Distributed Online Optimization on time-varying networks

by

Mohammad Akbari Varnousfaderani

A thesis submitted to the
Department of Mathematics and Statistics
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
August 2015

Abstract

This thesis introduces two classes of discrete-time distributed online optimization algorithms, with a group of agents which communicate over a network. At each time, a private 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. We design algorithms distributed over the network topologies, which guarantee 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. One algorithm is based on gradient-flow, which provably works for a sequence of time-varying uniformly strongly connected graphs. The other one is based on Alternating Direction Method of Multipliers, which works on fixed undirected graphs and gives an explicit regret bound in terms of the size of the network. We implement the proposed algorithms on a sensor network and the results demonstrate the good performance for both algorithms.

Acknowledgments

First and foremost, I would like to thank my supervisors, Prof. Bahman Gharesifard and Prof. Tamás Linder for their guidance, patience and support. I thank them for giving me the opportunity of studying and doing research under their supervision.

I would also like to thank Prof. Abdol-Reza Mansouri for his excellent teaching and support in the Modern Control Theory Course.

I am also grateful to my family, Yadollah, Fatemeh, Marzieh, and Hamed for their love and support.

Finally, I would like to thank all my teachers that help me to find my path to the academic world.

Contents

List of Figures

Chapter 1

Introduction

The area of network multi-agent systems has received considerable attention in recent years. These systems have a variety of applications such as traffic control and transportation, power engineering, manufacturing, and supply chain management. Multi-agent system models provide more natural ways of representing task allocation, team planning, and other tasks. In addition, these systems have the advantage of reducing data transmission rates, distributing computational resources across a network, and ensuring robustness in the presence of local failures.

Many scenarios concerning the coordination of multi-agent systems can be investigated in the framework of distributed optimization. In this framework, a group of agents cooperatively attempt to minimize a common cost function, which is distributed among them, and each agent has only limited (private) information about the cost function. The main feature of any implementable coordination protocol is that the agents only use the information from their neighboring agents, where the neighborhood structure is cast as a graph, to update their states. Each agent updates its states based on the states of its neighboring agents and the information of its own private cost function, in order to steer the agents' states to reach a consensus in the set of minimizers of the common cost function [29, 25, 36]. The problem has a variety of applications including localization and robust estimation [7], formation control [16], and energy dispatch in power distribution networks [6], and has been extensively studied in recent years [29, 25, 37, 26, 18, 36].

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 distributed *online* convex optimization, where the functions allocated to each agent possibly change with time. In this framework, an individual convex cost function is revealed to each agent after the agent has chosen its state. Because of this, the agents only see their cost functions in hindsight and hence their states do not necessarily correspond to the minimizers of these functions and each agent incurs a so-called *individual regret*. The individual regret for each agent is defined as the difference between the accumulated collective cost incurred by the agent's state estimation and the cost obtained by the best fixed state, when all functions are known in advance [21]. The objective here is to design algorithms such that the regret function will be sublinear in terms of time; in other words, the algorithm drives the average regret over time to zero.

There is a large body of research on online optimization problems; we refer the interested reader to [44, 11, 31, 5]. Several distributed online optimization algorithms exist in the literature [21, 14, 34]. In this thesis, we propose two different classes of distributed online optimization algorithms. The first algorithm is built on consensus-based gradient-descent methods that have been used extensively for online convex optimization; see [21, 13]. The second algorithm is based on the Alternating Direction Method of Multipliers (ADMM), which has been used for distributed optimization with linear constraint, see [3].

With the interest in decentralized architectures and motivated by the problem of distributed convex optimization, a distributed version of online optimization is proposed in [30, 43]. In [13] and [22, 21] consensus-based gradient-descent algorithms for distributed online optimization are proposed. In this setting, each agent aims at driving its individual average regret to zero. Given that the agents do not have access to the local cost functions of other agents, these individual regrets are not computable by the agents, 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 [13], and is extended in [14], to allow for time-varying weights, but on a fixed directed graph. In [22, 21], motivated by the saddle-point dynamics in [9], a discrete-time distributed online convex optimization algorithm on weight-balanced network topologies is introduced; in particular, the suggested protocol in [21] works on jointly connected weight-balanced digraphs. Other recent work includes [35], 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 [19, 33] 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 [27, 23]. In contrast, here we are interested in distributed *online* optimization.

The second algorithm that we investigate in this work is based on the Alternating Direction Method of Multipliers (ADMM), which is suitable for large-scale constrained optimization problems [3]. It has already been used for distributed optimization [3, 40, 41], by casting the "consensus" step as a linear constraint. In a nutshell, the ADMM splits this (linearly) constrained optimization problem into two subproblems. Interestingly, although each step of the computation is more expensive than the one in the subgradient algorithms, using ADMM, one can achieve a convergence rate of $\mathcal{O}(\frac{1}{T})$ for distributed optimization, in contrast to the $\mathcal{O}(\frac{1}{\sqrt{T}})$ rate of consensus-based methods [40]. In this work, we investigate the application of such distributed optimization protocols in online settings, and the regret bound that can be guaranteed using them. In the centralized setting, it is already known that ADMM achieves a regret bound similar to the subgradient algorithms [39]. A distributed version of the ADMM is proposed for minimizing the network regret in [15]. Unlike this work, we consider the more challenging problem of minimizing the individual regret. Moreover, our proposed algorithm does not have subgradient step, in contrast to [15].

1.1 Contributions of Thesis

The contributions of this thesis are as follows. We propose two classes of distributed online convex optimization algorithms. In each algorithm, we consider a network of agents communicating with each other. At each time instance, each agent uses the information about the states of its neighboring agents and makes a decision about its next state. After that, the agent receives a convex cost function and incurs a cost for its state estimation. Following the framework of [13] and [21], the regret for each individual agent is defined as the difference between the accumulated collective cost of the network incurred by the agent's state estimation and the cost of the best fixed state, made by a decision maker that has access to the objective functions.

For the first algorithm, we assume that the network topology is described by a sequence of time-varying directed graphs. We also assume that the individual cost functions are strongly convex on a compact neighborhood of their minimizers and have bounded subgradients. The proposed algorithm is based on subgradient push-sum algorithm introduced in [23], which achieves a 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. 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 long time horizon, this upper bound grows linearly with the size of the network.

For the second algorithm, we assume that the network topology is described by an undirected fixed graph. We introduce an online protocol, distributed over the graph and motivated by the ADMM algorithm [3], through which agents can achieve a sublinear individual regret of $\mathcal{O}(\sqrt{T})$.

Unlike the subgradient flow algorithms that use projections for constrained online optimization, the proposed online ADMM protocol is projection-free and more importantly provably works in distributed settings, in the presence of linear constraints. Moreover, this algorithm is gradient free, in the sense that it does not need to compute gradient in the algorithm (on the other hand, the ADMM relies on instantaneous calculation of the minimizer of a constrained convex optimization problem, see [3]). In addition to the fact that the proposed distributed online ADMM algorithm has the same performance guarantee in terms for the regret as the best known centralized algorithms, this algorithm gives a regret that can be written explicitly in terms of the size of the network.

Finally, we discuss an application of the proposed algorithms to a sensor network estimation problem, where a group of sensors with independent observations cooperatively attempt to estimate a target. Each sensor's observation is corrupted by noise which makes this problem fit into the framework of distributed online optimization. Our results demonstrate the excellent performance of both our algorithms.

1.2 Organization of Thesis

The rest of this thesis is organized as follows. Chapter 2 contains some mathematical preliminaries on linear algebra, convex analysis, and graph theory, that we use to formulate and derive our results. In Chapter 3, we introduce the centralized and distributed online convex optimization problems. In Chapter 4, we propose our distributed online subgradient push-sum algorithm which achieves sublinear regret. In this chapter we assume time-varying directed graphs and convex cost functions. These assumptions lead to results which are used to derive a sublinear individual regret bound. Chapter 5 contains the results on the Distributed Online ADMM algorithm. We illustrate the performance of the results by simulations on localization in sensor network in Chapter 6. Finally, Chapter 7 gives conclusions and ideas for future work.

Chapter 2

Mathematical Preliminaries

In this chapter, we introduce some notational conventions and background that we use throughout the thesis.

2.1 Basic Notions from Analysis

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{Z}$, and $\mathbb{Z}_{>0}$ 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. We recall the following results.

Proposition 2.1.1. (Schwarz Inequality[1, Proposition 1.1.2]): *For any two vectors $x, y \in \mathbb{R}^d$, we have*

$$|x^\mathsf{T} y| \leq \|x\|_2 \|y\|_2,$$

with equality holding if and only if $x = \alpha y$, for some $\alpha \in \mathbb{R}$.

Proposition 2.1.2. ([1, Proposition 1.1.7]): *For any two norms $\|\cdot\|$ and $\|\cdot\|'$ on $\mathbb{R}^d$, there exists $c \in \mathbb{R}_{>0}$ such that $\|x\| \leq c\|x\|'$ for all $x \in \mathbb{R}^d$.*

Using Proposition 2.1.1, we have the following corollary.

Corollary 2.1.3. *For any vector $x \in \mathbb{R}^d$, we have $\|x\|_2 \leq \|x\|_1 \leq \sqrt{d}\|x\|_2$.*

We also denote by $\mathcal{B}(x,r) = \{y \in \mathbb{R}^d \mid \|y-x\|_2 < r\}$ and $\bar{\mathcal{B}}(x,r) = \{y \in \mathbb{R}^d \mid \|y-x\|_2 \leq r\}$, the open and the closed balls of radius r centered at $x \in \mathbb{R}^d$, respectively. We use the short-hand notation $\mathbf{1}_d = (1,\ldots,1)^\top \in \mathbb{R}^d$ and $\mathbf{0}_d = (0,\ldots,0)^\top \in \mathbb{R}^d$. 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 $A \in \mathbb{R}^{d_1\times d_2}$ 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 ith largest singular value of matrix A. For the matrix A, we denote by $[A]^i$ and $[A]_j$, the ith row and the jth column of the matrix A, respectively.

Definition 2.1.4. ([21]): *Given $w \in \mathbb{R}^d \backslash \{\mathbf{0}\}$ and $c \in [0,1]$, we let*

$$\mathcal{H}_c(w) := \{v \in \mathbb{R}^d : v^\top w \geq c\|v\|_2\|w\|_2\} \tag{2.1}$$

denote the convex cone of vectors in $\mathbb{R}^d$ whose angle with w has a cosine lower bounded by c.

The function $f : \mathbb{R}^d \to \mathbb{R}$ is called Lipschitz, if for all $x, y \in \mathbb{R}^d$, $|f(x) - f(y)| \leq C\|x-y\|_2$ for some $C \in \mathbb{R}_{\geq 0}$.

2.2 Convex Analysis

In this section, we present some notions from convex analysis. A subset $S \subseteq \mathbb{R}^d$ is called convex if for all $x, y \in S$ and for all $\lambda \in [0,1]$, we have $\lambda x + (1-\lambda)y \in S$. We denote by $\mathrm{conv}(X)$, the convex hull of the set X, the intersection of all convex sets containing X. A function $f : S \to \mathbb{R}$ is convex if for all x, y in the convex set $S \subseteq \mathbb{R}^d$ and for all $\lambda \in [0, 1]$, we have

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

Also, the function $f : S \to \mathbb{R}$ is strictly convex if for all $x \neq y \in S$ and for all $\lambda \in (0, 1)$, we have

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

Definition 2.2.1. ([1]): *For an (extended)-real-valued function $f : S \to \mathbb{R} \cup \{+\infty\}$, where $S \subseteq \mathbb{R}^d$, the epigraph of f is defined as the subset of $\mathbb{R}^{d+1}$, given by*

$$\mathrm{epi}(f) = \{(x, w) \mid x \in S, w \in \mathbb{R}, f(x) < w\}.$$

The effective domain of f is defined as the set

$$\mathrm{dom}(f) = \{x \in S \mid f(x) < +\infty\}$$

We say f is proper if $f(x) < +\infty$ for at least one $x \in S$.

Definition 2.2.2. ([1, Definition 1.2.4]): *Let $S \subseteq \mathbb{R}^d$ be a convex set. An extended-real-valued function $f : S \to \mathbb{R} \cup \{+\infty\}$ is called convex if $\mathrm{epi}(f)$ is a convex subset of $\mathbb{R}^{d+1}$. The function f is also called closed, if the $\mathrm{epi}(f)$ is a closed set.*

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

f at x, if

$$f(y) - f(x) \geq g_x^{\mathsf{T}}(y - x),$$

for all $y \in S$. We denote by $\partial f(x)$ the set of subgradients of f at x.

Proposition 2.2.3. ([1, Proposition 4.2.1]): *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function, then the set of subgradients of f is nonempty, convex, and compact for all $x \in \mathbb{R}^d$.*

Note that for a differentiable convex function f, the set of subgradient of f at x contains only the gradient 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 $x \in \mathbb{R}^d$. Note that a function with L-bounded subgradients is Lipschitz.

Definition 2.2.4. (Strong convexity): *The function $f : S \to \mathbb{R}$ is μ-strongly convex, for some $\mu \in \mathbb{R}_{>0}$, if for each $x \in S$ and $g_x \in \partial f(x)$, we have*

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

for all $y \in S$.

We have the following proposition.

Proposition 2.2.5. *$f : S \to \mathbb{R}$ is μ-strongly convex on S if and only if*

$$(g_y - g_x)^{\mathsf{T}}(y - x) \geq \mu\|y - x\|_2^2$$

for each $g_x \in \partial f(x)$ and $g_y \in \partial f(y)$, for all $x, y \in S$.

Definition 2.2.6. ([10]): *Let $f : S \to \mathbb{R}$ be a function on a set $S \subseteq \mathbb{R}^d$. The point $x^\star \in S$ is called* *a) a local minimizer of* f*, if* $f(x^\star) \le f(x)$ *for all* x *in some ball* $\mathcal{B}(x^\star, r)$ *and a strict local minimizer of* f *if* $f(x^\star) < f(x)$ *for all* x *in* $\mathcal{B}(x^\star, r)\backslash\{x^\star\}$*;*

b) a global minimizer of f *on* S*, if* $f(x^\star) \le f(x)$ *for all* $x \in S$ *and a strict global minimizer of* f *if* $f(x^\star) < f(x)$ *for all* $x \in S\backslash\{x^\star\}$*;*

c) a critical point of f*, if* f *is differentiable at* $x^\star$ *and* $\partial f(x) = \{\mathbf{0}_d\}$*;*

d) a saddle point of f*, if it is a critical point and there exist* y, z *in any ball* $\mathcal{B}(x^\star, r)$ *such that* $f(y) < f(x^\star) < f(z)$*.*

We let $\operatorname{argmin}(f)$ denote the set of global 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 $\operatorname{argmin}(f)$. We have the following property of convex functions.

Theorem 2.2.7. ([10]): *Let* $f : S \to \mathbb{R}$ *be a convex function on a convex set* $S \subseteq \mathbb{R}^d$*. Any local minimizer of* f *on* S *is a global minimizer of* f *on* S*. If* f *is strictly convex, then there exists at most one global minimizer of* f *on* S*.*

Proposition 2.2.8. (Weierstrass' Theorem [1, Proposition 2.2.1]): *Consider a closed proper function* $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ *and assume that one of the following three conditions holds:*

1. $\operatorname{dom}(f)$ *is bounded.*

2. *There exists a scalar* α *such that the level set* $\{x \mid f(x) \le \alpha\}$ *is nonempty and bounded.*

3. f *is coercive, i.e., for every sequence* $\{x_k\} \subset \mathbb{R}^d$ *such that* $\|x_k\|_2 \to \infty$*, we have* $\lim_{k\to\infty} f(x_k) = \infty$*.*

Then the set of minimizers of f is nonempty and compact.

We also recall the notion of β-centrality from [21].

Definition 2.2.9. (β-centrality): *For $\beta \in [0,1]$, a convex function $f : \mathbb{R}^d \to \mathbb{R}$ with $\operatorname{argmin}(f) \neq \emptyset$ is β-central on $Z \subset \mathbb{R}^d \setminus \operatorname{argmin}(f)$ if for each $x \in Z$, there exists $y \in \operatorname{argmin}(f)$ such that $-\partial f(x) \subseteq \mathcal{H}_\beta(y - x)$, i.e.,*

$$-g_x^\mathsf{T}(y - x) \geq \beta \|g_x\|_2 \|y - x\|_2,$$

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

2.3 Graph Theory

A weighted directed graph (or digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathsf{A})$ consists of a vertex set $\mathcal{V} = \{1, \cdots, N\}$, an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and an adjacency matrix $\mathsf{A} \in \mathbb{R}_{\geq 0}^{N \times N}$ with $a_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$, for $i, j \in \mathcal{V}$, and $a_{ij} = 0$ otherwise. Note that the edge set may contain self-loop. 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 i, respectively, as $\mathcal{N}_i^{\text{in}} = \{j \mid (j,i) \in \mathcal{E}\}$ and $\mathcal{N}_i^{\text{out}} = \{j \mid (i,j) \in \mathcal{E}\}$. The in- and out-degree of vertex i are, respectively, $d_i^{\text{in}} = |\mathcal{N}_i^{\text{in}}|$ and $d_i^{\text{out}} = |\mathcal{N}_i^{\text{out}}|$. The directed graph $\mathcal{G}$ is balanced if for every $i \in \mathcal{V}$, we have $d_i^{\text{in}} = d_i^{\text{out}}$ and it is weight-balanced if $\mathsf{A}\mathbf{1}_N = \mathsf{A}^\mathsf{T}\mathbf{1}_N$. The graph $\mathcal{G}$ is undirected if the edge set $\mathcal{E}$ is a set of unordered pair of vertices, i.e., $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ with the assumption that if $(i,j) \in \mathcal{E}$, then $(j,i) \in \mathcal{E}$. For an undirected graph $\mathcal{G}$, the set of neighbors of vertex i is defined as $\mathcal{N}(i) = \{j \mid (i,j) \in \mathcal{E}\}$. We define the Laplacian matrix $\mathsf{L}(\mathcal{G})$ for the graph $\mathcal{G}$ by

$$\mathsf{L}(\mathcal{G}) = \mathrm{diag}(\mathsf{A}\mathbf{1}_N) - \mathsf{A}.$$

The following properties of the Laplacian matrix can easily be verified [4].

1. $\mathsf{L}(\mathcal{G})\mathbf{1}_N = \mathbf{0}_N$, i.e., 0 is an eigenvalue of $\mathsf{L}(\mathcal{G})$ with eigenvector $\mathbf{1}_N$.

2. $\mathcal{G}$ is undirected if and only if $\mathsf{L}(\mathcal{G})$ is symmetric.

3. if $\mathcal{G}$ is strongly connected, then $\mathrm{rank}(\mathsf{L}(\mathcal{G})) = N-1$, i.e., 0 is a single eigenvalue of $\mathsf{L}(\mathcal{G})$.

Definition 2.3.1. ([23]): *We say the sequence of graphs $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t), \mathsf{A}(t)\}_{t=1}^{T}$ is uniformly strongly connected (or* B*-strongly-connected) if there exists* $\mathrm{B} \in \mathbb{Z}_{>0}$ *such that for each $k \in \mathbb{Z}_{\geq 0}$, the digraph with vertices $\mathcal{V}$ and edge set $\mathcal{E}_{\mathrm{B}}(k) = \bigcup_{t=k\mathrm{B}+1}^{(k+1)\mathrm{B}} \mathcal{E}(t)$ is strongly connected.*

Definition 2.3.2. ([20]): *A regular (undirected) graph is a graph where every vertex has the same number of neighbours. A regular graph with vertices of degree r is called an r-regular graph. A Ramanujan graph is an r-regular graph satisfying $\sigma_2(\mathsf{A}) \leq 2\sqrt{r-1}$, where $\mathsf{A} = [a_{ij}]$ is the unweighted adjacency matrix of the graph, i.e., $a_{ij} = 1$ if $(i,j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise.*

Chapter 3

Problem Statement

In this section, we describe the problem of distributed online convex optimization, which is developed from the online optimization problem for the multi-agent systems. We begin by describing the problem of online convex optimization.

3.1 Online Optimization

Suppose we have a sequence of convex cost functions $\{f^1, f^2, \cdots, f^T\}$, where $f^t : S \to \mathbb{R}$ for each $t \in \{1, \cdots, T\}$ ($T \in \mathbb{Z}_{>0}$ is the time horizon), where $S \subseteq \mathbb{R}^d$ is a convex set. At each time step $t \in \{1, \cdots, T\}$, a decision maker chooses an action $z(t) \in S$. After committing to this decision, a convex cost function $f^t : \mathbb{R}^d \to \mathbb{R}$ is revealed and the decision maker is faced with the cost of $f^t(z(t))$. We can summarize this as the following model [31], *Online Convex Optimization*

input: A convex set $S \in \mathbb{R}^d$

for $t = 1, 2, \cdots$

predict a vector $z(t) \in S \subseteq \mathbb{R}^d$

receive a convex cost function $f^t : \mathbb{R}^d \to \mathbb{R}$

suffer cost $f^t(z(t))$

In this scenario, due to lack of access to the cost function before the decision is made, the decision does not necessarily correspond to the minimizers and the decision maker is faced with 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 [44, 11]. Formally, the regret is

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

where

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

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

3.2 Distributed Online Optimization

Next, we review the setup for a distributed version of the online optimization problem [13, 21]. Consider a group of agents communicating with each other over a network, modeled by a graph, possibly directed and possibly time-varying, with properties that will be described shortly. At each time step $t \in \{1, 2, \cdots, T\}$, an agent $i \in \mathcal{V} = \{1, \cdots, N\}$ chooses its state $z_i(t) \in \mathbb{R}^d$. After this, a convex cost function $f_i^t : \mathbb{R}^d \to \mathbb{R}$ is revealed, and the agent incurs the cost $f_i^t(z_i(t))$. 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 estimates its state based on what it thinks the whole network would choose. Moreover, due to lack of access to the cost functions before the decisions are made, the decisions do not necessarily correspond to the minimizers and each agent faces with a regret.

The individual regret of agent $j \in \mathcal{V}$, see [13, 22], is now defined as

$$\mathsf{R}^j(T) := \sum_{t=1}^{T} \sum_{i=1}^{N} f_i^t(z_j(t)) - \sum_{t=1}^{T} \sum_{i=1}^{N} f_i^t(z^\star), \tag{3.1}$$

where

$$z^\star \in \underset{z \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{t=1}^{T} \sum_{i=1}^{N} f_i^t(z). \tag{3.2}$$

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.

The individual regret function is different from the *network* regret, which is defined as the difference between the collective accumulated cost over time of all agents and the cost resulting from the best offline fixed choice, selected by assuming that the information about the cost functions is available in advance. Specifically, we can write the network regret as

$$\mathsf{R}(T) := \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z_i(t)) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z^\star), \tag{3.3}$$

where

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

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.

The main objective of this thesis is to design distributed algorithms over some network topology which allow 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 in the time T. Our first proposed algorithm is given in Chapter 4 which works on time-varying uniformly strongly connected graphs. The second one will be introduced in Chapter 5, which works on a fixed undirected graph, but gives a regret bound which scales with the size of the network.

Chapter 4

Distributed Online Optimization on Time-Varying Directed Graphs

4.1 Distributed Online Subgradient Push-Sum Algorithm

In this section, we introduce a distributed online subgradient push-sum algorithm motivated by [27, 19], which allows the agents to have a sublinear average individual regret. To this end, let us consider a group of agents $\mathcal{V} = \{1, \cdots, N\}$ with the communication topology prescribed by a sequence of B-strongly-connected time-varying digraph $\{\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), \mathsf{A}(t))\}_{t=1}^{T}$. The distributed online subgradient push-sum algorithm is a discrete-time dynamical system, which is described next. We assume that at each time $t \in \{1, \cdots, T\}$, each agent has four state variables: $x_i(t) \in \mathbb{R}^d$, $y_i(t) \in \mathbb{R}$, $w_i(t) \in \mathbb{R}^d$ 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 $t \in \{1, \cdots, T\}$, the agent $i \in \mathcal{V}$ computes its next time state values by

$$\begin{aligned} w_i(t+1) =& \sum_{j \in \mathcal{N}_i^{\text{in}}(t)} \frac{x_j(t)}{d_j^{\text{out}}(t)}, \\ y_i(t+1) =& \sum_{j \in \mathcal{N}_i^{\text{in}}(t)} \frac{y_j(t)}{d_j^{\text{out}}(t)}, \\ z_i(t+1) =& \frac{w_i(t+1)}{y_i(t+1)}, \\ x_i(t+1) =& w_i(t+1) - \alpha(t+1) g_i^{t+1}(z_i(t+1)), \end{aligned}$$

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 $\alpha : \mathbb{Z}_{>0} \to \mathbb{R}_{>0}$ is the *learning rate*. Throughout the rest of this chapter, 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 $x_i(0) \in \mathbb{R}^d$ arbitrary 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 $j \in \mathcal{N}_i^{\text{in}}(t)$ 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. Note that we assume all agents has a self-loop, i.e., $i \in \mathcal{N}_i^{\text{in}}(t)$ for all $i \in \mathcal{V}$ and $t \in \mathbb{Z}_{>0}$.

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 $\mathbf{A}(t) = A(t) \otimes \mathsf{I}_d$, where

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

Note that the matrix $A(t)$ is column stochastic. The algorithm described above can now be written as

$$\begin{aligned} w(t+1) =& \mathbf{A}(t)x(t), \\ y(t+1) =& A(t)y(t), \\ z_i(t+1) =& \frac{w_i(t+1)}{y_i(t+1)}, \qquad \text{for all } i \in \mathcal{V}, \\ x(t+1) =& w(t+1) - \alpha(t+1)g(t+1), \end{aligned} \tag{4.1}$$

where $w(t) = (w_1^\mathsf{T}(t), \ldots, w_N^\mathsf{T}(t))^\mathsf{T}$, $x(t) = (x_1^\mathsf{T}(t), \ldots, x_N^\mathsf{T}(t))^\mathsf{T}$, $y(t) = (y_1(t), \ldots, y_N(t))^\mathsf{T}$, and $g(t) = (g_1^\mathsf{T}(t), \ldots, g_N^\mathsf{T}(t))^\mathsf{T}$.

4.2 Main Result

In this section, we show how the distributed online subgradient push-sum algorithm (4.1) can be used to bound the individual regret defined in (3.1). Before stating our main result, we specify the sequence of cost functions $\{f_1^t, f_2^t, ..., f_N^t\}_{t=1}^T$ that we consider in this chapter. This assumption selects a subset of convex functions with bounded subgradients, that are additionally strongly convex on a neighborhood, which we give a precise description for later on in Section 4.4.

Assumption 4.2.1. *$\{f_1^t, f_2^t, ..., f_N^t\}_{t=1}^T$ is a sequence of convex functions with nonempty set of minimizers, where for each $i \in \{1, \ldots, N\}$, the function f_i^t:*

1. *has L_i-bounded subgradients, where $L_i \in \mathbb{R}_{>0}$, and*
2. *is μ-strongly convex on $\bar{\mathcal{B}}(\mathbf{0}, H(\frac{\mu K_1}{2L}))$ 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 (4.21), and $\cup_{i=1}^N \cup_{t=1}^T$*

$\operatorname{argmin} f_i^t \subset \bar{\mathcal{B}}(\mathbf{0}, K_1/2).$

The following theorem is the main result of this chapter.

Theorem 4.2.2. (Sublinear agent's regret bound): *Consider a group of agents $\mathcal{V} = \{1, \cdots, N\}$ over a sequence of* B*-strongly connected graphs, where $T, N \in \mathbb{Z}_{>0}$. Let $\{f_1^t, f_2^t, ..., f_N^t\}_{t=1}^T$ be a sequence of convex cost functions that satisfies Assumption 4.2.1. Suppose that the learning rate is given by $\alpha(t) = \frac{1}{\mu t}$ and that the agents use* (4.1) *to generate the sequence $\{z(t) = (z_1(t), z_2(t), ..., z_N(t))\}_{t=1}^T$. Then for each agent $j \in \mathcal{V}$, we have*

$$\mathsf{R}^j(T) \leq C_1 + C_2(1 + \ln(T)) + C_3(1 + \ln(T))^2, \tag{4.2}$$

where

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

$z^\star$ is defined in (3.2)*, $\delta \in \mathbb{R}_{>0}$ and $\lambda \in \mathbb{R}_{>0}$ depend on the network topology, via* (4.6)

and (4.5)*, respectively, and*

$$L = \sum_{i=1}^{N} L_i, \qquad \text{and}$$
$$\bar{x}(t) = \frac{1}{N}(x_1(t) + x_2(t) + \cdots + x_N(t)) = \frac{1}{N}(\mathbf{1}_N \otimes \mathsf{I}_d)^{\mathsf{T}}[x_1^{\mathsf{T}}(t), x_2^{\mathsf{T}}(t), \cdots, x_N^{\mathsf{T}}(t)]^{\mathsf{T}}. \tag{4.3}$$

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 (4.1) and our result in Theorem 4.2.2 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 [22, 13, 21]. 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.

We recall some technical results in the literature that we will use in the proof of our main result. In Section 4.3, we recall some results on the product of stochastic matrices, which we use to bound $\|z_i(t+1) - \bar{x}(t)\|_2$. In Section 4.4, we prove that under the Assumption 4.2.1, we can bound $\|z_i(t)\|_2$. Finally, in Section 4.5, we prove Theorem 4.2.2.

4.3 Product of Stochastic Matrices

The following fact is well-known and has been established in various places, for example [2, 42, 17, 36, 32, 28].

Proposition 4.3.1. (Product of stochastic matrices on uniformly strongly connected digraphs [28]): *Suppose that the graph sequence $\{\mathcal{G}(t)\}_{t=1}^{T}$ is uniformly strongly connected. Then, the following statements are true:*

1. *For every $s \in \mathbb{Z}_{>0}$, the limit $\lim_{t\to\infty} A^{\mathsf{T}}(t)A^{\mathsf{T}}(t-1)\cdots A^{\mathsf{T}}(s+1)A^{\mathsf{T}}(s)$ exists. In particular, the limiting matrix is a rank-one stochastic matrix, i.e., there is a stochastic vector $\phi(s)$ such that*

$$\lim_{t\to\infty} A^{\mathsf{T}}(t)A^{\mathsf{T}}(t-1)\cdots A^{\mathsf{T}}(s+1)A^{\mathsf{T}}(s) = \mathbf{1}\phi^{\mathsf{T}}(s) \text{ for all } s \geq 0.$$

2. *The convergence rate is geometric, and is given by*

$$\left|[A^{\mathsf{T}}(t)A^{\mathsf{T}}(t-1)\cdots A^{\mathsf{T}}(s+1)A^{\mathsf{T}}(s)]_j^i - \phi_j(s)\right| \leq C\lambda^{t-s} \text{ for all } i,j = 1,\cdots,N,$$

   for some C and $\lambda \in (0,1)$.

The following propositions specifies the values of C and λ. Before stating it, we need a definition. Given a graph sequence $\{\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), A(t))\}_{t=1}^{T}$, let us define

$$\delta := \inf_{t\geq 0}\Big(\min_{1\leq i\leq N}[\mathbf{1}^{\mathsf{T}}A^{\mathsf{T}}(t)\cdots A^{\mathsf{T}}(0)]_i\Big).$$

We now recall the following proposition form [28].

Proposition 4.3.2. ([28]): *We have:*

1. *For any* B*-strongly connected sequence of graphs with* N *vertices, we may choose*

$$C = 2, \qquad \lambda = (1 - \frac{1}{N^{N\mathrm{B}}})^{1/\mathrm{B}}.$$

Moreover, $\delta \geq \frac{1}{N^{N\mathrm{B}}}$.

2. *If in addition every graph* $\mathcal{G}(t)$ *is regular, we have*

$$C = \sqrt{2}, \qquad \lambda = \min\{(1 - \frac{1}{4N^3})^{1/\mathrm{B}}, \max_{t \geq 0} \sigma_2(A(t))\}.$$

Moreover, $\delta = 1$.

Although the proof of this is established in [28], we include a proof for completeness of this manuscript.

Proof. From [2], under the assumption of B-strongly connectivity, and definition of matrix A, it can be verified that if

$$x(t) = A^{\mathsf{T}}(t-1)A^{\mathsf{T}}(t-2)\cdots A^{\mathsf{T}}(s)x(s) \qquad \text{for all } t > s > 0,$$

then

$$\max_{i \in \mathcal{V}} x_i(t) - \min_{i \in \mathcal{V}} x_i(t) \leq (1 - \frac{1}{N^{N\mathrm{B}}})^{\lfloor (t-s)/(N\mathrm{B}) \rfloor} \times \left(\max_{i \in \mathcal{V}} x_i(s) - \min_{i \in \mathcal{V}} x_i(s) \right).$$

This implies that

$$\max_{i \in \mathcal{V}} x_i(t) - \min_{i \in \mathcal{V}} x_i(t) \leq 2(1 - \frac{1}{N^{N\mathrm{B}}})^{(t-s)/(N\mathrm{B})} \times \left(\max_{i \in \mathcal{V}} x_i(s) - \min_{i \in \mathcal{V}} x_i(s) \right). \quad (4.4)$$

We start by proving the statement for C and λ. Since (4.4) holds for every $x(s)$, by choosing $x(s)$ to be each of N basis vectors, we have for every $j = 1, \cdots, N$,

$$\max_{i \in \mathcal{V}} [A^{\mathsf{T}}(t-1) \cdots A^{\mathsf{T}}(s)]_j^i - \min_{i \in \mathcal{V}} [A^{\mathsf{T}}(t-1) \cdots A^{\mathsf{T}}(s)]_j^i \leq 2(1 - \frac{1}{N^{N\mathrm{B}}})^{(t-s)/(N\mathrm{B})}.$$

Since each matrix $A^{\mathsf{T}}(t)$ is row stochastic, the entry $\phi_j(s)$ is a limit of the convex combination of the N numbers $[A^{\mathsf{T}}(t-1) \cdots A^{\mathsf{T}}(s)]_j^i$, $i = 1, \cdots, N$, as $t \to \infty$. Hence, we conclude that

$$\left|[A^{\mathsf{T}}(t)A^{\mathsf{T}}(t-1) \cdots A^{\mathsf{T}}(s+1)A^{\mathsf{T}}(s)]_j^i - \phi_j(s)\right| \leq C\lambda^{t-s},$$

where we may choose $C = 2$ and $\lambda = (1 - \frac{1}{N^{N\mathrm{B}}})^{1/\mathrm{B}}$, which proves part 1.

For the second statement, note that when the graphs $\mathcal{G}(t)$ are regular, the matrices $A(t)$ are doubly stochastic. Then, from the result of [24] we have

$$\|x(t) - \bar{x}(s)\mathbf{1}_N\|_2^2 \leq \left(1 - \frac{1}{2N^3}\right)^{\lfloor (t-s)/\mathrm{B} \rfloor} \|x(s) - \bar{x}(s)\mathbf{1}\|_2^2,$$

where $x(t) = A^{\mathsf{T}}(t-1) \cdots A^{\mathsf{T}}(s)x(s)$ and $\bar{x}(s)$ is the average of the entries of $x(s)$. From the last inequality, we have

$$\|x(t) - \bar{x}(s)\mathbf{1}_N\|_2^2 \leq 2\left(1 - \frac{1}{2N^3}\right)^{(t-s)/\mathrm{B}} \|x(s) - \bar{x}(s)\mathbf{1}\|_2^2,$$

which implies that

$$\max_{i \in \mathcal{V}} |x_i(t) - \bar{x}(s)\mathbf{1}_N| \leq \sqrt{2\left(1 - \frac{1}{2N^3}\right)^{(t-s)/\mathrm{B}}} \|x(s) - \bar{x}(s)\mathbf{1}\|_2.$$

Moreover, since the relation above holds for any vector $x(s) \in \mathbb{R}^n$, by choosing $x(s)$ to be each basis vector, we obtain for each j and all $t > s \geq 0$,

$$\max_{i\in\mathcal{V}}[A^\mathsf{T}(t-1)\cdots A^\mathsf{T}(s)]^i_j - \frac{1}{N} \leq \sqrt{2}\left(\sqrt{1-\frac{1}{2N^3}}\right)^{(t-s)/\mathrm{B}}.$$

Since $\sqrt{1-\beta/2} \leq 1-\beta/4$ for all $\beta \in (0,1)$, it follows that we may choose $C = \sqrt{2}$ and $\lambda = (1-\frac{1}{4N^3})^{1/\mathrm{B}}$. We can similarly show that we may choose $C = 1$ and $\lambda = \max_{t\geq 0}\sigma_2(A(t))$ for regular graphs.

Now, we prove the statements for δ. It is shown in [17, 36] that for every $t > (N-1)\mathrm{B}$, every entry of $A^\mathsf{T}(t)\cdots A^\mathsf{T}(1)$ is positive and has value at least $1/N^{N\mathrm{B}}$. We prove the same statement for $1 \leq t \leq N\mathrm{B}$. By the definition of matrices $A(t)$, we have $A(t)^i_i = 1/d_i(t)$, since every agent has a self-loop. Note that $d_i(t) \leq N$, which implies that $A(t)^i_i \geq 1/N$ for all t and i. Hence, for all $i \in \mathcal{V}$ and all $t \in \mathbb{Z}_{>0}$

$$[A^\mathsf{T}(t+1)A^\mathsf{T}(t)\cdots A^\mathsf{T}(0)]^i_i \geq \frac{1}{N}[A^\mathsf{T}(t)\cdots A^\mathsf{T}(0)]^i_i.$$

Therefore, we have $[\mathbf{1}^\mathsf{T}A^\mathsf{T}(t)\cdots A^\mathsf{T}(1)]_i \geq 1/N^{N\mathrm{B}}$ for all $i \in \mathcal{V}$ and $1 \leq t \leq N\mathrm{B}$. This proves the bound $\delta \geq 1/N^{N\mathrm{B}}$. It can be verified that for regular matrices we have $\delta = 1$. □

We recall the following corollary of Proposition 4.3.2, without stating the proof [28].

Corollary 4.3.3. *If the graphs $\mathcal{G}(t)$ are* B*-strongly connected, the following hold:*

1. *There exists a sequence of stochastic vectors $\phi(t)$, such that for all $i, j \in \mathcal{V}$, and*

all $t > s \geq 0$, *we have*

$$\left|[A^{\mathsf{T}}(t)A^{\mathsf{T}}(t-1)\cdots A^{\mathsf{T}}(s+1)A^{\mathsf{T}}(s)]_j^i - \phi_j(s)\right| \leq C\lambda^{t-s},$$

where we can always choose

$$C = 4, \qquad \lambda = (1 - 1/N^{N\mathrm{B}})^{1/\mathrm{B}}, \tag{4.5}$$

Moreover, if $\mathcal{G}(t)$ *are regular, we may choose*

$$C = 2\sqrt{2}, \qquad \lambda = \min\{(1 - \frac{1}{4N^3})^{1/\mathrm{B}}, \max_{t\geq 0}\ \sigma_2(A(t))\}.$$

2. *The parameter*

$$\delta := \inf_{t\geq 0}\Big(\min_{1\leq i\leq N}[A(t)\cdots A(0)\mathbf{1}]_i\Big).$$

satisfies

$$\delta \geq \frac{1}{N^{N\mathrm{B}}}. \tag{4.6}$$

In addition, if $\mathcal{G}(t)$ *are regular, we have* $\delta = 1$.

3. *The stochastic vectors* $\phi(t)$, *for all* $j \in \mathcal{V}$ *and* $t \in \mathbb{Z}_{>0}$ *satisfies*

$$\phi_j(t) \geq \frac{\delta}{N}.$$

The proof follows by taking transpose of matrices $A(t)$ and definition of δ; see [28]. We also recall the following result from [23].

Lemma 4.3.4. ([23, Corollary 1]): *Consider the sequences* $\{z_i(t)\}_{t=1}^{T}$, *for all* $i \in \mathcal{V}$, *generated by* (4.1) *on a sequence of* B*-strongly-connected digraphs. Then we have*

$$\|z_i(t+1) - \bar{x}(t)\|_2 \le \frac{8}{\delta}\Big(\lambda^t \sum_{i=1}^{N} \|x_i(0)\|_1 + \sum_{s=1}^{t} \lambda^{t-s} \sum_{i=1}^{N} \|\alpha(s) g_i(s)\|_1\Big), \tag{4.7}$$

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

$$\delta \ge \frac{1}{N^{N\mathrm{B}}} \qquad \text{and} \qquad \lambda \le (1 - \frac{1}{N^{N\mathrm{B}}})^{1/(N\mathrm{B})}.$$

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

$$\lambda \le \min\Big\{(1 - \frac{1}{4N^3})^{1/(\mathrm{B})}, \max_{t \in \{1,\cdots,T\}} \sigma_2(A(t))\Big\}.$$

The constant δ measures the imbalance of the network and λ is a measure of connectivity, see [28] for more details. Although the proof of the lemma is given in [28], we state it for completeness.

Proof. First, we assume that $z_i(t) \in \mathbb{R}$ for all $i \in \mathcal{V}$ and $t \in \{1, \cdots, T\}$. For all $t > s \ge 0$, we define $A(t:s)$ as

$$A(t:s) = A(t)A(t-1)\cdots A(s),$$

and $A(t:t) = A(t)$. We also let $\epsilon(t) = -\alpha(t)g(t)$. From (4.1) and for $t \ge 0$, we have

$$x(t+1) = A(t:0)x(0) + \sum_{s=1}^{t} A(t:s)\epsilon(s) + \epsilon(t+1),$$

which implies that

$$A(t+1)x(t+1) = A(t+1:0)x(0) + \sum_{s=1}^{t+1} A(t+1:s)\epsilon(s). \tag{4.8}$$

Since each $A(t)$ is column stochastic, we have that $\mathbf{1}^\intercal A(t) = \mathbf{1}^\intercal$. Using this and (4.8), we have

$$\mathbf{1}^\intercal x(t+1) = \mathbf{1}^\intercal x(0) + \sum_{s=1}^{t+1} \mathbf{1}^\intercal \epsilon(s). \tag{4.9}$$

Using (4.8) and (4.9), we have

$$\begin{aligned} A(t+1)x(t+1) - \phi(t+1)\mathbf{1}^\intercal x(t+1) =& (A(t+1:0) - \phi(t+1)\mathbf{1}^\intercal)x(0) \\ &+ \sum_{s=1}^{t+1} (A(t+1:s) - \phi(t+1)\mathbf{1}^\intercal)\epsilon(s), \end{aligned}$$

where $\phi(t) \in \mathbb{R}^N$ is given in Proposition 4.3.1. We define $D(t:s) = A(t:s) - \phi(t)\mathbf{1}^\intercal$. Using Corollary 4.3.3, for all $i, j \in \mathcal{V}$ and $t \geq s \geq 0$ we have

$$|[D(t:s)]_j^i| \leq C\lambda^{t-s}, \tag{4.10}$$

where C and λ is given in Corollary 4.3.3. Hence, we have

$$A(t+1)x(t+1) = \phi(t+1)\mathbf{1}^\intercal x(t+1) + D(t+1:0)x(0) + \sum_{s=1}^{t+1} D(t+1:s)\epsilon(s). \tag{4.11}$$

We also have

$$w(t+1) = A(t)x(t) = \phi(t)\mathbf{1}^\intercal x(t) + D(t:0)x(0) + \sum_{s=1}^{t} D(t:s)\epsilon(s). \tag{4.12}$$

We can derive a similar expression for $y(t+1)$:

$$y(t+1) = A(t:0)y(0) = \phi(t)\mathbf{1}^{\mathsf{T}}y(0) + D(t:0)y(0) = \phi(t)N + D(t:0)\mathbf{1}. \quad (4.13)$$

From (4.12) and (4.13), we have

$$z_i(t+1) = \frac{w_i(t+1)}{y_i(t+1)} = \frac{\phi_i(t)\mathbf{1}^{\mathsf{T}}x(t) + [D(t:0)x(0)]_i + \sum_{s=1}^{t}[D(t:s)\epsilon(s)]_i}{\phi_i(t)N + [D(t:0)\mathbf{1}]_i}.$$

Hence,

$$z_i(t+1) - \frac{\mathbf{1}^{\mathsf{T}}x(t)}{N} = \frac{\phi_i(t)\mathbf{1}^{\mathsf{T}}x(t) + [D(t:0)x(0)]_i + \sum_{s=1}^{t}[D(t:s)\epsilon(s)]_i}{\phi_i(t)N + [D(t:0)\mathbf{1}]_i} - \frac{\mathbf{1}^{\mathsf{T}}x(t)}{N},$$

and as a result,

$$z_i(t+1) - \frac{\mathbf{1}^{\mathsf{T}}x(t)}{N} = \frac{N[D(t:0)x(0)]_i + N\sum_{s=1}^{t}[D(t:s)\epsilon(s)]_i - \mathbf{1}^{\mathsf{T}}x(t)[D(t:0)\mathbf{1}]_i}{N(\phi_i(t)N + [D(t:0)\mathbf{1}]_i}),$$

The denominator of the above fraction is N times the ith row of $A(t:0)$. By definition of δ, this row sum is at least δ, and consequently

$$\phi_i(t)N + [D(t:0)\mathbf{1}]_i = [A(t:0)\mathbf{1}]_i \geq \delta.$$

Therefore, for all i and $t \geq 1$,

$$
\begin{aligned}
|z_i(t+1) - \bar{x}(t)| \leq & \frac{|[D(t:0)x(0)]_i| + \sum_{s=1}^{t} |[D(t:s)\epsilon(s)]_i|}{\phi_i(t)N + [D(t:0)\mathbf{1}]_i} + \frac{|\mathbf{1}^\mathsf{T} x(t)[D(t:0)\mathbf{1}]_i)|}{N(\phi_i(t)N + [D(t:0)\mathbf{1}]_i)} \\
\leq & \frac{1}{\delta}\left(\max_j |[D(t:0)^i_j]| \|x(0)\|_1 + \sum_{s=1}^{t} (\max_j |[D(t:s)^i_j]|) \|\epsilon(s)\|_1 \right) \\
& + \frac{1}{N\delta} |\mathbf{1}^\mathsf{T} x(t)| (\max_j |[D(t:0)^i_j]|) N,
\end{aligned}
$$

where, we have used the triangle and Hölder inequality in the first and second inequality, respectively. Using (4.10), we obtain

$$
|z_i(t+1) - \bar{x}(t)| \leq \frac{C}{\delta}\lambda^t \|x(0)\|_1 + \frac{C}{\delta}\left(\sum_{s=1}^{t} \lambda^{t-s} \|\epsilon(s)\|_1 + |\mathbf{1}^\mathsf{T} x(t)| \lambda^t \right). \tag{4.14}
$$

Using (4.9), we have

$$
|\mathbf{1}^\mathsf{T} x(t)| \leq \|x(0)\|_1 + \sum_{s=1}^{t} \|\epsilon(s)\|_1.
$$

Using this and (4.14), we obtain

$$
\begin{aligned}
|z_i(t+1) - \bar{x}(t)| &\leq \frac{C}{\delta}\lambda^t \|x(0)\|_1 + \frac{C}{\delta} \sum_{s=1}^{t} \lambda^{t-s} \|\epsilon(s)\|_1 + \frac{C}{\delta}\lambda^t \left(\|x(0)\|_1 + \sum_{s=1}^{t} \|\epsilon(s)\|_1 \right) \\
&\leq \frac{C}{\delta}\left(2\lambda^t \|x(0)\|_1 + 2\sum_{s=1}^{t} \lambda^{t-s} \|\epsilon(s)\|_1 \right).
\end{aligned}
$$

Since we can choose $C \leq 4$, we obtain the result. Note that in the last inequality, $\|x(0)\|_1 = \sum_{i=1}^{N} |x_i(0)|$ and $\|\epsilon(s)\|_1 = \sum_{i=1}^{N} |\alpha(s) g_i(s)|$. For $z_i(t) \in \mathbb{R}^d$, we can apply the same argument of the preceding proof in every coordinate of $z_i(t)$ and using the fact that the Euclidean norm of a vector is less than the 1-norm of it, to obtain the result. $\square$

We state a corollary of Lemma 4.3.4, which plays a key role in the proof of our main result.

Corollary 4.3.5. *Under the assumption of Theorem 4.2.2, where the learning rate is chosen as $\alpha(t) = \frac{1}{\mu t}$, we have*

$$\|z_i(t) - \bar{x}(t-1)\|_2 \leq \frac{8}{\delta}\left(\sum_{i=1}^{N} \|x_i(0)\|_1 + \frac{L}{\mu(1-\lambda)}\right), \tag{4.15}$$

$$\sum_{t=1}^{T}\sum_{i=1}^{N} L_i\|z_i(t) - \bar{x}(t-1)\|_2 \leq \frac{8L}{\delta(1-\lambda)}\Big(\sum_{i=1}^{N} \|x_i(0)\|_1 + \frac{L}{\mu}(1+\ln(T))\Big). \tag{4.16}$$

The proof follows immediately from the fact that $\lambda \in (0,1)$ and

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

4.4 Results on Locally Strongly Convex Functions

Proof of Theorem 4.2.2 also relies on some results on *locally* strongly convex functions. In this section, we study the boundedness of the agents' states, where agents use (4.1) 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 $\{f_1^t, \cdots, f_N^T\}_{t=1}^{T}$ satisfies Assumption 4.2.1. We recall the following lemmas from [21], without stating their proof.

Lemma 4.4.1. (Convex Cone Inclusion [21, Lemma 5.6]): *Given $\beta \in (0,1]$, $\epsilon \in (0,\beta)$, and any scalars $K_1, K_2 \in \mathbb{R}_{>0}$, let*

$$\hat{r}_\beta := \frac{K_1 + K_2}{\beta\sqrt{1-\epsilon^2} - \epsilon\sqrt{1-\beta^2}}.$$

Then, $\hat{r}_\beta \in (K_1 + K_2, +\infty)$ *and, for any* $x \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, \hat{r}_\beta)$,

$$\bigcup_{w \in \bar{\mathcal{B}}(-x, K_1+K_2)} \mathcal{H}_\beta(w) \subset \mathcal{H}_\epsilon(-x),$$

where $\mathcal{H}_\beta(w)$ *is a convex cone defined in* (2.1) *and the set on the left is convex.*

The proof is based on trigonometric functions.

Lemma 4.4.2. ([21, Lemma 5.9]): *Let* $f : \mathbb{R}^d \to \mathbb{R}$ *be a convex function on* $\mathbb{R}^d$ *that is also* γ*-strongly convex on* $\bar{\mathcal{B}}(y, \zeta)$ *for some* $\gamma, \zeta \in \mathbb{R}_{>0}$ *and* $y \in \mathbb{R}^d$*. Then, for any* $x \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(y, \zeta)$ *and* $g_x \in \partial f(x), g_y \in \partial f(y)$,

$$(g_x - g_y)^\mathsf{T}(x - y) \geq \gamma\zeta\|x - y\|_2. \tag{4.17}$$

If in addition, f *has* L*-bounded subgradient sets and* $\mathbf{0} \in \partial f(y)$*, then* f *is* $\frac{\gamma\zeta}{L}$*-central in* $\mathbb{R}^d \setminus \bar{\mathcal{B}}(y, \zeta)$*. (Note that if* $\mathbf{0} \in \partial f(y)$*, then* $\operatorname{argmin}_{x \in \mathbb{R}^d} f(x) = \{y\}$ *is a singleton by strong convexity in the ball* $\bar{\mathcal{B}}(y, \zeta)$*.)*

The proof is given in [21]. We reproduce it for completeness of this manuscript.

Proof. For the given $y \in \mathbb{R}^d$ and for all $x \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(y, \zeta)$, we choose $\tilde{x}$ on the line segment between x and y such that $\|\tilde{x} - y\|_2 = \zeta$, i.e., $\tilde{x}$ is on the boundary of $\bar{\mathcal{B}}(y, \zeta)$. Hence, we have

$$x - y = \frac{1}{v}(\tilde{x} - y) = \frac{1}{1 - v}(x - \tilde{x}), \tag{4.18}$$

for some $v \in (0, 1)$. Then, for any $g_x \in \partial f(x)$, $g_y \in \partial(y)$, and $g_{\tilde{x}} \in \partial f(\tilde{x})$, we can write

$$
\begin{aligned}
(g_x - g_y)^\top (x-y) =& (g_x - g_{\tilde{x}} + g_{\tilde{x}} - g_y)^\top (x-y) \\
=& (g_x - g_{\tilde{x}})^\top (x-y) + (g_{\tilde{x}} - g_y)^\top (x-y) \\
=& \frac{1}{1-v}(g_x - g_{\tilde{x}})^\top (x-\tilde{x}) + \frac{1}{v}(g_{\tilde{x}} - g_y)^\top (\tilde{x}-y) \\
\geq & 0 + \frac{\gamma}{v}\|\tilde{x}-y\|_2^2 = \gamma\|\tilde{x}-y\|_2\|x-y\|_2 = \gamma\zeta\|x-y\|_2,
\end{aligned}
$$

where in the third equality we have used (4.18) and in the last inequality we have used convexity for the first term and strong convexity for the second term. For the second part, note that we have $g_y = \mathbf{0}$. Using this in (4.17) and multiplying it with $\frac{\|g_x\|_2}{H}$, we conclude the second part. □

We now state the following result.

Lemma 4.4.3. *For $T \in \mathbb{R}_{>0}$, let $\{f_1^t, \cdots, f_N^T\}_{t=1}^T$ 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*

$$
\bigcup_{i=1}^{N}\bigcup_{t=1}^{T} \operatorname{argmin} f_i^t \subset \bar{\mathcal{B}}(\mathbf{0}, K_1),
$$

for some $K_1 \in \mathbb{R}_{>0}$ independent of T, and assume $\{f_1^t, \cdots, f_N^t\}_{t=1}^T$ are β-central on $\mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, K_1)$, where $\beta \in (0, 1]$. Then, for any sequence $\{(z_1(t), \cdots, z_N(t))\}_{t=1}^T$ and $\{\bar{x}(t)\}_{t=1}^T$ generated by (4.1) over a sequence of B*-strongly connected graphs, and any sequence of learning rates $\{\alpha(t)\}_{t=1}^T$, we have*

$$
\|\bar{x}(t)\|_2 \leq r_\beta + \frac{L}{N}\max_{s\geq 1}\alpha(s) + \|\bar{x}(0)\|_2, \tag{4.19}
$$

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

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

$$r_\beta = \max\Big\{\frac{K_1 + K_2}{\beta\sqrt{1-\epsilon^2} - \epsilon\sqrt{1-\beta^2}}, \frac{L}{2N\epsilon}\max_{s\geq 1}\alpha(s)\Big\}, \tag{4.20}$$

$$H(\beta) = r_\beta + \|\bar{x}(0)\|_2 + K_2 + \frac{L}{N}\max_{s\geq 1}\alpha(s), \tag{4.21}$$

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

A very similar result has been proved in [21]. We have modified this result to be fit with the proposed algorithm given in (4.1). The proof is given next and follows the same lines as [21, Lemma 5.7]

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 (4.19), and from (4.1) and the column stochasticity of matrices $A(t)$, we can write

$$\bar{x}(t) = \bar{x}(t-1) - \frac{\alpha(t)}{N}\sum_{i=1}^{N} g_i(t). \tag{4.22}$$

Here, we conclude that if $\bar{x}(t-1) \in \bar{\mathcal{B}}(\mathbf{0}, r_\beta)$, then

$$\bar{x}(t) \in \bar{\mathcal{B}}\Big(\mathbf{0}, r_\beta + L\max_{s\geq 1}\alpha(s)/N\Big).$$

By an argument very similar to the one in the proof [21, Lemma 5.7], we have that if $\bar{x}(t-1) \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, r_\beta)$, then $\|\bar{x}(t)\|_2 \leq \|\bar{x}(t-1)\|_2$. To establish this, we study the direction and the magnitude of the term $-\frac{\alpha(t)}{N}\sum_{i=1}^{N} g_i(t)$ in (4.22).

First, we study the direction of $-\frac{\alpha(t)}{N}\sum_{i=1}^{N} g_i(t)$. Recall (4.15), that for each $i \in \mathcal{V}$, we have

$$\|z_i(t) - \bar{x}(t-1)\|_2 \leq \frac{8}{\delta}\left(\sum_{i=1}^{N} \|x_i(0)\|_1 + \frac{L}{\mu(1-\lambda)}\right) \leq K_2. \tag{4.23}$$

Using this, since $\bar{x}(t-1) \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, r_\beta)$ and $r_\beta > K_1 + K_2$ from Lemma 4.4.1, we conclude that $z_i(t) \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, K_1)$ for all $i \in \mathcal{V}$. The β-centrality of the function f_i^t, for each $i \in \mathcal{V}$, and $t \in \{1, \cdots, T\}$ on $\mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, K_1)$, implies that, for each $z \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, K_1)$, we have

$$-\partial f_i^t(z) \subseteq \bigcup_{y \in \operatorname{argmin}(f_i^t)} \mathcal{H}_\beta(y - z) \subseteq \bigcup_{y \in \bar{\mathcal{B}}(\mathbf{0}, K_1)} \mathcal{H}_\beta(y - z), \tag{4.24}$$

where the last inclusion follows from the hypothesis that $\sum_{t=1}^{T} \sum_{i=1}^{N} \operatorname{argmin}(f_i^t) \subseteq \bar{\mathcal{B}}(\mathbf{0}, K_1)$. Now, using the change of variable $w = y - z$, we obtain

$$\bigcup_{y \in \bar{\mathcal{B}}(\mathbf{0}, K_1), z \in \bar{\mathcal{B}}(x, K_2)} \mathcal{H}_\beta(y - z) = \bigcup_{w \in \bar{\mathcal{B}}(-x, K_1 + K_2)} \mathcal{H}_\beta(w). \tag{4.25}$$

From Lemma 4.4.1, we conclude that the right side of (4.25) is convex whenever $x \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, r_\beta)$. Hence, taking the union when $z \in \bar{\mathcal{B}}(x, K_2)$ on both sides of (4.24) and using (4.25), we have

$$\operatorname{conv}\left(\bigcup_{z \in \bar{\mathcal{B}}(x, K_2)} -\partial f_i^t(z)\right) \subseteq \bigcup_{w \in \bar{\mathcal{B}}(-x, K_1 + K_2)} \mathcal{H}_\beta(w) \subseteq \mathcal{H}_\epsilon(-x), \tag{4.26}$$

where the last inclusion holds for any $x \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, K_1)$ by Lemma 4.4.1. Taking $x = \bar{x}(t-1)$ and noting that for all $i \in \mathcal{V}$, $z_i(t) \in \bar{\mathcal{B}}(\bar{x}(t-1), K_2)$ from (4.23), we have

$$-g_i(t) \in \operatorname{conv}\left(\bigcup_{z\in\bar{\mathcal{B}}(\bar{x}(t-1),K_2)} -\partial f_i^t(z)\right) \subseteq \mathcal{H}_\epsilon(-\bar{x}(t))$$

The convexity of $\mathcal{H}_\epsilon(-\bar{x}(t))$ implies that

$$-\frac{1}{N}\sum_{i=1}^{N} g_i(t) \in \mathcal{H}_\epsilon(-\bar{x}(t)). \tag{4.27}$$

From the definition of convex cone (2.1), we have

$$\big(\frac{1}{N}\sum_{i=1}^{N} g_i(t)\big)^\intercal \bar{x}(t-1) \geq \epsilon\|\frac{1}{N}\sum_{i=1}^{N} g_i(t)\|_2\|\bar{x}(t-1)\|_2. \tag{4.28}$$

Now we study the magnitude of $\|\bar{x}(t)\|_2$. From (4.22), we have

$$\begin{aligned}\|\bar{x}(t)\|_2^2 &= \|\bar{x}(t-1) - \frac{\alpha(t)}{N}\sum_{i=1}^{N} g_i(t)\|_2^2,\\ &= \|\bar{x}(t-1)\|_2^2 + \frac{(\alpha(t))^2}{N^2}\|\sum_{i=1}^{N} g_i(t)\|_2^2 - \frac{2\alpha(t)}{N}\big(\sum_{i=1}^{N} g_i(t)\big)^\intercal \bar{x}(t-1). \end{aligned}\tag{4.29}$$

Using (4.28) in (4.29) we have

$$\|\bar{x}(t)\|_2^2 \leq \|\bar{x}(t-1)\|_2^2 + \frac{(\alpha(t))^2}{N^2}\|\sum_{i=1}^{N} g_i(t)\|_2^2 - \frac{2\epsilon\alpha(t)}{N}\|\sum_{i=1}^{N} g_i(t)\|_2\|x(t-1)\|_2. \tag{4.30}$$

From definition of r_β in (4.20), and since $\bar{x}(t-1) \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, r_\beta)$, we can write

$$\|\bar{x}(t-1)\|_2 \geq \frac{L}{2N\epsilon}\max_{s\geq 1}\alpha(s) \geq \frac{\alpha(t)}{2N\epsilon}\|\sum_{i=1}^{N} g_i(t)\|_2,$$

where we have used $\|\sum_{i=1}^{N} g_i(t)\|_2 \leq L$ in the last inequality. Using this in (4.30), we conclude that

$$\|\bar{x}(t)\|_2 \leq \|\bar{x}(t-1)\|_2. \tag{4.31}$$

This guarantees that if the starting assumption that $\bar{x}(t-1) \in \mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, r_\beta)$ holds, then $\|\bar{x}(t)\|_2 \leq \|\bar{x}(t-1)\|_2$.

Next, using (4.23), we conclude that $\|z_i(t)\|_2 \leq H(\beta)$, for all $t \in \mathbb{Z}_{>0}$, where $H(\beta)$ is given in (4.21). □

Theorem 4.4.4. *For $T \in \mathbb{R}_{>0}$, let $\{f_1^t, \cdots, f_N^T\}_{t=1}^T$ 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*

$$\bigcup_{i=1}^{N}\bigcup_{t=1}^{T} \operatorname{argmin} f_i^t \subset \bar{\mathcal{B}}(\mathbf{0}, K_1/2),$$

for some $K_1 \in \mathbb{R}_{>0}$ independent of T. Suppose that $\{f_1^t, \cdots, f_N^t\}_{t=1}^T$ is a sequence of μ-strongly convex functions on $\bar{\mathcal{B}}(\mathbf{0}, H(\frac{\mu K_1}{2L}))$, for some $\mu \in \mathbb{R}_{>0}$, where $H(\cdot)$ is defined in (4.21). *Then $\{z_i(t)\}_{i=1}^N$, generated by* (4.1) *over a sequence of* B*-strongly connected graphs, stays in $\bar{\mathcal{B}}(\mathbf{0}, H(\frac{\mu K_1}{2L}))$, for all t.*

Proof. By an argument very similar to the one in the proof [21, Theorem 6.1], $K_1 < r_\beta < H(\frac{\mu K_1}{2L})$ and hence $K_1 < H(\frac{\mu K_1}{2L})$. Thus, f_i^t is μ-strongly convex in $\bar{\mathcal{B}}(\mathbf{0}, K_1)$ and an application of Lemma 4.4.2 implies that each f_i^t is β'-central on $\mathbb{R}^d \setminus \bar{\mathcal{B}}(\mathbf{0}, K_1)$, where $\beta' \leq \frac{\mu K_1}{2L}$. Hence, the assumptions of Lemma 4.4.3 are satisfied with $\beta = \frac{\mu K_1}{2L}$ and as a result, $z_i(t)$ remains in the region $\bar{\mathcal{B}}(\mathbf{0}, H(\frac{\mu K_1}{2L}))$. □

4.5 Proof of the Main Result

We now start the process of proving Theorem 4.2.2. In order to prove this result, we need to first present a result on *network regret*, which is defined in (3.3). We have the following theorem.

Theorem 4.5.1. (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* $\{f_1^t, f_2^t, ..., f_N^t\}_{t=1}^T$ *be a sequence of convex cost functions that satisfies Assumption 4.2.1. Then the sequence* $\{z(t) = (z_1(t), z_2(t), ..., z_N(t))\}_{t=1}^T$ *generated by* (4.1) *with the learning rate* $\alpha(t) = \frac{1}{\mu t}$ *satisfies the network regret bound*

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

where

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

$z^\star$ *is defined in* (3.2), $\delta \in \mathbb{R}_{>0}$ *and* $\lambda \in \mathbb{R}_{>0}$ *depend on the network topology as before, and* L *and* $\bar{x}(0)$ *are given by* (4.3).

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 4.2.2.

Lemma 4.5.2. *Let $\{f_1^t, f_2^t, ..., f_N^t\}_{t=1}^T$ be a sequence of convex cost functions that satisfies Assumption 4.2.1. Then the sequence $\{z(t)\}_{t=1}^T$ generated by (4.1), with the learning rate $\alpha(t)$, over a sequence of* B*-strongly connected graphs, satisfies*

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

where L and $\bar{x}(t)$ are given by (4.3).

Proof. Using Theorem 4.4.4 and Assumption 4.2.1, for any given initial condition and any agent $i \in \mathcal{V}$, we have that $z_i(t)$ stays in $\bar{\mathcal{B}}(\mathbf{0}, H(\frac{\mu K_1}{2L}))$ for all $t \in \{1, 2, \cdots, T\}$, where the modulus of strong convexity of f is μ. Using (3.3) and since $\{\{f_i^t\}_{i=1}^N\}_{t=1}^T$ is a sequence of μ-strongly convex functions, we have that

$$\begin{aligned}
\mathsf{R}(T) &= \sum_{t=1}^{T}\sum_{i=1}^{N}(f_i^t(z_i(t)) - f_i^t(z^\star)) \\
&\leq \sum_{t=1}^{T}\sum_{i=1}^{N}(g_i(t)^\mathsf{T}(z_i(t) - z^\star) - \frac{\mu}{2}\|z_i(t) - z^\star\|_2^2).
\end{aligned}$$

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

$$\begin{aligned}\mathsf{R}(T) \leq & \sum_{t=1}^{T}\sum_{i=1}^{N}(g_i(t)^{\mathsf{T}}(z_i(t)-\bar{x}(t-1)+\bar{x}(t-1)-z^{\star}) \\ & \qquad - \frac{\mu}{2}\|z_i(t)-\bar{x}(t-1)+\bar{x}(t-1)-z^{\star}\|_2^2), \\ = & \sum_{t=1}^{T}\sum_{i=1}^{N}\Big(g_i(t)^{\mathsf{T}}(z_i(t)-\bar{x}(t-1))+g_i(t)^{\mathsf{T}}(\bar{x}(t-1)-z^{\star}) \\ & \qquad - \frac{\mu}{2}\big(\|z_i(t)-\bar{x}(t-1)\|_2^2+\|\bar{x}(t-1)-z^{\star}\|_2^2 \\ & \qquad\quad + 2(z_i(t)-\bar{x}(t-1))^{\mathsf{T}}(\bar{x}(t-1)-z^{\star})\big)\Big) \end{aligned} \tag{4.32}$$

Using (4.1), we have

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

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

$$\bar{x}(t) = \bar{x}(t-1) - \frac{\alpha(t)}{N}\sum_{i=1}^{N} g_i(t), \tag{4.33}$$

where $\bar{x}(t)$ is given by (4.3). Subtracting $z^{\star}$ and taking the norm square, we get

$$\begin{aligned}\|\bar{x}(t)-z^{\star}\|_2^2 = & \|\bar{x}(t-1)-z^{\star}\|_2^2 + \frac{\alpha^2(t)}{N^2}\Big\|\sum_{i=1}^{N} g_i(t)\Big\|_2^2 \\ & - \frac{2\alpha(t)}{N}\Big[\sum_{i=1}^{N} g_i(t)\Big]^{\mathsf{T}}(\bar{x}(t-1)-z^{\star}).\end{aligned}$$

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

$$\Big[\sum_{i=1}^{N} g_i(t)\Big]^{\mathsf{T}}(\bar{x}(t-1)-z^\star) \leq \frac{N}{2\alpha(t)}(\|\bar{x}(t-1)-z^\star\|_2^2 - \|\bar{x}(t)-z^\star\|_2^2) + \frac{\alpha(t)}{2N}L^2,$$

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

$$\begin{aligned}
&\sum_{t=1}^{T}\Big[\sum_{i=1}^{N} g_i(t)\Big]^{\mathsf{T}}(\bar{x}(t-1)-z^\star) \\
&\qquad \leq \sum_{t=1}^{T}\frac{N}{2\alpha(t)}(\|\bar{x}(t-1)-z^\star\|_2^2 - \|\bar{x}(t)-z^\star\|_2^2) + \frac{L^2}{2N}\sum_{t=1}^{T}\alpha(t) \\
&\qquad \leq \frac{N}{2\alpha(1)}\|\bar{x}(0)-z^\star\|_2^2 + \frac{N}{2}\sum_{t=1}^{T-1}\|\bar{x}(t)-z^\star\|_2^2\Big(\frac{1}{\alpha(t+1)} - \frac{1}{\alpha(t)}\Big) + \frac{L^2}{2N}\sum_{t=1}^{T}\alpha(t).
\end{aligned} \tag{4.34}$$

The proof then follows immediately using (4.32) and (4.34), along with the fact that $g_i(\cdot)$ is L_i-bounded over $\mathbb{R}^d$. □

The final stepping stone in the proof of Theorem 4.5.1 is stated next.

Lemma 4.5.3. *Under the assumption of Theorem 4.5.1, where the learning rate is chosen as $\alpha(t) = \frac{1}{\mu t}$, we have*

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

Proof. Using the Cauchy-Schwarz inequality, we have

$$
\begin{aligned}
-\sum_{t=1}^{T}\sum_{i=1}^{N}&\mu(z_i(t)-\bar{x}(t-1))^{\mathsf{T}}(\bar{x}(t-1)-z^\star)\\
&\le \sum_{t=1}^{T}\sum_{i=1}^{N}\mu\|z_i(t)-\bar{x}(t-1)\|_2\|\bar{x}(t-1)-z^\star\|_2.
\end{aligned}
$$

Let

$$
X=\sum_{t=1}^{T}\sum_{i=1}^{N}\mu\|z_i(t)-\bar{x}(t-1)\|_2\|\bar{x}(t-1)-z^\star\|_2.
$$

From equation (4.33), we can write

$$
\|\bar{x}(t-1)-z^\star\|_2\le\|\bar{x}(0)-z^\star\|_2+\Big\|\sum_{s=1}^{t-1}\frac{\alpha(s)}{N}\sum_{i=1}^{N}g_i(s)\Big\|_2. \tag{4.35}
$$

Using (4.7) and (4.35), we can write

$$
\begin{aligned}
X\le&\sum_{t=1}^{T}\sum_{i=1}^{N}\mu\frac{8}{\delta}\left(\lambda^{t-1}\sum_{j=1}^{N}\|x_j(0)\|_1+\sum_{s=1}^{t-1}\lambda^{t-1-s}\sum_{j=1}^{N}\|\alpha(s)g_j(s)\|_1\right)\\
&\qquad\times\left(\|\bar{x}(0)-z^\star\|_2+\|\sum_{s=1}^{t-1}\frac{\alpha(s)}{N}\sum_{j=1}^{N}g_j(s)\|_2\right)\\
\le&\sum_{t=1}^{T}\sum_{i=1}^{N}\mu\frac{8}{\delta}\Bigg(\lambda^{t-1}\sum_{j=1}^{N}\|x_j(0)\|_1\|\bar{x}(0)-z^\star\|_2\\
&\qquad+\lambda^{t-1}\sum_{j=1}^{N}\|x_j(0)\|_1\Big(\sum_{s=1}^{t-1}\frac{\alpha(s)}{N}L\Big)\\
&\qquad+L\|\bar{x}(0)-z^\star\|_2\sum_{s=1}^{t-1}\lambda^{t-1-s}\alpha(s)\\
&\qquad+\frac{L^2}{N}\sum_{s=1}^{t-1}\lambda^{t-1-s}\alpha(s)\sum_{s=1}^{t-1}\alpha(s)\Bigg).
\end{aligned} \tag{4.36}
$$

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

$$
\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{N}\mu\frac{8}{\delta}\lambda^{t-1}\sum_{j=1}^{N}\|x_j(0)\|_1\|\bar{x}(0)-z^\star\|_2 \\
&\qquad = \mu\frac{8N}{\delta}\sum_{j=1}^{N}\|x_j(0)\|_1\|\bar{x}(0)-z^\star\|_2\sum_{t=1}^{T}\lambda^{t-1} \\
&\qquad \leq \mu\frac{8N}{\delta(1-\lambda)}\sum_{j=1}^{N}\|x_j(0)\|_1\|\bar{x}(0)-z^\star\|_2, \qquad (4.37)
\end{aligned}
$$

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

$$
\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{N}\mu\frac{8}{\delta N}\sum_{j=1}^{N}\|x_j(0)\|_1 L\lambda^{t-1}\sum_{s=1}^{t-1}\frac{1}{\mu s} \\
&\qquad = \frac{8}{\delta}\sum_{j=1}^{N}\|x_j(0)\|_1 L\sum_{t=1}^{T}\lambda^{t-1}\sum_{s=1}^{t-1}\frac{1}{s} \\
&\qquad \leq \frac{8}{\delta}\sum_{j=1}^{N}\|x_j(0)\|_1 L\sum_{t=1}^{T}\lambda^{t-1}(1+\ln(t)) \\
&\qquad \leq \frac{8}{\delta(1-\lambda)}\sum_{j=1}^{N}\|x_j(0)\|_1 L(1+\ln(T)), \qquad (4.38)
\end{aligned}
$$

where we have 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

$$
\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{N}\mu\frac{8}{\delta}L\|\bar{x}(0)-z^\star\|_2\sum_{s=1}^{t-1}\lambda^{t-1-s}\frac{1}{\mu s} \\
&\qquad = N\frac{8}{\delta}L\|\bar{x}(0)-z^\star\|_2\sum_{t=1}^{T}\sum_{s=1}^{t-1}\frac{\lambda^{t-1-s}}{s} \\
&\qquad \leq N\frac{8}{\delta}L\|\bar{x}(0)-z^\star\|_2\frac{1+\ln(T)}{1-\lambda}. \qquad (4.39)
\end{aligned}
$$

Finally,

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{N}\mu\frac{8}{\delta}\frac{L^2}{N}\sum_{s=1}^{t-1}\Big(\frac{\lambda^{t-1-s}}{\mu s}\Big)\sum_{s=1}^{t-1}\frac{1}{\mu s} \\
&\quad\le \frac{8}{\delta}\frac{L^2}{\mu}\sum_{t=1}^{T}(1+\ln(t))\sum_{s=1}^{t-1}\frac{\lambda^{t-1-s}}{s} \\
&\quad\le \frac{8}{\delta(1-\lambda)}\frac{L^2}{\mu}(1+\ln(T))^2. \qquad (4.40)
\end{aligned}$$

In (4.39) and (4.40), by rearranging the summation, we have

$$\sum_{t=1}^{T}\sum_{s=1}^{t-1}\frac{\lambda^{t-1-s}}{s}\le\frac{1+\ln(T)}{1-\lambda}.$$

Using (4.37)-(4.40) in (4.36) then yields the result. □

We are now in a position to prove Theorem 4.5.1.

Proof of Theorem 4.5.1. Using Lemma 4.5.2 and the assumption that the learning rate is chosen as $\alpha(t)=\frac{1}{\mu t}$, we have that

$$\begin{aligned}
\mathsf{R}(T)\le&\sum_{t=1}^{T}\sum_{i=1}^{N}L_i\|z_i(t)-\bar{x}(t-1)\|_2+\frac{N}{2\alpha(1)}\|\bar{x}(0)-z^\star\|_2^2 \\
&-\sum_{t=1}^{T}\sum_{i=1}^{N}\mu(z_i(t)-\bar{x}(t-1))^{\mathsf{T}}(\bar{x}(t-1)-z^\star)+\frac{L^2}{2N}\sum_{t=1}^{T}\alpha(t),
\end{aligned}$$

where we have used the fact that

$$\frac{N}{2}\sum_{t=1}^{T-1}\|\bar{x}_t-z^\star\|_2^2\Big(\frac{1}{\alpha(t+1)}-\frac{1}{\alpha(t)}-\mu\Big)=0.$$

Using Corollary 4.3.5 and Lemma 4.5.3, we have

$$\begin{aligned}
\mathsf{R}(T) \leq & \frac{8L}{\delta(1-\lambda)}\Big(\sum_{i=1}^{N}\|x_i(0)\|_1 + \frac{L}{\mu}(1+\ln(T))\Big) + \frac{N}{2\alpha(1)}\|\bar{x}(0) - z^\star\|_2^2 \\
& + \mu\frac{8N}{\delta(1-\lambda)}\sum_{i=1}^{N}\|x_i(0)\|_1\|\bar{x}(0) - z^\star\|_2 + \frac{8}{\delta(1-\lambda)}\sum_{i=1}^{N}\|x_i(0)\|_1 L(1+\ln(T)) \\
& + \frac{8N}{\delta(1-\lambda)}L\|\bar{x}(0) - z^\star\|_2(1+\ln(T)) + \frac{8}{\delta(1-\lambda)}\frac{L^2}{\mu}(1+\ln(T))^2 \\
& + \frac{L^2}{2N\mu}(1+\ln(T)).
\end{aligned}$$

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

In order to establish the proof of Theorem 4.2.2, using the previous result about the network regret, we provide an upper bound on the individual regrets.

Proposition 4.5.4. *Let* $\{f_1^t, f_2^t, ..., f_N^t\}_{t=1}^T$ *be a sequence of convex cost functions that satisfies Assumption 4.2.1. Suppose that the learning rate is chosen as* $\alpha(t) = \frac{1}{\mu t}$*, and the agents use* (4.1)*, over a sequence of* B*-strongly connected graphs, to generate their states. Then for any agent* $j \in \mathcal{V}$*, we have*

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

Proof. First, note that

$$\begin{aligned}
\mathsf{R}^j(T) - \mathsf{R}(T) &= \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z_j(t)) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z_i(t)) \\
&= \sum_{t=1}^{T}\sum_{i=1}^{N} \big(f_i^t(z_j(t)) - f_i^t(z_i(t))\big) \\
&\le \sum_{t=1}^{T}\sum_{i=1}^{N} g_j(t)^{\mathsf{T}}(z_j(t) - z_i(t)) \le \sum_{t=1}^{T}\sum_{i=1}^{N} L_j\|z_j(t) - z_i(t)\|_2.
\end{aligned}$$

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

$$\begin{aligned}
\|z_j(t+1) - z_i(t+1)\|_2^2 &= \|z_j(t+1) - \bar{x}(t)\|_2^2 + \|z_i(t+1) - \bar{x}(t)\|_2^2 \\
&\qquad - 2(z_j(t+1) - \bar{x}(t))^{\mathsf{T}}(z_i(t+1) - \bar{x}(t)) \\
&\le \|z_j(t+1) - \bar{x}(t)\|_2^2 + \|z_i(t+1) - \bar{x}(t)\|_2^2 \\
&\qquad + 2\|z_j(t+1) - \bar{x}(t)\|_2\|z_i(t+1) - \bar{x}(t)\|_2.
\end{aligned}$$

As a result,

$$\|z_j(t+1) - z_i(t+1)\|_2^2 \le 4\Big[\frac{8}{\delta}\Big(\lambda^t\sum_{i=1}^{N}\|x_i(0)\|_1 + \sum_{s=1}^{t}\lambda^{t-s}\sum_{i=1}^{N}\|\alpha(s)g_i(s)\|_1\Big)\Big]^2,$$

where we have used Cauchy-Schwarz inequality and the last inequality follows from Lemma 4.3.4. As a result,

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

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

$$\sum_{t=1}^{T}\sum_{i=1}^{N} L_j \|z_j(t) - z_i(t)\|_2 \leq \sum_{t=1}^{T} N L_j \Big(\frac{16}{\delta} \lambda^{t-1} \sum_{i=1}^{N} \|x_i(0)\|_1 + \sum_{s=1}^{t-1} \lambda^{t-1-s} \frac{L}{\mu s} \Big)$$
$$\leq \frac{16 N L_j}{\delta(1-\lambda)} \Big(\sum_{i=1}^{N} \|x_i(0)\|_1 + \frac{L}{\mu}(1 + \ln(T)) \Big),$$

which establishes the result. □

Proof of Theorem 4.2.2. The proof of Theorem 4.2.2 follows by using the network regret bound in Theorem 4.5.1 and the bound on the difference between the network regret and the individual regret, obtained in Proposition 4.5.4. □

It is worth noting that one can proceed with the proof of Theorem 4.2.2 if the learning rate is instead given by $\alpha(t) = \frac{C}{t}$ where $C \geq 1/\mu$ is a constant.

4.6 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 4.2.2 explicit, at least for some special cases. Motivated by the second statement of Lemma 4.3.4, let us consider the class of regular (undirected) graphs and in particular, the subclass of Ramanujan graphs.

Proposition 4.6.1. *Suppose that $\{\mathcal{G}(t)\}_{t=1}^{T}$ is a* B*-strongly connected sequence of Ramanujan r-regular graphs, $r \geq 3$, of order N. Under the conditions of Theorem 4.2.2, we have*

$$\mathsf{R}^j(T) \leq c_1 \frac{rn^2}{r - 2\sqrt{r-1}} + c_2 \frac{rN}{r - 2\sqrt{r-1}}(1 + \ln(T)) + c_3 \frac{r}{r - 2\sqrt{r-1}}(1 + \ln(T))^2$$

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

Proof. Suppose $\mathcal{G}(t)$ is a Ramanujan r-regular graph with the unweighted adjacency matrix $\mathsf{A}(t)$. Then, using [20, Definition 2.2], we have that $\sigma_2(\mathsf{A}(t)) \leq 2\sqrt{r-1}$. We hence obtain $\lambda \leq \sigma_2(A(t)) \leq \frac{2\sqrt{r-1}}{r}$, where $A(t) = \frac{1}{r}\mathsf{A}(t)$. Consider now the distributed online subgradient push-sum algorithm (4.1), with $A(t)$ as described. We also have $\delta = 1$ for regular graphs. Using Theorem 4.2.2, in particular (4.2), we have that

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

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

$$
\begin{aligned}
C_1 \leq& \frac{8Lr}{r-2\sqrt{r-1}} \sum_{i=1}^{N} \|x_i(0)\|_1 + \frac{N}{2\alpha(1)} \|\bar{x}(0) - z^\star\|_2^2 \\
&+ \frac{8\mu Nr}{r-2\sqrt{r-1}} \sum_{i=1}^{N} \|x_i(0)\|_1 \|\bar{x}(0) - z^\star\|_2 + \frac{16NrL_j}{r-2\sqrt{r-1}} \sum_{i=1}^{N} \|x_i(0)\|_1
\end{aligned}
$$

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

$$
\begin{aligned}
C_1 \leq& \frac{8LNr}{r-2\sqrt{r-1}} \max_{i \in \mathcal{V}} \|x_i(0)\|_1 + \frac{N}{2\alpha(1)} \|\bar{x}(0) - z^\star\|_2^2 \\
&+ \frac{8\mu N^2 r}{r-2\sqrt{r-1}} \max_{i \in \mathcal{V}} \|x_i(0)\|_1 \|\bar{x}(0) - z^\star\|_2 + \frac{16N^2 rL_j}{r-2\sqrt{r-1}} \max_{i \in \mathcal{V}} \|x_i(0)\|_1 \\
\leq& c_1 \frac{N^2 r}{r-2\sqrt{r-1}},
\end{aligned}
$$

where

$$c_1 = \max_{i\in\mathcal{V}} \|x_i(0)\|_1\Big(\frac{8L}{N} + 8\mu\|\bar{x}(0) - z^\star\|_2 + 16L_j\Big) + \frac{\|\bar{x}(0) - z^\star\|_2^2}{2\mu N}.$$

Similarly, we have

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

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

$$\begin{aligned} C_2 \leq& \frac{8rL^2}{\mu(r-2\sqrt{r-1})} + \frac{8Lr}{r-2\sqrt{r-1}}\sum_{i=1}^{N}\|x_i(0)\|_1 + \frac{16NrL_j}{r-2\sqrt{r-1}}\frac{L}{\mu} \\ &+ \frac{8Nr}{r-2\sqrt{r-1}}L\|\bar{x}(0) - z^\star\|_2 + \frac{L^2}{2N\mu}, \end{aligned}$$

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

$$\begin{aligned} C_2 \leq& \frac{8rL^2}{\mu(r-2\sqrt{r-1})} + \frac{8LNr}{r-2\sqrt{r-1}}\max_{i\in\mathcal{V}}\|x_i(0)\|_1 + \frac{16NrL_j}{r-2\sqrt{r-1}}\frac{L}{\mu} \\ &+ \frac{8Nr}{r-2\sqrt{r-1}}L\|\bar{x}(0) - z^\star\|_2 + \frac{L^2}{2N\mu}, \\ \leq& c_2\frac{Nr}{r-2\sqrt{r-1}}, \end{aligned}$$

where

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

Finally, we have

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

where

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

which yields the result. $\square$

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

Chapter 5

Distributed Online ADMM

The main objective of this chapter is to introduce an algorithm, distributed over a network of agents, so that each agent can achieve an individual regret that is sublinear in T. Our proposed algorithm relies on a distributed version of the so-called Alternating Direction Method of Multipliers (ADMM) algorithm. We start by reviewing the ADMM algorithm [3]. The main advantage of this algorithm with respect to the one presented in the previous chapter is that it gives an explicit regret bound in terms of the size of the network. Moreover, this algorithm does not have subgradient step in contrast to the previous one.

5.1 Alternating Direction Method of Multipliers

We start by recalling the Alternating Direction Method of Multipliers algorithm. Consider the following optimization problem with linear constraints:

$$\begin{aligned} &\min_{z\in Z,\xi\in\Xi} f(z)+g(\xi), \qquad (5.1)\\ &\text{s.t.} \quad Az+B\xi=c, \end{aligned}$$

where $z \in Z \subseteq \mathbb{R}^N, \xi \in \Xi \subseteq \mathbb{R}^M, c \in \mathbb{R}^W, A \in \mathbb{R}^{W \times N}, B \in \mathbb{R}^{W \times M}$, $W, M, N \in \mathbb{Z}_{>0}$, and $f : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^M \to \mathbb{R} \cup \{+\infty\}$ are (extended)-real-valued functions, assumed to be closed, proper, and strictly convex. The optimal value of (5.1) is denoted by $p^\star = \inf\{f(z) + g(\xi) \mid Az + B\xi = c\}$. A so-called *augmented Lagrangian function* is defined to find the optimal value of the problem (5.1), using the ADMM algorithm which will be described next. For $\beta \in \mathbb{R}_{>0}$, the augmented Lagrangian is defined as the function $L_\beta : Z \times \Xi \times \mathbb{R}^W \to \mathbb{R} \cup \{+\infty\}$, given by

$$L_\beta(z, \xi, \lambda) = f(z) + g(\xi) + \lambda^\mathsf{T}(Az + B\xi - c) + \frac{\beta}{2}\|Az + B\xi - c\|_2^2. \tag{5.2}$$

The variable λ is called the dual variable and β is called the penalty parameter. The ADMM at each time $t \in \mathbb{Z}_{\geq 0}$ updates variable (z, ξ, λ) using

$$\begin{aligned} z(t+1) &= \operatorname*{argmin}_{z \in Z} \; L_\beta(z, \xi(t), \lambda(t)), \\ \xi(t+1) &= \operatorname*{argmin}_{\xi \in \Xi} \; L_\beta(z(t+1), \xi, \lambda(t)), \\ \lambda(t+1) &= \lambda(t) + \beta(Az(t+1) + B\xi(t+1) - c), \end{aligned}$$

It is shown in [3] that the ADMM algorithm converges to the solution of (5.1), with the following assumptions. Moreover, it can be shown that the converges rate is $O(\frac{1}{T})$ [40, 38].

Assumption 5.1.1. *The unaugmented Lagrangian* $L_0(z, \xi, \lambda) = f(z) + g(\xi) + \lambda^\mathsf{T}(Az + B\xi - c)$ *has a saddle point, i.e., there exist* $(z^\star, \xi^\star, \lambda^\star)$*, for which*

$$L_0(z^\star, \xi^\star, \lambda) < L_0(z^\star, \xi^\star, \lambda^\star) < L_0(z, \xi, \lambda^\star)$$

holds for all $z \neq z^\star, \xi \neq^\star$*, and* $\lambda \neq \lambda^\star$*.*

5.2 Distributed Online Alternating Direction Method of Multipliers

In the distributed optimization problem, a function $F(z) = \sum_{i=1}^{N} f_i(z)$ is distributed among N agents, where each agent $i \in \{1, \cdots, N\}$ has information about its own function $f_i : \mathbb{R} \to \mathbb{R}$ and they cooperatively try to minimize $F(z)$. To do this, they communicate their states through a graph. We consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, N\}$ is the set of agents and $\mathcal{E}$ is the set of edges, through which agents can communicate their states. We label each agent by a number from 1 to N. We denote by $e_{ij} \in \mathcal{E}$ the edge between agent i and j, with $i < j$. We also label each e_{ij} with a number from 1 to M, where $M = |\mathcal{E}|$.

We define the vector $\mathbf{z} = (z_1, \cdots, z_N)^\mathsf{T} \in \mathbb{R}^N$. We also assign a state ξ_{ij} to each edge $e_{ij} \in \mathcal{E}$ and let $\boldsymbol{\xi} = (\xi^1, \cdots, \xi^M)^\mathsf{T}$ be the vector containing all ξ_{ij}s, where l in $\xi^l = \xi_{ij}$ is the label assigned to e_{ij}. For each pair of agents i and j connected with an edge $e_{ij} \in \mathcal{E}$, we assign the constraints $z_i = \xi_{ij}$ and $z_j = \xi_{ij}$, which means that all agents' states must agree, since the graph is connected. Consider now the optimization problem with linear constraints, given by

$$\min_{\mathbf{z} \in Z, \boldsymbol{\xi} \in \Xi} F(\mathbf{z}) = \sum_{i=1}^{N} f_i(z_i), \quad \text{s.t.} \quad A\mathbf{z} + B\boldsymbol{\xi} = 0,$$

where we define $A \in \mathbb{R}^{2M \times N}$, $B \in \mathbb{R}^{2M \times M}$, $M = |\mathcal{E}|$ and $N = |\mathcal{V}|$. Matrix $B = \mathsf{I}_M \otimes \mathbf{1}_2$. For each $e_{ij} \in \mathcal{E}$, the row $[A]^{2l-1}$ has 1 in the i^{th} column and 0 in other columns, and the row $[A]^{2l}$ has 1 in the j^{th} column and 0 in other columns, where l is the label assigned to e_{ij}. We also assign the dual variables λ_{ij} and λ_{ji}, to each

$e_{ij} \in \mathcal{E}$, where λ_{ij} and $\lambda_{ji} \in \mathbb{R}$.

We partition the neighbours of a node i, which we denote by $\mathcal{N}(i)$, into two sets, denoted by $\mathcal{N}_s(i)$ and $\mathcal{N}_\ell(i)$, where $\mathcal{N}_s(i) = \{j \mid e_{ji} \in \mathcal{E}, j < i\}$ is the set of neighbors of agent i with index smaller than i, and $\mathcal{N}_\ell(i) = \{j \mid e_{ij} \in \mathcal{E}, i < j\}$ is the set of neighbors of agent i with index larger than i. Let us clarify our notations using a simple example.

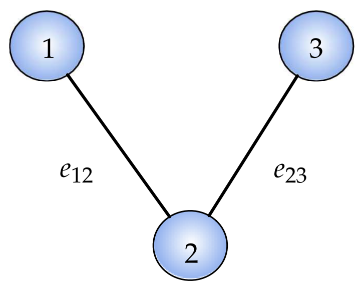

Figure 5.1: The graph for Example 5.2.1

Example 5.2.1. *For the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{1, 2, 3\}$ and $\mathcal{E} = \{e_{12}, e_{23}\}$, shown in Figure 5.1, we have:*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \mathbf{z} = \begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix}, \boldsymbol{\xi} = \begin{Bmatrix} \xi_{12} \\ \xi_{23} \end{Bmatrix}, \boldsymbol{\lambda} = \begin{Bmatrix} \lambda_{12} \\ \lambda_{21} \\ \lambda_{23} \\ \lambda_{32} \end{Bmatrix}, \quad \text{and}$$

$$\mathcal{N}_s(1) = \emptyset,\ \mathcal{N}_\ell(1) = \{2\},\ \mathcal{N}_s(2) = \{1\},\ \mathcal{N}_\ell(2) = \{3\},\ \mathcal{N}_s(3) = \{2\},\ \mathcal{N}_\ell(3) = \emptyset.$$

We also have the following constraints:

$$z_1 = \xi_{12},\ z_2 = \xi_{12},\ z_2 = \xi_{23},\ z_3 = \xi_{23}, \quad \text{so } z_1 = z_2 = z_3.$$

•

In the distributed *online* optimization problem, the function f_i for each agent i changes with time. Since agents do not have access to their function before they choose their states, they cannot minimize the whole function $F(z)$. In this setting, the agents' objective is to bound their regret function, defined in (3.1), sublinearly. Hence, in each iteration $t \in \mathbb{Z}_{>0}$, each agent $i \in \mathcal{V}$ chooses $z_i(t) \in \mathbb{R}^d$, and after that a convex cost function $f_i^t : \mathbb{R}^d \to \mathbb{R}$ is revealed and agent i incurs cost $f_i^t(z_i(t))$.

The *Distributed Online ADMM algorithm* is defined as follows:

1) **Initialization**: choose arbitrary $z_i(1)$ for $i \in \mathcal{V}$ and arbitrary $\lambda_{ij}(1) = -\lambda_{ji}(1)$ and $\xi_{ij}(1)$ for all $e_{ij} \in \mathcal{E}$ which are not necessarily all equal.

2) **Updates**: For $t \geq 0$,

   a) Each agent i updates its estimate $z_i(t)$ in a sequential order using the rule

$$\begin{aligned} z_i(t+1) = \underset{z_i}{\operatorname{argmin}}\ & f_i^t(z_i) + \frac{\beta}{2} \sum_{j \in \mathcal{N}_s(i)} |z_i + \xi_{ji}(t) + \frac{1}{\beta}\lambda_{ij}(t)|^2 \\ & + \frac{\beta}{2} \sum_{j \in \mathcal{N}_\ell(i)} |z_i + \xi_{ij}(t) + \frac{1}{\beta}\lambda_{ij}(t)|^2, \end{aligned} \tag{5.3}$$

b) Each agent i updates ξ_{ji}, for all j in $\mathcal{N}_s(i)$ as

$$\xi_{ji}(t+1) = \underset{\xi_{ji}}{\operatorname{argmin}} \frac{\beta}{2}\Big(|z_j(t+1) + \xi_{ji} + \frac{1}{\beta}\lambda_{ji}(t)|^2 + |z_i(t+1) + \xi_{ji} + \frac{1}{\beta}\lambda_{ij}(t)|^2\Big), \tag{5.4}$$

c) Each agent i updates λ_{ji} and λ_{ij}, for all j in $\mathcal{N}_s(i)$,

$$\lambda_{ij}(t+1) = \lambda_{ij}(t) + \beta(z_i(t+1) + \xi_{ji}(t+1)), \tag{5.5}$$

$$\lambda_{ji}(t+1) = \lambda_{ji}(t) + \beta(z_j(t+1) + \xi_{ji}(t+1)). \tag{5.6}$$

We now collect all λ_{ij}s in a vector $\boldsymbol{\lambda}$, such that for all neighbor pairs i and j, λ_{ij} and λ_{ji} appear consecutively, i.e., $\boldsymbol{\lambda} = [\lambda_{ij}, \lambda_{ji}]_{i,j\in\mathcal{V},e_{ij}\in\mathcal{E}}$. Note that this algorithm is different from the one presented in [15], in the sense it does not have a subgradient step.

5.3 Regret Bounds for the Distributed Online ADMM

We provide an upper bound on each agent's individual regret, given by (3.1), under the Distributed Online ADMM algorithm.

Theorem 5.3.1. *Let $\{f_1^t, \cdots, f_N^t\}_{t=1}^T$ be a sequence of convex functions where, for each $i \in \mathcal{V}$, f_i^t has L-bounded subgradient. Let the sequences $\{\mathbf{z}(t) = (z_1(t), \cdots, z_N(t))\}_{t=1}^T$, $\{\boldsymbol{\xi}(t) = (\xi^1(t), \cdots, \xi^M(t))\}_{t=1}^T$, and $\{\boldsymbol{\lambda}(t) = (\lambda^1(t), \cdots, \lambda^{2M}(t))\}_{t=1}^T$ be generated by the Distributed Online ADMM over an undirected fixed connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$. Suppose that there exist $z^\star \in \operatorname{argmin} \sum_{i=1}^N \sum_{t=1}^T f_i^t(z)$ and choose $\mathbf{z}^\star = (z^\star, \cdots, z^\star), \boldsymbol{\xi}^\star$ satisfying $A\mathbf{z}^\star + B\boldsymbol{\xi}^\star = 0$, and let $\beta = (N+1)\sqrt{T}$.*

Then we have

$$\begin{aligned}\mathsf{R}^j(T) \leq & \frac{1}{2(N+1)\sqrt{T}}\|\boldsymbol{\lambda}(1)\|_2^2 + \sqrt{T}ML^2(N+1), \\ & + \frac{(N+1)\sqrt{T}}{2}\|B\boldsymbol{\xi}^\star - B\boldsymbol{\xi}(1)\|_2^2, \\ & + \frac{N\sqrt{T}}{2}\|A\mathbf{z}(1) + B\boldsymbol{\xi}(1)\|_2^2.\end{aligned}$$

Before stating the proof, let us mention a few key points about this results, especially in comparison with other available distributed algorithms for online optimization. This algorithm provides a regret bound of order $\mathcal{O}(\sqrt{T})$, which is similar to the ones with subgradient flow protocols. However, this algorithm gives us an *explicit* dependency of the regret bound on the size of the network. In particular, the regret bounds for most existing gradient flow algorithms rely on spectral radius of adjacency matrix, see for example [21]. It is also worth mentioning that the Distributed Online ADMM algorithm presented in [15] considers the problem of bounding the *network* regret, and not the *individual* regret. As our proof illustrates, the latter problem is more challenging.

We now start the process of proving Theorem 5.3.1. Our proof relies on a sequence of results, which we establish via Lemmas 5.3.2-5.3.7.

Lemma 5.3.2. *Let the sequence* $\{\mathbf{z}(t), \boldsymbol{\xi}(t), \boldsymbol{\lambda}(t)\}_{t=1}^T$ *be generated by the Distributed Online ADMM. Then for any* $\tilde{\mathbf{z}} \in \mathbb{R}^N$*, we have*

$$\sum_{i=1}^N f_i^t(z_i(t+1)) - \sum_{i=1}^N f_i^t(\tilde{z}_i) \leq -\left(A(\mathbf{z}(t+1) - \tilde{\mathbf{z}})\right)^\mathsf{T}\Big(\boldsymbol{\lambda}(t+1) + \beta B(\boldsymbol{\xi}(t) - \boldsymbol{\xi}(t+1))\Big).$$

Proof. For each $t \in \mathbb{Z}_{>0}$, since $z_i(t+1)$ satisfies (5.3), we have

$$0 \in \partial f_i^t(z_i(t+1)) + \beta\Big(\sum_{j \in \mathcal{N}_s(i)} (z_i(t+1) + \xi_{ji}(t) + \frac{1}{\beta}\lambda_{ij}(t)) + \sum_{j \in \mathcal{N}_\ell(i)} (z_i(t+1) + \xi_{ij}(t) + \frac{1}{\beta}\lambda_{ij}(t))\Big).$$

Hence,

$$-\beta\Big(\sum_{j \in \mathcal{N}_s(i)} (z_i(t+1) + \xi_{ji}(t) + \frac{1}{\beta}\lambda_{ij}(t)) + \sum_{j \in \mathcal{N}_\ell(i)} (z_i(t+1) + \xi_{ij}(t) + \frac{1}{\beta}\lambda_{ij}(t))\Big), \in \partial f_i^t(z_i(t+1)). \tag{5.7}$$

Since each f_i^t is convex, we have

$$f_i^t(z_i(t+1)) - f_i^t(\tilde{z}_i) \leq (z_i(t+1) - \tilde{z}_i) g_i^t(z_i(t+1)),$$

for all $g_i^t(z_i(t+1)) \in \partial f_i^t(z_i(t+1))$. As a result, using (5.7), we have

$$f_i^t(z_i(t+1)) - f_i^t(\tilde{z}_i) \leq -\beta(z_i(t+1) - \tilde{z}_i)\Big(\sum_{j \in \mathcal{N}_s(i)} (z_i(t+1) + \xi_{ji}(t) + \frac{1}{\beta}\lambda_{ij}(t)) + \sum_{j \in \mathcal{N}_\ell(i)} (z_i(t+1) + \xi_{ij}(t) + \frac{1}{\beta}\lambda_{ij}(t))\Big).$$

Using (5.5) and (5.6), we have

$$\begin{aligned} f_i^t(z_i(t+1)) - f_i^t(\tilde{z}_i) \leq &- (z_i(t+1) - \tilde{z}_i)\Big(\sum_{j \in \mathcal{N}_s(i)} (\lambda_{ij}(t+1) + \beta(\xi_{ji}(t) - \xi_{ji}(t+1))) \\ &+ \sum_{j \in \mathcal{N}_\ell(i)} (\lambda_{ij}(t+1) + \beta(\xi_{ij}(t) - \xi_{ij}(t+1)))\Big), \\ = &- (z_i(t+1) - \tilde{z}_i)\Big(([A]_i)^\mathsf{T} \boldsymbol{\lambda}(t+1) \\ &+ \beta([A^\mathsf{T} B]_i)(\boldsymbol{\xi}(t) - \boldsymbol{\xi}(t+1))\Big), \end{aligned}$$

where we have used the definition of A and B in the last equality. Hence,

$$\begin{aligned} \sum_{i=1}^N f_i^t(z_i(t+1)) - \sum_{i=1}^N f_i^t(\tilde{z}_i) \leq &- (\mathbf{z}(t+1) - \tilde{\mathbf{z}})^\mathsf{T}\Big(A^\mathsf{T} \boldsymbol{\lambda}(t+1) \\ &+ \beta[A^\mathsf{T} B](\boldsymbol{\xi}(t) - \boldsymbol{\xi}(t+1))\Big) \\ = &- (A(\mathbf{z}(t+1) - \tilde{\mathbf{z}}))^\mathsf{T}\Big(\boldsymbol{\lambda}(t+1) + \beta B(\boldsymbol{\xi}(t) - \boldsymbol{\xi}(t+1))\Big), \end{aligned}$$

which proves the claim. □

Lemma 5.3.3. *Let the sequence $\{\mathbf{z}(t), \boldsymbol{\xi}(t), \boldsymbol{\lambda}(t)\}_{t=1}^T$ be generated by the Distributed Online ADMM. Then, for all $t \in \mathbb{Z}_{>0}$, we have*

$$B^\mathsf{T} \boldsymbol{\lambda}(t) = 0. \tag{5.8}$$

Proof. Since $\xi_{ji}(t+1)$ is given by (5.4), we have

$$\beta\Big(z_j(t+1) + \xi_{ji}(t+1) + \frac{1}{\beta}\lambda_{ji}(t) + z_i(t+1) + \xi_{ji}(t+1) + \frac{1}{\beta}\lambda_{ij}(t)\Big) = 0.$$

Now, using (5.5) and (5.6),

$$(\lambda_{ji}(t+1) + \lambda_{ij}(t+1)) = 0,$$

or equivalently, $B^{\mathsf{T}}\boldsymbol{\lambda}(t) = 0$, for all $t \in \mathbb{Z}_{>0}$. □

Lemma 5.3.4. *Let the sequence $\{\mathbf{z}(t), \boldsymbol{\xi}(t), \boldsymbol{\lambda}(t)\}_{t=1}^{T}$ be generated by the Distributed Online ADMM. For any $\mathbf{z}^\star, \boldsymbol{\xi}^\star$ satisfying $A\mathbf{z}^\star + B\boldsymbol{\xi}^\star = 0$, we have*

$$\begin{aligned}\sum_{i=1}^{N} f_i^t(z_i(t+1)) - \sum_{i=1}^{N} f_i^t(z^\star) \leq & \frac{1}{2\beta}(\|\boldsymbol{\lambda}(t)\|_2^2 - \|\boldsymbol{\lambda}(t+1)\|_2^2) - \frac{\beta}{2}\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t)\|_2^2 \\ & + \frac{\beta}{2}\Big(\|B\boldsymbol{\xi}^\star - B\boldsymbol{\xi}(t)\|_2^2 - \|B\boldsymbol{\xi}^\star - B\boldsymbol{\xi}(t+1)\|_2^2\Big).\end{aligned}$$

Proof. From Lemma 5.3.2, for $\tilde{\mathbf{z}} = \mathbf{z}^\star$, we have

$$\begin{aligned}\sum_{i=1}^{N} f_i^t(z_i(t+1)) - \sum_{i=1}^{N} f_i^t(z^\star) \leq & -(A\mathbf{z}(t+1) - A\mathbf{z}^\star)^{\mathsf{T}}\Big(\boldsymbol{\lambda}(t+1) \\ & \quad + \beta B(\boldsymbol{\xi}(t) - \boldsymbol{\xi}(t+1))\Big) \\ = & -(A\mathbf{z}(t+1) + B\boldsymbol{\xi}^\star)^{\mathsf{T}}\boldsymbol{\lambda}(t+1) \\ & \quad - \beta(A\mathbf{z}(t+1) + B\boldsymbol{\xi}^\star)^{\mathsf{T}}(B\boldsymbol{\xi}(t) - B\boldsymbol{\xi}(t+1)), \\ = & -(A\mathbf{z}(t+1) + B\boldsymbol{\xi}^\star)^{\mathsf{T}}\boldsymbol{\lambda}(t+1) \\ & \quad - \frac{\beta}{2}\Big(\|B\boldsymbol{\xi}(t+1) - B\boldsymbol{\xi}^\star\|_2^2 - \|B\boldsymbol{\xi}(t) - B\boldsymbol{\xi}^\star\|_2^2 \\ & \quad + \|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t)\|_2^2 \\ & \quad - \|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2\Big),\end{aligned} \tag{5.9}$$

where we have used the fact that for all $u_1, u_2, u_3, u_4 \in \mathbb{R}^d$,

$$(u_1 + u_2)^\mathsf{T}(u_3 - u_4) = \frac{1}{2}(\|u_4 - u_2\|_2^2 - \|u_3 - u_2\|_2^2 + \|u_1 + u_3\|_2^2 - \|u_1 + u_4\|_2^2), \quad (5.10)$$

to obtain the last equality. Note that by Lemma 5.3.3, we have $B^\mathsf{T}\boldsymbol{\lambda}(t+1) = 0$; hence, we can rewrite the sum of the first and the last terms in (5.9) as

$$\begin{aligned}
&-(A\mathbf{z}(t+1) + B\boldsymbol{\xi}^\star)^\mathsf{T}\boldsymbol{\lambda}(t+1) + \frac{\beta}{2}\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 \\
&= -(A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1))^\mathsf{T}\boldsymbol{\lambda}(t+1) + \frac{\beta}{2}\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2, \\
&= \frac{1}{2\beta}\Big(-2(\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t))^\mathsf{T}\boldsymbol{\lambda}(t+1) + \|\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)\|_2^2\Big), \\
&= \frac{1}{2\beta}\Big(\|\boldsymbol{\lambda}(t)\|_2^2 - \|\boldsymbol{\lambda}(t+1)\|_2^2\Big),
\end{aligned}$$

which completes the proof. ☐

Lemma 5.3.5. *Let the sequence $\{\mathbf{z}(t), \boldsymbol{\xi}(t), \boldsymbol{\lambda}(t)\}_{t=1}^T$ be generated by the Distributed Online ADMM. For all $t \in \mathbb{Z}_{>0}$,*

$$\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 + \|B\boldsymbol{\xi}(t+1) - B\boldsymbol{\xi}(t)\|_2^2 = \|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t)\|_2^2.$$

Proof. Using Lemma 5.3.3, we can write

$$\begin{aligned}
0 &= (\boldsymbol{\xi}(t+1) - \boldsymbol{\xi}(t))^\mathsf{T} B^\mathsf{T}\boldsymbol{\lambda}(t+1), \\
0 &= (\boldsymbol{\xi}(t+1) - \boldsymbol{\xi}(t))^\mathsf{T} B^\mathsf{T}\boldsymbol{\lambda}(t).
\end{aligned}$$

Adding these equation, we can write

$$
\begin{aligned}
0 =&(\boldsymbol{\xi}(t+1) - \boldsymbol{\xi}(t))^\mathsf{T} B^\mathsf{T}(\boldsymbol{\lambda}(t+1) - \boldsymbol{\lambda}(t)),\\
=&\beta(B\boldsymbol{\xi}(t+1) - B\boldsymbol{\xi}(t))^\mathsf{T}(A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)),\\
=&\frac{\beta}{2}(\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 + \|B\boldsymbol{\xi}(t+1) - B\boldsymbol{\xi}(t)\|_2^2 - \|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t)\|_2^2),
\end{aligned}
$$

where we have used $(u_1 - u_2)^\mathsf{T}(u_3 + u_1) = \frac{1}{2}(\|u_3 + u_1\|_2^2 + \|u_1 - u_2\|_2^2 - \|u_3 + u_2\|_2^2)$, for all $u_1, u_2, u_3 \in \mathbb{R}^d$. This proves the claim. $\square$

Lemma 5.3.6. *Let $\{f_1^t, \cdots, f_N^t\}_{t=1}^T$ be a sequence of convex functions, where for each $i \in \mathcal{V}$, f_i^t has an L-bounded subgradient. Let the sequence $\{\mathbf{z}(t), \boldsymbol{\xi}(t), \boldsymbol{\lambda}(t)\}_{t=1}^T$ be generated by the Distributed Online ADMM. Then for all $t \in \mathbb{Z}_{>0}$ and some $\alpha \in \mathbb{R}$, we have*

$$
\begin{aligned}
\sum_{i=1}^N f_i^t(z_i(t)) - \sum_{i=1}^N f_i^t(z_i(t+1)) \leq& \frac{2ML^2}{\alpha} + \frac{\alpha}{2}(\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2\\
&+ \|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2 + \|B\boldsymbol{\xi}(t+1) - B\boldsymbol{\xi}(t)\|_2^2).
\end{aligned}
$$

Proof. Since f_i^t is a convex function, we can write

$$
\begin{aligned}
f_i^t(z_i(t)) - f_i^t(z_i(t+1)) &\leq g_i^t(z_i(t))(z_i(t) - z_i(t+1)),\\
&\leq |g_i^t(z_i(t))||z_i(t) - z_i(t+1)| \leq L|z_i(t) - z_i(t+1)|,
\end{aligned}
$$

for all $g_i^t(z_i(t)) \in \partial f(z_i(t))$, where we have used that $g_i^t(z) \leq L$. Now we have

$$
\sum_{i=1}^N f_i^t(z_i(t)) - \sum_{i=1}^N f_i^t(z_i(t+1)) \leq L \sum_{i=1}^N |z_i(t) - z_i(t+1)|.
$$

Hence, we can write

$$\begin{aligned}\sum_{i=1}^{N}|z_i(t)-z_i(t+1)| \leq& \|A\mathbf{z}(t)-A\mathbf{z}(t+1)\|_1 \leq \sqrt{2M}\|A\mathbf{z}(t)-A\mathbf{z}(t+1)\|_2 \\ \leq& \sqrt{2M}(\|A\mathbf{z}(t)+B\boldsymbol{\xi}(t)\|_2+\|A\mathbf{z}(t+1)+B\boldsymbol{\xi}(t)\|_2).\end{aligned}$$

where we have used the fact that since the graph is connected, for all $i \in \mathcal{V}$, $z_i(t)$ has appeared in some elements of the vector $A\mathbf{z}(t)$. Using this, we have

$$\begin{aligned}\sum_{i=1}^{N} f_i^t(z_i(t)) - \sum_{i=1}^{N} f_i^t(z_i(t+1)) \leq& L\sqrt{2M}\|A\mathbf{z}(t)+B\boldsymbol{\xi}(t)\|_2 \\ &+ L\sqrt{2M}\|A\mathbf{z}(t+1)+B\boldsymbol{\xi}(t)\|_2, \\ \leq& \frac{2ML^2}{2\alpha}+\frac{\alpha}{2}\|A\mathbf{z}(t)+B\boldsymbol{\xi}(t)\|_2^2+\frac{2ML^2}{2\alpha} \\ &+\frac{\alpha}{2}\|A\mathbf{z}(t+1)+B\boldsymbol{\xi}(t)\|_2^2 \\ \leq& \frac{2ML^2}{\alpha}+\frac{\alpha}{2}(\|A\mathbf{z}(t)+B\boldsymbol{\xi}(t)\|_2^2 \\ &+\|A\mathbf{z}(t+1)+B\boldsymbol{\xi}(t+1)\|_2^2+\|B\boldsymbol{\xi}(t)-B\boldsymbol{\xi}(t+1)\|_2^2),\end{aligned}$$

where in the second inequality we have used the Young's inequality: $2u_1^{\mathsf{T}}u_2 \leq \|u_1\|_2^2 + \|u_2\|_2^2$ for all $u_1, u_2 \in \mathbb{R}^d$. This completes the proof. □

The next lemma will be instrumental in bounding the *individual regret function* for one particular agent's states, say the jth agent. It is the final step in proving our main result.

Lemma 5.3.7. *Let $\{f_1^t, \cdots, f_N^t\}_{t=1}^T$ be a sequence of convex functions, where for each $i \in \mathcal{V}$, f_i^t has L-bounded subgradient. Let the sequence $\{\mathbf{z}(t), \boldsymbol{\xi}(t), \boldsymbol{\lambda}(t)\}_{t=1}^T$ be generated by the Distributed Online ADMM. Then for all $t \in \mathbb{Z}_{>0}$, all $j \in \mathcal{V}$, and all*

$\alpha \in \mathbb{R}_{>0}$,

$$\sum_{i=1}^{N} f_i^t(z_j(t)) - \sum_{i=1}^{N} f_i^t(z_i(t)) \leq \frac{2ML^2(N-1)}{2\alpha} + \frac{(N-1)\alpha}{2}\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2.$$

Proof. Since f_i^t is convex with L-bounded subgradients, we have that

$$f_i^t(z_j(t)) - f_i^t(z_i(t)) \leq g_i^t(z_j(t))(z_j(t) - z_i(t)) \leq |g_i^t(z_j(t))||z_j(t) - z_i(t)| \leq L|z_j(t) - z_i(t)|.$$

Hence,

$$\sum_{i=1}^{N} f_i^t(z_j(t)) - \sum_{i=1}^{N} f_i^t(z_i(t)) \leq L\sum_{i=1}^{N} |z_j(t) - z_i(t)|.$$

Note that for neighbor pairs i and j, we have

$$|z_j(t) - z_i(t)| \leq |z_j(t) - \xi_{ij}(t)| + |z_i(t) - \xi_{ij}(t)| \leq \|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_1.$$

Note that, since the graph is connected, for agent i and j that are not neighbors, there exists a path connecting them, and we have

$$|z_j(t) - z_i(t)| \leq \|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_1.$$

As a result,

$$\sum_{i=1}^{N} |z_j(t) - z_i(t)| \leq (N-1)\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_1 \leq (N-1)\sqrt{2M}\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2.$$

Now, for any $\alpha > 0$,

$$\begin{aligned}\sum_{i=1}^{N} f_i^t(z_j(t)) - \sum_{i=1}^{N} f_i^t(z_i(t)) \leq & L(N-1)\sqrt{2M}\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2 \\ \leq & \frac{2ML^2(N-1)}{2\alpha} + \frac{(N-1)\alpha}{2}\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2,\end{aligned}$$

where we have used the Young's inequality. This completes the proof. □

We are now in a position to prove Theorem 5.3.1.

Proof. (Theorem 5.3.1): Using Lemmas 5.3.4, 5.3.6, and 5.3.7, we can write

$$\begin{aligned}\sum_{i=1}^{N} f_i^t(z_j(t)) - \sum_{i=1}^{N} f_i^T(z^\star) \leq & \frac{1}{2\beta}(\|\boldsymbol{\lambda}(t)\|_2^2 - \|\boldsymbol{\lambda}(t+1)\|_2^2) + \frac{ML^2(N+1)}{\alpha} \\ & - \frac{\beta}{2}(\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 + \|B\boldsymbol{\xi}(t) - B\boldsymbol{\xi}(t+1)\|_2^2) \\ & + \frac{\beta}{2}\left(\|B\boldsymbol{\xi}^\star - B\boldsymbol{\xi}(t)\|_2^2 - \|B\boldsymbol{\xi}^\star - B\boldsymbol{\xi}(t+1)\|_2^2\right) \\ & + \frac{\alpha}{2}(\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 + \|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2) \\ & + \frac{\alpha}{2}\|B\boldsymbol{\xi}(t+1) - B\boldsymbol{\xi}(t)\|_2^2 + \frac{(N-1)\alpha}{2}\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2.\end{aligned} \tag{5.11}$$

Note that

$$\begin{aligned}& - \frac{\beta}{2}\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 + \frac{\alpha}{2}(\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 \\ & \quad + \|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2) + \frac{(N-1)\alpha}{2}\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2 \\ & = -\frac{\beta-\alpha}{2}\|A\mathbf{z}(t+1) + B\boldsymbol{\xi}(t+1)\|_2^2 + \frac{N\alpha}{2}\|A\mathbf{z}(t) + B\boldsymbol{\xi}(t)\|_2^2.\end{aligned} \tag{5.12}$$

For $\beta \geq (N+1)\alpha$, using (5.12), we can write

$$
\begin{aligned}
&\sum_{t=1}^{T} -\frac{\beta-\alpha}{2}\|A\mathbf{z}(t+1)+B\boldsymbol{\xi}(t+1)\|_2^2+\sum_{t=1}^{T}\frac{N\alpha}{2}\|A\mathbf{z}(t)+B\boldsymbol{\xi}(t)\|_2^2,\\
&=\sum_{t=1}^{T} -\frac{\beta-\alpha}{2}\|A\mathbf{z}(t+1)+B\boldsymbol{\xi}(t+1)\|_2^2+\sum_{s=0}^{T-1}\frac{N\alpha}{2}\|A\mathbf{z}(s+1)+B\boldsymbol{\xi}(s+1)\|_2^2,\\
&=\sum_{t=1}^{T-1} -\frac{\beta-\alpha-N\alpha}{2}\|A\mathbf{z}(t+1)+B\boldsymbol{\xi}(t+1)\|_2^2-\frac{\beta-\alpha}{2}\|A\mathbf{z}(T+1)+B\boldsymbol{\xi}(T+1)\|_2^2\\
&\qquad+\frac{N\alpha}{2}\|A\mathbf{z}(1)+B\boldsymbol{\xi}(1)\|_2^2,\\
&\leq\frac{N\alpha}{2}\|A\mathbf{z}(1)+B\boldsymbol{\xi}(1)\|_2^2.
\end{aligned}
\tag{5.13}
$$

We also have

$$
-\frac{\beta}{2}\|B\boldsymbol{\xi}(t)-B\boldsymbol{\xi}(t+1)\|_2^2+\frac{\alpha}{2}\|B\boldsymbol{\xi}(t+1)-B\boldsymbol{\xi}(t)\|_2^2\leq 0, \tag{5.14}
$$

for $\beta \geq \alpha$. Using (5.13) and (5.14) along with (5.11), we have

$$
\begin{aligned}
\sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z_j(t))-\sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z_i(t)) \leq &\sum_{t=1}^{T}\frac{1}{2\beta}(\|\boldsymbol{\lambda}(t)\|_2^2-\|\boldsymbol{\lambda}(t+1)\|_2^2)+\sum_{t=1}^{T}\frac{ML^2(N+1)}{\alpha}\\
&+\sum_{t=1}^{T}\frac{\beta}{2}\Big(\|B\boldsymbol{\xi}^\star-B\boldsymbol{\xi}(t)\|_2^2-\|B\boldsymbol{\xi}^\star-B\boldsymbol{\xi}(t+1)\|_2^2\Big)\\
&+\frac{N\alpha}{2}\|A\mathbf{z}(1)+B\boldsymbol{\xi}(1)\|_2^2,\\
\leq&\frac{1}{2\beta}\|\boldsymbol{\lambda}(1)\|_2^2+\frac{T}{\alpha}ML^2(N+1)\\
&+\frac{\beta}{2}\|B\boldsymbol{\xi}^\star-B\boldsymbol{\xi}(1)\|_2^2+\frac{N\alpha}{2}\|A\mathbf{z}(1)+B\boldsymbol{\xi}(1)\|_2^2,
\end{aligned}
$$

where we have used the telescopic series to establish the second inequality. By choosing $\alpha = \sqrt{T}$ and $\beta = (N+1)\sqrt{T}$, we have

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z_j(t)) - \sum_{t=1}^{T}\sum_{i=1}^{N} f_i^t(z_i(t)) \leq & \frac{1}{2(N+1)\sqrt{T}}\|\boldsymbol{\lambda}(1)\|_2^2 + \sqrt{T}ML^2(N+1) \\ & + \frac{(N+1)\sqrt{T}}{2}\|B\boldsymbol{\xi}^\star - B\boldsymbol{\xi}(1)\|_2^2 \\ & + \frac{N\sqrt{T}}{2}\|A\mathbf{z}(1) + B\boldsymbol{\xi}(1)\|_2^2.\end{aligned}$$

This completes the proof. □

Chapter 6

Application to Sensor Networks

In this chapter, we provide some examples on localization in sensor networks to illustrate the performance of our algorithms.

6.1 Sensor Networks

We provide an example using our results for localization in sensor networks, motivated by [14]. 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 $t \in \{1, \cdots, T\}$, 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 $v = 0$. We consider the average squared error, so 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, i.e., the w_i^t are zero mean, independent and identically distributed random variables. In the offline setting, we have all the information to compute the optimal estimate, which is given by

$$s^\star = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{N} P_i^\mathsf{T} P_i\right)^{-1}\left(\sum_{i=1}^{N} P_i^\mathsf{T} q_i^t\right).$$

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. We use both algorithms to estimate the target.

In our simulations, $d = d_i = 1$ and 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 and independently from a uniform distribution. The cost function for sensor i at each time t is given by the mapping $f_i^t : \mathbb{R} \to \mathbb{R}$, where $f_i^t(\hat{s}) = \frac{1}{2}(q_i^t - P_i\hat{s})^2$ and $P_i \in \mathbb{R}$.

6.2 Results using Subgradient-Push Algorithm

Using subgradient-push algorithm, we consider a scenario in which a network of 100 sensors is used to observe. At each time step $t \in \{1, \cdots, T\}$, 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.

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.

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 $t \in \{1, \cdots, T\}$ 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 $i \in \{1, \ldots, N\}$, 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 6.8 shows the states of four sensors over 100 time iterations. By using the distributed online subgradient push-sum algorithm (4.1), 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 6.9 shows the average individual regret of the two sensors with the maximum and minimum average regrets over time.

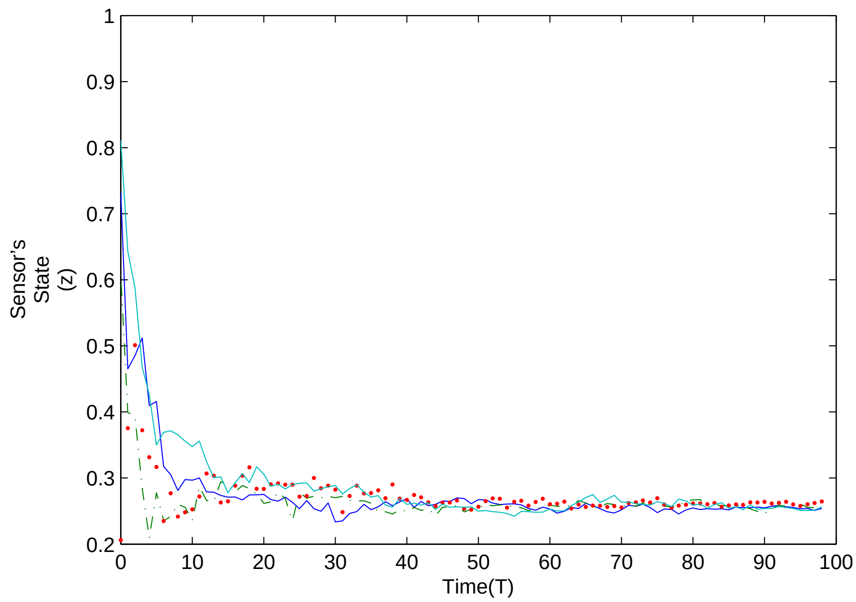

Figure 6.1: Sensors' state estimation vs. time for four of the sensors are shown. The network consists of 100 sensors communicating over a sequence of $\mathcal{G}(100, 0.2, 2)$ random directed graph. The ith sensor observes $q_i^t = a_i^t s + b_i^t$, where a_i^t and b_i^t are chosen at random from a uniform distribution on $[0, 2]$ and $[-\frac{1}{2}, \frac{1}{2}]$, respectively. We use the 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 illustrates consensus among sensors.

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: Consider a scenario with the assumptions as before, with the exception that there is no communication

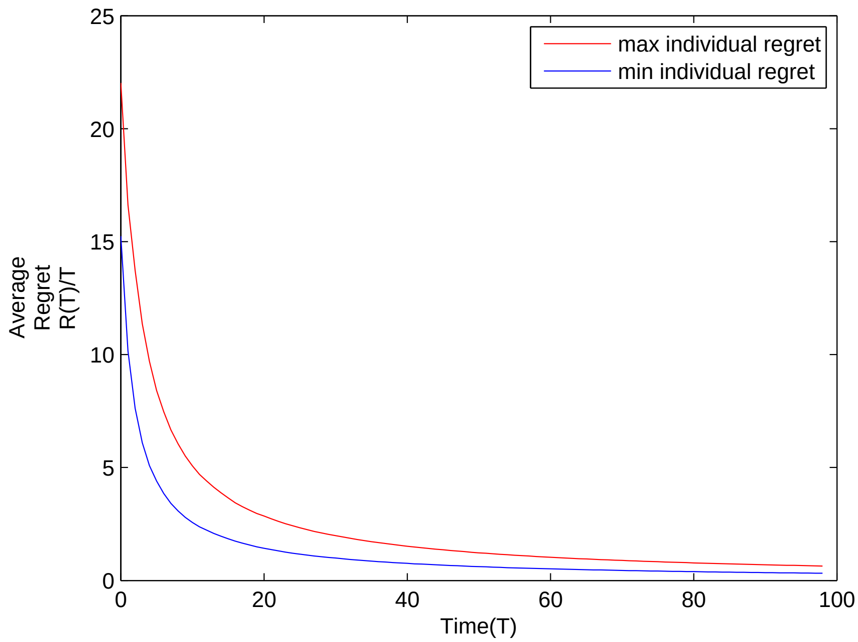

Figure 6.2: Average regrets over time ($\mathsf{R}^j(T)/T$) vs. T for two sensors with the maximum and minimum average regrets are shown, where the same assumptions as the ones in Figure 6.8 hold.

between sensors. Figure 6.3 and Figure 6.4 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 6.5, where the sensors' estimates

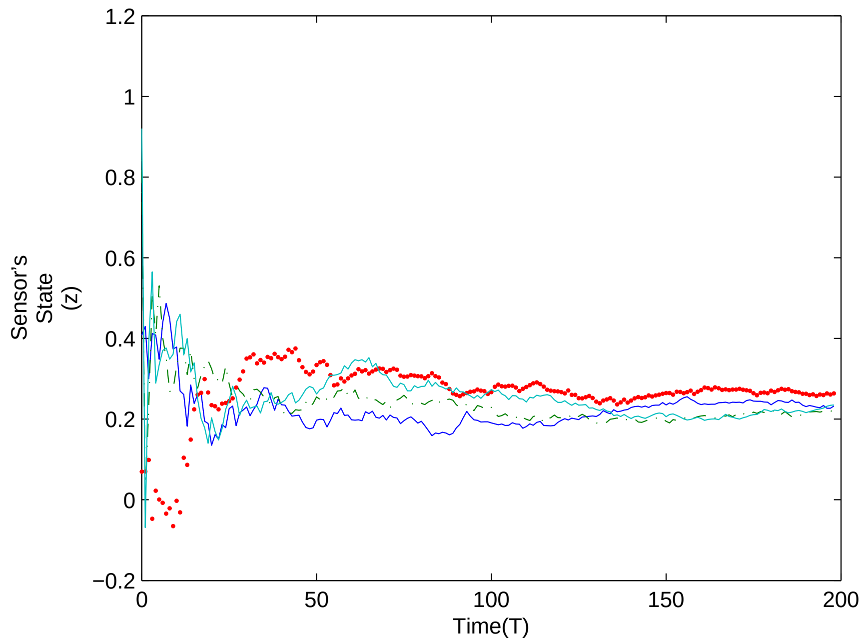

Figure 6.3: Sensors' state estimation vs. time for four of the sensors are shown. The network consists of 100 sensors with no communications. The ith sensor observes $q_i^t = a_i^t s + b_i^t$, where a_i^t and b_i^t are chosen at random from a uniform distribution on $[0, 2]$ and $[-0.5, 0.5]$, respectively. We use distributed online subgradient push-sum algorithm to estimate $\hat{s}$ which minimizes the cost function $f(\hat{s}) = \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{1}{2}(q_i^t - P_i\hat{s})^2$.

approach the expected value of sensor observation. Figure 6.6 shows the individual regret goes to zero as time increases without bound.

6.3 Results using Distributed Online ADMM Algorithm

Consider a network of $N = 8$ sensors that are used to estimate a variable $s \in \mathbb{R}$. The communication graph is given in Figure 6.7. Each sensor $i \in \mathcal{V}$, at each time $t \in \{1, \cdots, T\}$, observes $q_i^t \in \mathbb{R}$, which is a noisy observation of variable s. In this example, we assume that this noisy observation given by $q_i^t = a_i^t s + b_i^t$, where $a_i^t \in [0, 2]$ and $b_i^t \in [-0.5, 0.5]$ are chosen at random and independently from a uniform

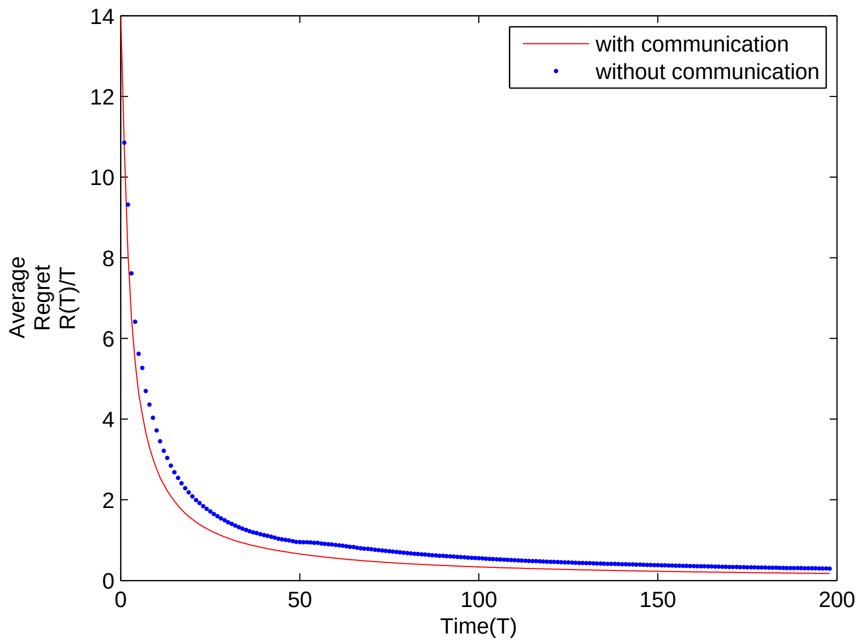

Figure 6.4: Average individual regrets vs. time for one sensor, picked at random among 100 sensors, in the presence and absence of communications over time are shown, where the same assumptions as the ones in Figure 6.3 hold. and we consider two cases: First, there is communication between sensors and second, there is no communication between them. The results shows that communication gives a better regret.

distribution. We use the squared observation error as the cost function, i.e., $f_i^t(s) = \frac{1}{2}(q_i^t - s)^2$. Hence, the best estimation of s is the minimizer of the function

$$\hat{s} = \underset{s}{\text{argmin}} F(s) = \underset{s}{\text{argmin}} \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{1}{2}(q_i^t - s)^2.$$

Let us assume the actual value of $s = \frac{1}{4}$.

We use the Distributed Online ADMM to estimate the variable s. Here, the noise characteristics are not known to the agents. The following figures show the results.

Figure 6.8 shows the states of four sensors over 200 time iterations. By using the Distributed Online ADMM algorithm, the sensors reach a consensus near the

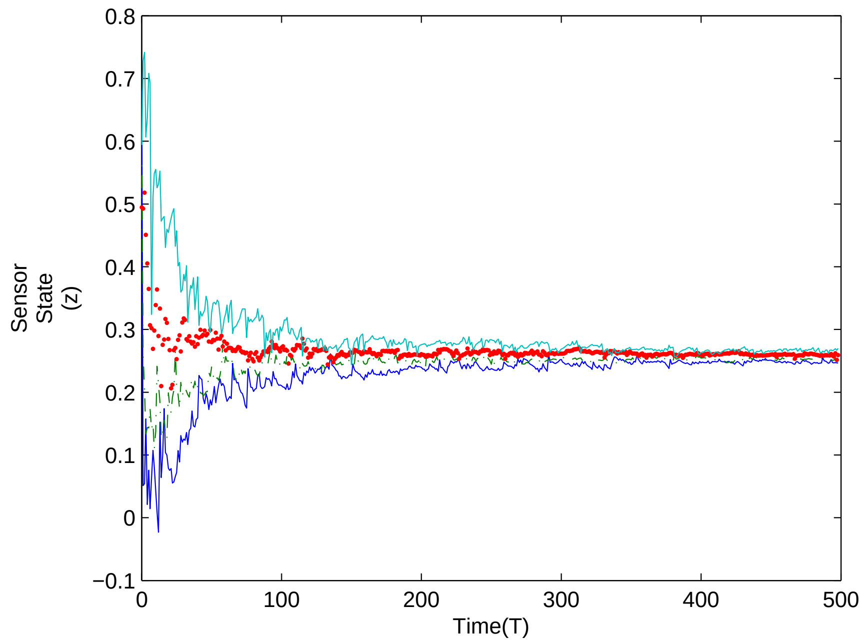

Figure 6.5: Sensors' state estimation vs. time for four of the sensors are shown. The network consists of 100 sensors communicating through a sequence of $\mathcal{G}(100, 0.2, 2)$ random directed graph. The ith sensor observes $q_i^t = a_i^t s + b_i^t$ where a_i^t and b_i^t are chosen at random from 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.

expected value of $\hat{s}$, which is $\frac{1}{4}$, as shown in Figure 6.8. Figure 6.9 shows the average individual regret of the two sensors with the maximum and minimum average regrets over time.

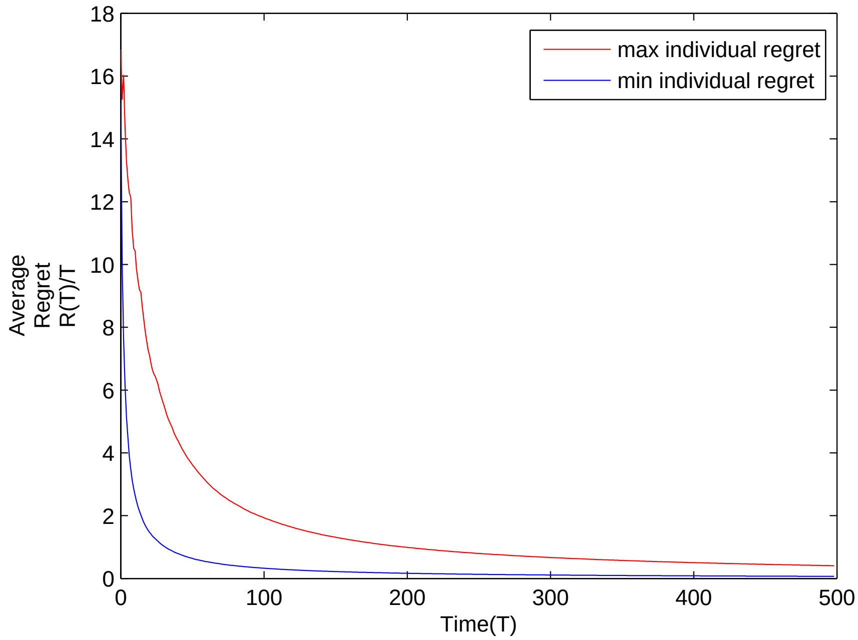

Figure 6.6: Average individual regret over time $R^j(T)/T$ vs. time for two sensors is shown, one has the maximum average regret and the other one has the minimum average regret, where the same assumptions as the ones in Figure 6.5 hold.

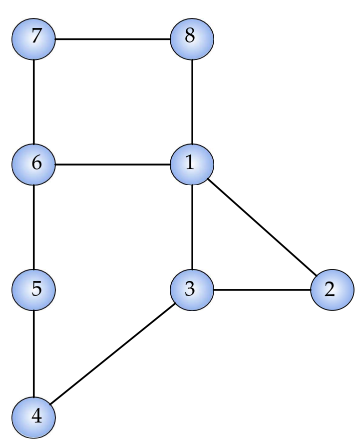

Figure 6.7: $N = 8$ sensors are communicating through the graph depicted

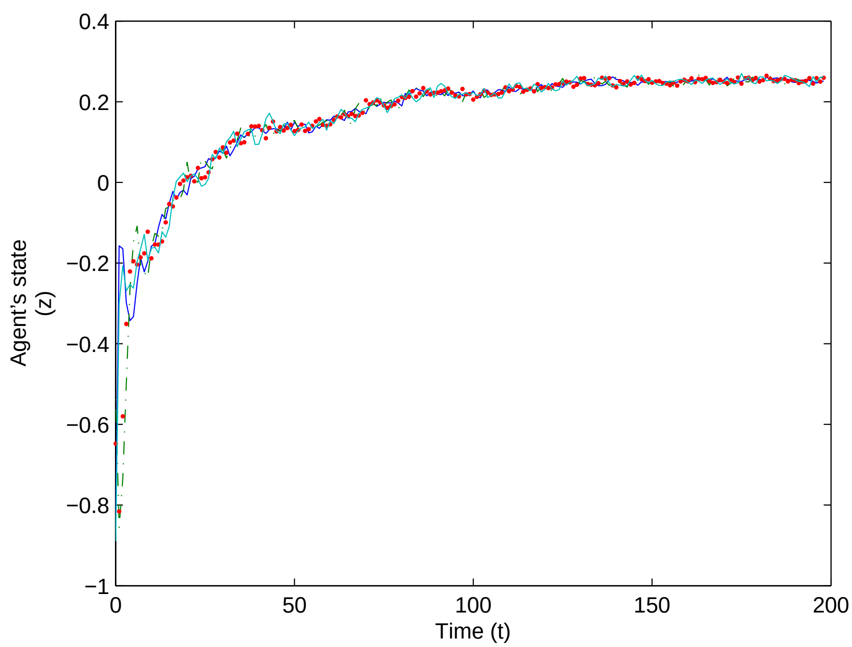

Figure 6.8: Sensors' state estimations vs. time for four of the sensors are shown. The network consists of $N = 8$ sensors communicating over an undirected graph. The ith sensor observes $q_i^t = a_i^t s + b_i^t$, where a_i^t and b_i^t are chosen at random from a uniform distribution on $[0, 2]$ and $[-\frac{1}{2}, \frac{1}{2}]$, respectively. We use the Distributed Online ADMM algorithm to estimate $\hat{s}$ which minimizes the cost function $f(s) = \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{1}{2}(q_i^t - s)^2$. The result demonstrates consensus among sensors.

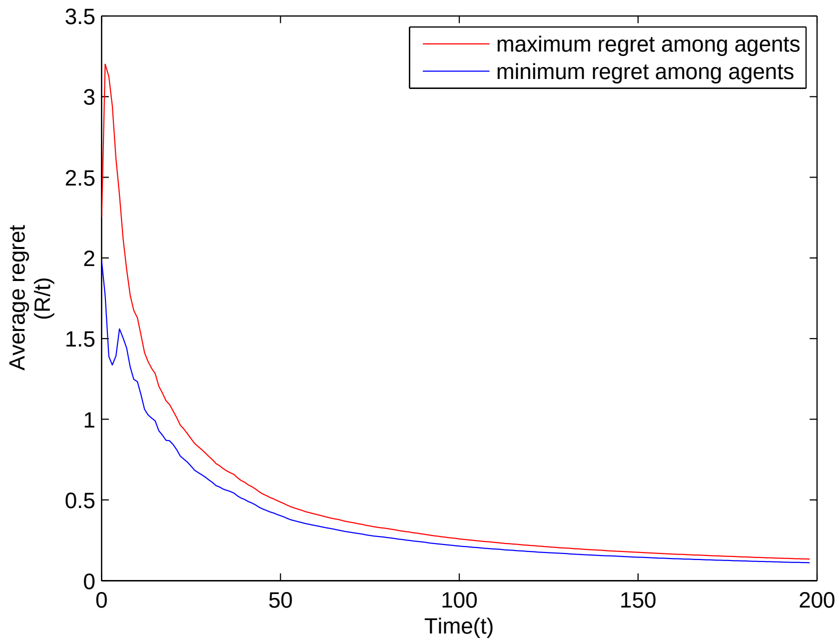

Figure 6.9: Average regrets over time $\mathsf{R}^j(T)/T$ vs. T for two sensors with the maximum and minimum average regrets are shown, under the same assumptions as in Figure 6.8.

Chapter 7

Conclusions and future work

7.1 Summary

In this thesis, after describing the problem of distributed online convex optimization, we introduced two classes of distributed online algorithms which achieve sublinear individual regrets.

In the first algorithm, which is a subgradient push-sum discrete-time algorithm, 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 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 showed 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.

Our second algorithm is a distributed online version of the Alternating Direction

Method of Multipliers. We showed that by choosing proper parameters, we can bound the individual regret by $\mathcal{O}(\sqrt{T})$. This bound is similar to the one for subgradient flow protocols; however, our algorithm exhibits an *explicit* dependency of the regret bound on the size of the network. Unlike the existing Distributed Online ADMM algorithms, our proposed algorithm is subgradient free, and our regret bounds are valid for the individual regret, in addition to the network regret.

7.2 Future research directions

It will be interesting to study the performance of the proposed algorithms on random graphs, along the lines of what is done for distributed averaging on sequence of random graphs [32]. One can improve the regret bound of the algorithms by investigating other procedures or considering other assumptions, for example, the regret bound of the subgradient push-sum algorithm on general convex functions. A challenging problem is studying problems with constraint. Distributed online optimization also has applications to the theory of learning in games, and a distributed version of such learning protocols might be feasible by what is presented in this thesis [8]. Studying the problem of failure, and network topology design for achieving faster convergence rates [12] are among the avenues for future work.

Bibliography

[1] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 1st edition, 2003.

[2] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 2996–3000, Seville, Spain, December 2005.

[3] S. Boyd, N. Parikh, E. Chu ad B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[4] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[5] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[6] 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*, pages 3688–3693, Hawaii, USA, December 2012.

[7] J. W. Durham, A. Franchi, and F. Bullo. Distributed pursuit-evasion without global localization via local frontiers. *Autonomous Robots*, 32(1):81–95, 2012.

[8] Eyal Even-Dar, Yishay Mansour, and Uri Nadav. On the convergence of regret minimization dynamics in concave games. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 523–532. ACM, 2009.

[9] B. Gharesifard and J. Cortés. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, 59(3):781–786, 2014.

[10] Osman Güler. *Foundations of optimization*, volume 258. Springer Science & Business Media, 2010.

[11] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[12] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulations. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[13] S. Hosseini, A. Chapman, and M. Mesbahi. Online distributed optimization via dual averaging. In *IEEE 52nd Annual Conference on Decision and Control*, pages 1484–1489. IEEE, 2013.

[14] S. Hosseini, A. Chapman, and M. Mesbahi. Online distributed optimization on dynamic networks. 2014. http://arxiv.org/pdf/1412.7215.pdf.

[15] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi. Online distributed ADMM on networks. *arXiv preprint arXiv:1412.7116*, 2014.

[16] 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*, pages 276–286, Reston, VA, 1998.

[17] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

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

[19] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *IEEE Symposium on Foundations of Computer Science*, pages 482–491, Washington, DC, October 2003.

[20] 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, pages 240–246, New York, NY, USA, 1986. ACM.

[21] D. Mateos-Nunez and J. Cortes. Distributed online convex optimization over jointly connected digraphs. *IEEE Transactions on Network Science and Engineering*, 1(1):23–37, Jan 2014.

[22] 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*, pages 15–22, Groningen, The Netherlands, 2014.

[23] A. Nedic and A. Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. 2014. http://arxiv.org/abs/1406.2075v1.

[24] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. Technical report, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, November 2007.

[25] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[26] A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[27] Angelia Nedic and Alex Olshevsky. Distributed optimization over time-varying directed graphs. In *IEEE Conf. on Decision and Control*, pages 6855–6860. IEEE, 2013.

[28] Angelia Nedic and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.

[29] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *Symposium on Information Processing of Sensor Networks*, pages 20–27, Berkeley, CA, April 2004.

[30] M. Raginsky, N. Kiarashi, and R. Willett. Decentralized online convex programming with local information. In *American Control Conference*, pages 5363–5369, San Francisco, CA, 2011.

[31] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[32] B. Touri. *Product of random stochastic matrices and distributed averaging*. PhD thesis, University of Illinois at Urbana-Champaign, 2011.

[33] K. I. Tsianos, S. Lawlor, and M. G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *IEEE Conf. on Decision and Control*, pages 5453–5458, Maui, HI, 2012.

[34] K. I. Tsianos and M. G. Rabbat. Distributed strongly convex optimization. 2012. arXiv:1207.3031v2.

[35] 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.

[36] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

[37] P. Wan and M. D. Lemmon. Event-triggered distributed optimization in sensor networks. In *Symposium on Information Processing of Sensor Networks*, pages 49–60, San Francisco, CA, 2009.

[38] H. Wang and A. Banerjee. Online alternating direction method. In *International Conference on Machine Learning*, pages 1119–1126, Edinburgh, Scotland, July 2012.

[39] Huahua Wang and Arindam Banerjee. Online alternating direction method (longer version). *arXiv preprint arXiv:1306.3721*, 2013.

[40] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *IEEE Conf. on Decision and Control*, pages 5445–5450, Maui, HI, 2012.

[41] Ermin Wei and Asuman Ozdaglar. On the $\mathcal{O}(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers. *arXiv preprint arXiv:1307.8254*, 2013.

[42] J. Wolfowitz. Product of indecomposable, aperiodic, stochastic matrices. *Proceedings of American Mathematical Society*, 14(5):733–737, 1963.

[43] Feng Yan, Shreyas Sundaram, SVN Vishwanathan, and Yuan Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *Knowledge and Data Engineering, IEEE Transactions on*, 25(11):2483–2493, 2013.

[44] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 928–936, Washighton, D.C., 2003.