & c_2 \frac{nd}{d-2\sqrt{d-1}}, \end{aligned}$$

where

$$c_2 = \frac{8L^2}{\mu n} + 8L \max_{i \in \mathcal{V}} \|x_i(0)\|_1 + \frac{16LL_j}{\mu} + 8L \|\bar{x}(0) - z^*\|_2 + \frac{L^2}{2n^2\mu}$$

Finally, we have

$$C_3 = \frac{8}{\delta(1-\lambda)} \frac{L^2}{\mu} \leq c_3 \frac{d}{d-2\sqrt{d-1}},$$

where

$$c_3 = \frac{8L^2}{\mu},$$

which yields the result. ■

Note that, using this result, for large values of T , the upper bound grows linearly with the size of the network n .

VI. SENSOR NETWORKS

We provide an example using our results for localization in sensor networks, motivated by [20]. Consider a network of n sensors, which is used to observe a vector $s \in \mathbb{R}^d$. Each sensor $i \in \mathcal{V}$, at each time $t \in \{1, \dots, T\}$, receives an observation vector $q_i^t \in \mathbb{R}^{d_i}$, which is time-varying due to, say, observation noise. Each sensor i is assumed to have a linear model of the form $p_i(s) = P_i s$, where $P_i \in \mathbb{R}^{d_i \times d}$ and $P_i v = 0$ if and only if $v = 0$. The best estimation for s is the vector $\hat{s} \in \mathbb{R}^d$ that minimizes the cost function

$$f(\hat{s}) = \sum_{t=1}^T \sum_{i=1}^n \frac{1}{2} \|q_i^t - P_i \hat{s}\|_2^2.$$

The observation vector is modeled as $q_i^t = P_i s + w_i^t$ where w_i^t is assumed to be white noise. In the offline setting, we have all the information to compute the optimal estimate, which is given by

$$s^* = \frac{1}{T} \sum_{t=1}^T \left(\sum_{i=1}^n P_i^\top P_i \right)^{-1} \left(\sum_{i=1}^n P_i^\top q_i^t \right).$$

As we describe shortly, when the noise characteristics are not known, or in some cases where some sensors fail to work properly, we can use a distributed online algorithm to find an estimate for the state s . Here, we consider a scenario in which $d = 1$ and a network of 100 sensors is used to observe. At each time step $t \in \{1, \dots, T\}$, a random directed graph is generated, describing the sensor communication. This random directed graph, denoted by $\mathcal{G}(n, p, r)$, where r is an even number and is generated as follows: First, we label each vertex a number from 1 to n and we generate an r -regular directed graph of order n , which has rn edges by imposing that vertex i and vertex j are connected by two directed edges if $|i - j| \leq r/2$ or $|i - j| \geq n - r/2$. Then we delete each edge, independently of others, with probability p . Next, among all the vertices that are incident to the set of deleted edges, say N edges, we randomly choose N ordered pairs and connect each pair with a directed edge. Now we have a random directed graph of order n with rn edges.

In our model, sensor i observes $q_i^t = a_i^t s + b_i^t$, where $a_i^t \in [a_{\min}, a_{\max}]$ and $b_i^t \in [b_{\min}, b_{\max}]$ are chosen at random from a uniform distribution. The cost function for sensor i at each time t is given by the mapping $f_i^t : \mathbb{R} \rightarrow \mathbb{R}$, where $f_i^t(\hat{s}) = \frac{1}{2} (q_i^t - P_i \hat{s})^2$ and $P_i \in \mathbb{R}$. We use the distributed online subgradient push-sum algorithm to estimate the state s . We consider three scenarios:

- 1) sensors have the same observation model, i.e., the model we use for q_i^t is the same for all sensors, and can communicate over a sequence of time-varying directed graphs;
- 2) sensors have the same observation model, but they cannot communicate with each other;
- 3) sensors have different observation models and they can communicate over a sequence of time-varying directed graphs.

In what follows, we simulate the sensors' state estimation over time and study the sensors' regret for each of these scenarios.

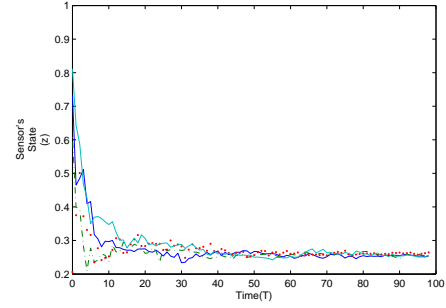


Fig. 1. Sensors' state estimation vs. time for four of the sensors are shown. The network consists of 100 sensors communicating over a sequence of $\mathcal{G}(100, 0.2, 3)$ random directed graph. The i th sensor observes $q_i^t = a_i^t s + b_i^t$, where a_i^t and b_i^t are random variables chosen from $[0, 2]$ and $[-\frac{1}{2}, \frac{1}{2}]$, respectively, with a uniformly probability distribution. We use distributed online subgradient push-sum algorithm to estimate \hat{s} which minimizes the cost function $f(\hat{s}) = \sum_{t=1}^T \sum_{i=1}^n \frac{1}{2} (q_i^t - P_i \hat{s})^2$. The result demonstrates consensus among sensors.

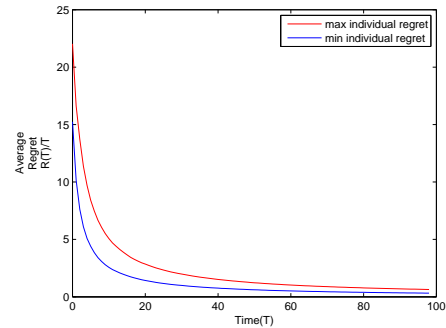


Fig. 2. Average regrets over time ($R^j(T)/T$) vs. T for two sensors with the maximum and minimum average regrets are shown, where the same assumptions as the ones in Figure 1 hold.

1) Same observation model with communication: We assume the actual value $s = 1/4$ which is unavailable to sensors. Each sensor $i \in \mathcal{V}$, at each time $t \in \{1, \dots, T\}$ observes q_i^t . In this model, we assume $q_i^t = a_i^t s + b_i^t$, where a_i^t and b_i^t are chosen at random from a uniform distribution on $[0, 2]$ and $[-\frac{1}{2}, \frac{1}{2}]$, respectively. We also have $P_i = 1$, for all $i \in \{1, \dots, n\}$, which is the expected value of random variable a_i^t . The communication topology is given by a time-varying $\mathcal{G}(100, 0.2, 2)$ random directed graph.

Figure 1 shows the states of four sensors over 100 time iterations. By using the distributed online subgradient push-sum algorithm (3), the subgradient of cost functions and the communication between sensors result in a consensus between sensors as shown in the figure. The consensus value is $\frac{1}{4}$, the expected value of sensor observations. Figure 2 shows the average individual regret of the two sensors with the maximum and minimum average regrets over time.

In the previous example, the expected value of the minimizer of the cost functions for each sensor is the same. Therefore, if each sensor uses an online algorithm without communicating with other sensors, they converge to the same value; however, the communication might accelerate this convergence, as demonstrated next.

2) Same observation model without communication:

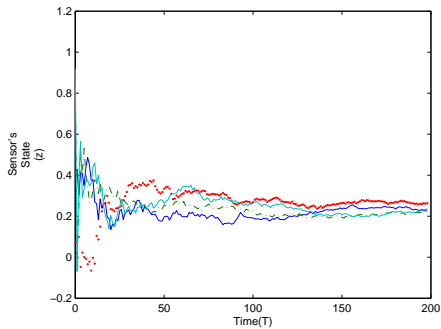


Fig. 3. Sensors' state estimation vs. time for four of the sensors are shown. The network consists of 100 sensors with no communications. The i th sensor observes $q_i^t = a_i^t s + b_i^t$, where a_i^t and b_i^t are chosen at random from a uniform distribution on $[0, 2]$ and $[-0.5, 0.5]$, respectively. We use distributed online subgradient push-sum algorithm to estimate \hat{s} which minimizes the cost function $f(\hat{s}) = \sum_{t=1}^T \sum_{i=1}^n \frac{1}{2} (q_i^t - P_i \hat{s})^2$.

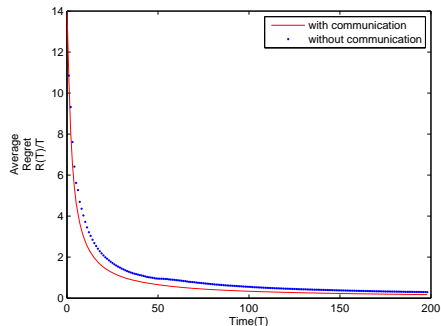


Fig. 4. Average individual regrets vs. time for one sensor, picked at random among 100 sensors, in the presence and absence of communications over time are shown, where the same assumptions as the ones in Figure 3 hold, and we consider two cases: First, there is communication between sensors and second, there is no communication between them. The results shows that communication gives a better regret.

Consider a scenario with the assumptions as before, with the exception that there is no communication between sensors. Figure 3 and Figure 4 show, respectively, the estimates of four sensors and the average individual regret of one sensor, picked at random, in the presence and absence of communications over time.

3) Different observation model with communication: Consider a scenario with the same assumptions as above, with the exception that the observation vector $q_i^t = a_i^t s + b_i^t$ is available to sensor i , where a_i^t and b_i^t are chosen at random from a uniform distribution on $[0, 2]$ and $[-0.5 + \frac{i-50}{100}, 0.5 + \frac{i-50}{100}]$, respectively. In this sense, and in contrast to the previous case, sensors do not use the same observations model. The communication network is a time-varying $\mathcal{G}(100, 0.2, 2)$ random directed graph. We use the distributed online subgradient push-sum algorithm to estimate \hat{s} . The consensus among sensors is shown in Figure 5, where the sensors' estimates approach the expected value of sensor observation. Figure 6 shows the individual regret goes to zero as time increases without bound.

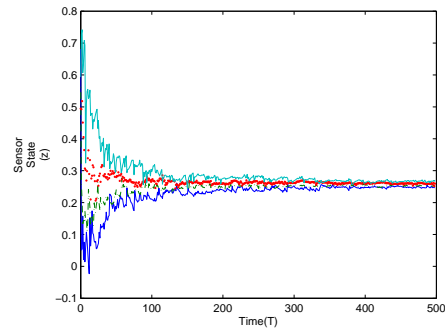


Fig. 5. Sensors' state estimation vs. time for four of the sensors are shown. The network consists of 100 sensors communicating through a sequence of $\mathcal{G}(100, 0.2, 2)$ random directed graph. The i th sensor observes $q_i^t = a_i^t s + b_i^t$ where a_i^t and b_i^t are chosen at random from a uniform distribution on $[0, 2]$ and $[-0.5 + \frac{i-50}{100}, 0.5 + \frac{i-50}{100}]$, respectively. We use distributed online subgradient push-sum algorithm to estimate \hat{s} which minimizes the cost function $f(\hat{s}) = \sum_{t=1}^T \sum_{i=1}^n \frac{1}{2} (q_i^t - P_i \hat{s})^2$. The result demonstrates consensus among sensors.

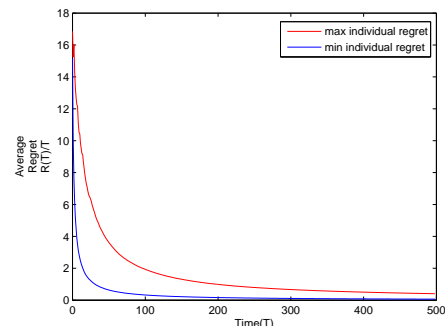


Fig. 6. Average individual regret over time $R^j(T)/T$ vs. time for two sensors is shown, one has the maximum average regret and the other one has the minimum average regret, where the same assumptions as the ones in Figure 5 hold.

VII. CONCLUSIONS AND FUTURE WORK

We have introduced a subgradient-push discrete-dynamical system for distributed online convex optimization, where agents can communicate their state estimates over a sequence of time-varying directed graphs. Under the assumption that agents' cost functions are locally Lipschitz and locally strongly convex, we have proved that the proposed algorithm achieves sublinear worst-case regret bound on any sequence of uniformly strongly connected time-varying directed graphs. In particular, by choosing a suitable learning rate, we have shown that the network regret bound is logarithmic, up to a square. Although, this bound is slightly worse than the known regret bounds in the centralized case, the algorithm works for general time-varying network topologies. We also showed that the individual regret bound grows linearly by the size of network for Ramanujan graphs. Improving the regret bound, considering scenarios with constraints, extending the results to general convex functions, investigating other application areas, and studying the regret bound on special classes of graphs are among the avenues for future work.

REFERENCES

- [1] M. Akbari, B. Ghahsifard, and T. Linder, "Distributed subgradient-push online convex optimization on time-varying directed graphs," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 264–269, Sept 2014.
- [2] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Symposium on Information Processing of Sensor Networks*, (Berkeley, CA), pp. 20–27, Apr. 2004.
- [3] 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.
- [4] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [5] J. W. Durham, A. Franchi, and F. Bullo, "Distributed pursuit-evasion without global localization via local frontiers," *Autonomous Robots*, vol. 32, no. 1, pp. 81–95, 2012.
- [6] J. How, R. Twiggs, D. Weidow, K. Hartman, and F. Bauer, "Orion - a low cost demonstration of formation flying in space using GPS," in *AIAA/AAS Astrodynamics Specialist Conf. and Exhibit*, (Reston, VA), pp. 276–286, 1998.
- [7] A. D. Dominguez-Garcia, S. T. Cady, and C. N. Hadjicostis, "Decentralized optimal dispatch of distributed energy resources," in *IEEE Conf. on Decision and Control*, (Hawaii, USA), pp. 3688–3693, Dec. 2012.
- [8] P. Wan and M. D. Lemmon, "Event-triggered distributed optimization in sensor networks," in *Symposium on Information Processing of Sensor Networks*, (San Francisco, CA), pp. 49–60, 2009.
- [9] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [10] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Control and Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.
- [11] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," in *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*, pp. 170–174, IEEE, 2013.
- [12] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the Twentieth International Conference on Machine Learning*, (Washington, D.C.), pp. 928–936, 2003.
- [13] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, 2007.
- [14] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.
- [15] M. Raginsky, N. Kiarashi, and R. Willett, "Decentralized online convex programming with local information," in *American Control Conference*, (San Francisco, CA), pp. 5363–5369, 2011.
- [16] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 11, pp. 2483–2493, 2013.
- [17] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 1484–1489, IEEE, 2013.
- [18] D. Mateos-Núñez and J. Cortés, "Distributed online second-order dynamics for convex optimization over switching connected graphs," in *Mathematical Theory of Networks and Systems*, (Groningen, The Netherlands), pp. 15–22, 2014.
- [19] D. Mateos-Nunez and J. Cortes, "Distributed online convex optimization over jointly connected digraphs," *IEEE Transactions on Network Science and Engineering*, vol. 1, pp. 23–37, Jan 2014.
- [20] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization on dynamic networks," 2014. <http://arxiv.org/pdf/1412.7215.pdf>.
- [21] B. Ghahsifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.
- [22] K. I. Tsianos and M. G. Rabbat, "Efficient distributed online prediction and stochastic optimization with approximate distributed mini-batches," 2014. <http://arxiv.org/abs/1406.2075v1>.
- [23] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *IEEE Symposium on Foundations of Computer Science*, (Washington, DC), pp. 482–491, Oct. 2003.
- [24] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 5453–5458, 2012.
- [25] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.
- [26] A. Nedic and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," 2014. <http://arxiv.org/abs/1406.2075v1>.
- [27] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 1st ed., 2003.
- [28] A. Lubotzky, R. Phillips, and P. Sarnak, "Explicit expanders and the ramanujan conjectures," in *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, (New York, NY, USA), pp. 240–246, ACM, 1986.

APPENDIX

In this section, we study the boundedness of agents' states, where agents use (3) to generate the sequence $\{z(t)\}_{t=1}^T$ over a sequence of B-strongly connected graphs. We assume that the sequence of cost functions $\{f_1^t, \dots, f_n^t\}_{t=1}^T$ satisfies Assumption 5.1.

Lemma A.1: For $T \in \mathbb{R}_{>0}$, let $\{f_1^t, \dots, f_n^t\}_{t=1}^T$ be a sequence of convex functions on \mathbb{R}^d with nonempty set of minimizers, where each f_i^t has L_i -bounded subgradient set. Let

$$\cup_{i=1}^n \cup_{t=1}^T \operatorname{argmin} f_i^t \subset \bar{B}(0, K_1),$$

for some $K_1 \in \mathbb{R}_{>0}$ independent of T , and assume $\{f_1^t, \dots, f_n^t\}_{t=1}^T$ are β -central on $\mathbb{R}^d \setminus \bar{B}(0, K_1)$, where $\beta \in (0, 1]$. Then, for any sequence $\{(z_1(t), \dots, z_n(t))\}_{t=1}^T$ and $\{\bar{x}(t)\}_{t=1}^T$ generated by (3) over a sequence of B-strongly connected graphs, and any sequence of learning rates $\{\alpha(t)\}_{t=1}^T$, we have

$$\|\bar{x}(t)\|_2 \leq r_\beta + \frac{L}{n} \max_{s \geq 1} \alpha(s) + \|\bar{x}(0)\|_2, \quad (18)$$

$$\|z_i(t)\|_2 \leq H(\beta),$$

for all $t > 0$, where $L = \sum_{i=1}^n L_i$,

$$r_\beta = \max \left\{ \frac{K_1 + K_2}{\beta \sqrt{1 - \epsilon^2} - \epsilon \sqrt{1 - \beta^2}}, \frac{L}{2n\epsilon} \max_{s \geq 1} \alpha(s) \right\},$$

$$H(\beta) = r_\beta + \|\bar{x}(0)\|_2 + K_2 + \frac{L}{n} \max_{s \geq 1} \alpha(s), \quad (19)$$

with $\epsilon \in (0, \beta)$ and $K_2 \in \mathbb{R}_{>0}$.

Proof: First we prove the boundedness of $\|\bar{x}(t)\|_2$ by induction on t . Note that the initial condition $\|\bar{x}(0)\|_2$ satisfies (18). Using (8), we conclude that if $\bar{x}(t) \in \bar{B}(0, r_\beta)$, then

$$\bar{x}(t+1) \in \bar{B}\left(0, r_\beta + \frac{L}{n} \max_{s \geq 1} \alpha(s)\right).$$

By an argument very similar to the one in the proof [19, Lemma V.7], we have that if $\bar{x}(t) \in \mathbb{R}^d \setminus \bar{B}(0, r_\beta)$, then $\|\bar{x}(t+1)\|_2 \leq \|\bar{x}(t)\|_2$. Next, using Lemma 5.5, for all t , we have that

$$\|z_i(t+1) - \bar{x}(t)\|_2 \leq K_2,$$

for some $K_2 \in \mathbb{R}_{>0}$. We hence conclude that $\|z_i(t)\|_2 \leq H(\beta)$, where $H(\beta)$ is given in (19). ■

Theorem A.2: For $T \in \mathbb{R}_{>0}$, let $\{f_1^t, \dots, f_n^t\}_{t=1}^T$ be a sequence of convex functions on \mathbb{R}^d with nonempty set of minimizers, where each f_i^t has L_i -bounded subgradient set. Let

$$\cup_{i=1}^n \cup_{t=1}^T \operatorname{argmin} f_i^t \subset \bar{B}(0, K_1/2),$$

for some $K_1 \in \mathbb{R}_{>0}$ independent of T . Suppose that $\{f_1^t, \dots, f_n^t\}_{t=1}^T$ is a sequence of μ -strongly convex functions on $\bar{B}(0, H(\frac{\mu K_1}{2L}))$, for some $\mu \in \mathbb{R}_{>0}$, where $H(\cdot)$ is defined in (19). Then $\{z_i(t)\}_{i=1}^n$, generated by (3) over a sequence of B -strongly connected graphs, stays in $\bar{B}(0, H(\frac{\mu K_1}{2L}))$, for all t .

Proof: By an argument very similar to the one in the proof [19, Lemma V.6], $K_1 < r_\beta < H(\frac{\mu K_1}{2L})$ and hence $K_1 < H(\frac{\mu K_1}{2L})$. Thus, f_i^t is μ -strongly convex in $\bar{B}(0, K_1)$ and an application of [19, Lemma V.9] implies that each f_i^t is β' -central on $\mathbb{R}^d \setminus \bar{B}(0, K_1)$, where $\beta' \leq \frac{\mu K_1}{2L}$. Hence, the assumptions of Lemma A.1 are satisfied with $\beta = \frac{\mu K_1}{2L}$ and as a result, $z_i(t)$ remains in the region $\bar{B}(0, H(\frac{\mu K_1}{2L}))$. ■