

ABSTRACT

Title of dissertation: CONSENSUS, PREDICTION AND
OPTIMIZATION IN
DIRECTED NETWORKS

Van Sy Mai, Doctor of Philosophy, 2017

Dissertation directed by: Professor Eyad H. Abed
Dept. Electrical and Computer Engineering

This dissertation develops theory and algorithms for distributed consensus in multi-agent networks. The models considered are opinion dynamics models based on the well known DeGroot model. We study the following three related topics: consensus of networks with leaders, consensus prediction, and distributed optimization.

First, we revisit the problem of agreement seeking in a weighted directed network in the presence of leaders. We develop new sufficient conditions that are weaker than existing conditions for guaranteeing consensus for both fixed and switching network topologies, emphasizing the importance not only of persistent connectivity between the leader and the followers but also of the strength of the connections. We then study the problem of a leader aiming to maximize its influence on the opinions of the network agents through targeted connection with a limited number of agents, possibly in the presence of another leader having a competing opinion. We reveal fundamental properties of leader influence defined in terms of either the transient

behavior or the achieved steady state opinions of the network agents. In particular, not only is the degree of this influence a supermodular set function, but its continuous relaxation is also convex for any strongly connected directed network. These results pave the way for developing efficient approximation algorithms admitting certain quality certifications, which when combined can provide effective tools and better analysis for optimal influence spreading in large networks.

Second, we introduce and investigate problems of network monitoring and consensus prediction. Here, an observer, without exact knowledge of the network, seeks to determine in the shortest possible time the asymptotic agreement value by monitoring a subset of the agents. We uncover a fundamental limit on the minimum required monitoring time for the case of a single observed node, and analyze the case of multiple observed nodes. We provide conditions for achieving the limit in the former case and develop algorithms toward achieving conjectured bounds in the latter through local observation and local computation.

Third, we study a distributed optimization problem where a network of agents seeks to minimize the sum of the agents' individual objective functions while each agent may be associated with a separate local constraint. We develop new distributed algorithms for solving this problem. In these algorithms, consensus prediction is employed as a means to achieve fast convergence rates, possibly in finite time. An advantage of our distributed optimization algorithms is that they work under milder assumptions on the network weight matrix than are commonly assumed in the literature. Most distributed algorithms require undirected networks. Consensus-based algorithms can apply to directed networks under an assumption

that the network weight matrix is doubly stochastic (i.e., both row stochastic and column stochastic), or in some recent literature only column stochastic. Our algorithms work for directed networks and only require row stochasticity, a mild assumption. Doubly stochastic or column stochastic weight matrices can be hard to arrange locally, especially in broadcast-based communication. We achieve the simplification to the row stochastic assumption through a distributed rescaling technique. Next, we develop a unified convergence analysis of a distributed projected subgradient algorithm and its variation that can be applied to both unconstrained and constrained problems without assuming boundedness or commonality of the local constraint sets.

CONSENSUS, PREDICTION AND OPTIMIZATION
IN DIRECTED NETWORKS

by

Van Sy Mai

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor Eyad H. Abed, Chair/Advisor
Professor Richard J. La
Professor P. S. Krishnaprasad
Professor André L. Tits
Associate Professor Nikhil Chopra

© Copyright by
Van Sy Mai
2017

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Eyad Abed for his invaluable guidance and unstinting support over the past five years. It has been a honor to work with and learn from him, without whom this thesis would not have been possible.

I am also grateful to Professor P. S. Krishnaprasad, Professor Richard La, Professor André Tits and Professor Nikhil Chopra for agreeing to serve on my thesis committee and providing me with insightful comments and suggestions to broaden my research from various angles. I would also like to thank my Master's thesis advisor, Professor Suchin Arunsawatwong, for encouraging me to pursuit a PhD degree and helping me with the application process.

Throughout my years at the University of Maryland, I have been fortunate to have many wonderful friends who deserve a special mention. I would like to thank Dzung Ta, Sanmeet Narula and Devon Harbaugh for their help and company during the first year of my graduate life in the US. Many thanks also go to my office-mates Bhaskar Ramasubramanian, Alborz Alavian, and James Ferlez for their friendship, feedback and support. My interaction with Dipankar Maity has been very fruitful and he deserves special thanks. My days in Maryland would not have been enjoyable without my Vietnamese friends, including Chanh Kieu, My Le, Khoa Trinh, and especially the Sean Lam's family, and I would like to thank them all.

I would like to acknowledge the financial support from the Air Force Office of Scientific Research through MURI AFOSR Grant #FA9550-09-1-0538 and the

Department of Electrical and Computer Engineering at the University of Maryland.

Last but never least, I am most grateful to my family: my parents and brother for having always stood by and believed in me through my whole life, and especially my wonderful and loving wife and our beloved daughter for their incredible support, endless patience and unconditional encouragement. Words, by no means, can express the gratitude I owe them.

Table of Contents

| | |
|---|------|
| List of Tables | viii |
| List of Figures | ix |
| List of Abbreviations | xi |
| 1 Introduction | 1 |
| 1.1 Motivation and Thesis Objectives | 2 |
| 1.1.1 Consensus and Information Sharing Model | 2 |
| 1.1.2 Network Asymmetry | 3 |
| 1.2 Main Problems and Thesis Contributions | 6 |
| 1.2.1 Network with Leaders | 7 |
| 1.2.2 Consensus Prediction | 9 |
| 1.2.3 Distributed Optimization | 11 |
| 1.3 Literature Survey | 12 |
| 1.3.1 Consensus in Networks with Leaders | 14 |
| 1.3.2 Consensus Prediction | 17 |
| 1.3.3 Distributed Optimization | 18 |
| 1.4 Thesis Organization | 20 |
| 1.5 Notation and Mathematical Background | 22 |
| 1.5.1 Notation and Definitions | 22 |
| 1.5.2 Convergence of DeGroot Model | 24 |
| I Consensus Network with Leaders | 27 |
| 2 Opinion Dynamics with Persistent Leaders | 28 |
| 2.1 Introduction | 28 |
| 2.2 Problem Formulation | 30 |
| 2.3 Opinion Dynamics with One Leader | 32 |
| 2.4 Opinion Dynamics with Two Leaders | 46 |
| 2.5 Conclusion and Extensions | 48 |

| | | |
|---------|---|-----|
| 3 | Optimizing Leader Influence in Networks through Selection of Direct Followers | 51 |
| 3.1 | Introduction | 52 |
| 3.2 | Problem Formulation and Related Works | 56 |
| 3.2.1 | Formulation of Influence Optimization Problem for the Single Leader Case | 57 |
| 3.2.2 | Formulation of Influence Optimization Problem in the Presence of a Competing Leader | 63 |
| 3.2.3 | Comparison to Previous Work | 66 |
| 3.2.3.1 | Single leader case | 66 |
| 3.2.3.2 | Multiple leaders case | 68 |
| 3.2.3.3 | Our Contributions | 70 |
| 3.3 | Special Cases $K = 1, 2$: Optimal Solutions | 71 |
| 3.3.1 | Single Agent Selection | 72 |
| 3.3.2 | Two-Agent Selection | 75 |
| 3.4 | General Case: Convexification Approach | 80 |
| 3.4.1 | Convexity of Relaxation | 80 |
| 3.4.2 | Numerical Methods | 90 |
| 3.5 | Supermodularity and Greedy Algorithms | 93 |
| 3.5.1 | Supermodularity Results | 93 |
| 3.5.2 | Greedy Algorithms and Ratio Bounds | 96 |
| 3.6 | Numerical Examples | 105 |
| 3.6.1 | Example 1: Small Network with One Leader | 105 |
| 3.6.2 | Example 2: Medium-Size Network with Two Leaders | 108 |
| 3.7 | Closing Discussion | 110 |
| 3.7.1 | Application to Friedkin's Model | 111 |
| 3.7.2 | Further Convexity Results | 112 |
| 3.7.3 | Towards Relaxing Strong Connectivity Assumption | 113 |
| II | Consensus Prediction by Observer | 114 |
| 4 | Consensus Prediction in Minimum Time | 115 |
| 4.1 | Introduction | 116 |
| 4.2 | Problem Statement and Previous Results | 117 |
| 4.2.1 | Problem Description | 117 |
| 4.2.2 | Previous Results on Consensus in Finite Time | 119 |
| 4.3 | Shortest Time Prediction of Consensus and Local Computation of Minimal Polynomials | 122 |
| 4.3.1 | Optimality of $(D_i + 1)$ | 123 |
| 4.3.2 | Local Computation of q_i | 126 |
| 4.4 | Toward Minimizing Observation Time | 128 |
| 4.4.1 | Observed Nodes with Identical Minimal Polynomials | 129 |
| 4.4.2 | Observed Nodes with Different Minimal Polynomials | 133 |
| 4.5 | Numerical Examples | 134 |

| | | |
|------------------------------|--|-----|
| 4.5.1 | Example 1: Network with Identical Minimal Polynomials . . . | 134 |
| 4.5.2 | Example 2: Network with Different Minimal Polynomials . . . | 135 |
| 4.6 | Toward Selecting Observed Nodes | 137 |
| 4.6.1 | When $q_i = q_j$? | 140 |
| 4.6.2 | Bounds on $\deg(q_i)$ | 141 |
| 4.7 | Limitations and Future Work | 143 |
| III Distributed Optimization | | 144 |
| 5 | Local Prediction for Enhanced Convergence of Distributed Optimization Algorithms | 145 |
| 5.1 | Introduction | 146 |
| 5.2 | Problem Statement and Background | 150 |
| 5.2.1 | Problem Statement | 150 |
| 5.2.2 | Subgradient Methods | 151 |
| 5.2.3 | Finite-Time Consensus Using Minimal Polynomials | 153 |
| 5.3 | Distributed Subgradient Optimization Using Finite Time Consensus . | 155 |
| 5.3.1 | Main Algorithm | 156 |
| 5.3.2 | Extensions of the Algorithm 5.1 | 165 |
| 5.4 | Finite-Time Optimization for Quadratic Cost Functions | 168 |
| 5.4.1 | Ratio-Consensus based Algorithm | 169 |
| 5.4.2 | Gradient-based Algorithm | 170 |
| 5.5 | On Minimal Value of κ and Performance Limits of Distributed Subgradient Methods | 177 |
| 5.5.1 | Minimal Value of κ | 177 |
| 5.5.2 | Performance Limit of Distributed Subgradient Methods | 179 |
| 5.6 | Simulations | 181 |
| 5.6.1 | Example 1: Network of 5 agents with differentiable cost functions having Lipschitz continuous gradient | 181 |
| 5.6.2 | Example 2: Network of 200 agents with ℓ_1 cost functions | 185 |
| 5.7 | Concluding Remarks | 188 |
| 6 | Distributed Optimization over Directed Graphs with Row Stochasticity and Constraint Regularity | 191 |
| 6.1 | Introduction | 192 |
| 6.2 | Problem Formulation and Proposed Algorithms | 197 |
| 6.3 | Basic Relations and Convergence Result | 203 |
| 6.4 | Rate of Convergence | 219 |
| 6.5 | Numerical Example | 227 |
| 6.6 | Conclusions and Extensions | 229 |
| 7 | Conclusions | 233 |
| 7.1 | Summary of Results | 233 |
| 7.2 | Directions for Future Work | 236 |

| | | |
|-------|--|-----|
| A | Omitted Proofs | 240 |
| A.1 | Known Matrix Results | 240 |
| A.2 | Omitted Proofs in Chapter 3 | 241 |
| A.2.1 | Proof of Theorem 3.3.1 | 241 |
| A.2.2 | Proof of Theorem 3.3.2 | 242 |
| A.2.3 | Proof of Theorem 3.3.4 | 243 |
| A.2.4 | Proof of Lemma 3.5.3 | 244 |
| A.2.5 | Proof of Lemma 3.5.5 | 246 |
| A.3 | Omitted Proofs in Chapter 5 | 248 |
| A.3.1 | Proof of Theorem 5.3.5 | 248 |
| A.3.2 | Proof of Theorem 5.3.12 | 250 |
| A.3.3 | Proof of Theorem 5.3.14 | 251 |
| A.3.4 | Proof of Extension to Row Stochastic Weight Matrix | 252 |
| A.3.5 | Proof of Theorem 5.5.1 | 252 |
| A.3.6 | Proof of Lemma 5.4.3 | 253 |
| A.3.7 | Proof of Theorem 5.4.4 | 255 |
| A.3.8 | Proof of Theorem 5.4.5 | 256 |
| A.3.9 | Distributed Evaluation of Global Cost Function and Algorithm Local Termination | 257 |
| A.4 | Omitted Proofs in Chapter 6 | 259 |
| A.4.1 | Proof of Theorem 6.3.6 for Algorithm 6.2. | 259 |
| | Bibliography | 261 |

List of Tables

| | | |
|-----|--|-----|
| 3.1 | Comparison results for Network in example 1 (* denotes an optimal value). In the last column, $J_{\mathcal{K}_1^{P-Rlxd}}$ denotes $J_{\mathcal{K}_1^{P-Rlxd}}(J_{\mathcal{K}_2^{P-Rlxd}})$ | 107 |
| 4.1 | Observation times using Algorithm 4.1. | 135 |
| 4.2 | Observation time for each node to compute consensus value in Example 2 | 136 |
| 4.3 | Optimal time T^* when the observer can choose any m nodes | 137 |

List of Figures

| | | |
|-----|---|-----|
| 1.1 | A directed network of 5 agents. | 5 |
| 1.2 | A network in the presence of 2 leaders T and Q | 7 |
| 1.3 | A network with an observer. | 9 |
| 1.4 | A network of five agents that try to minimize the sum of individual cost functions. | 11 |
| | | |
| 3.1 | Network in example 1. | 106 |
| 3.2 | $\mathcal{K}^* = \{7, 8, 15, 25\}$ | 108 |
| 3.3 | $\mathcal{K} = \{7, 13, 16, 25\}$ | 108 |
| 3.4 | $\mathcal{K} = \{8, 13, 16, 25\}$ | 108 |
| 3.5 | Alg. 3.1 every 5 time steps | 108 |
| 3.6 | Upper bounds (solid lines) and lower bounds (dashed line) on J^* ; The global lower bound $J(\mathcal{V}_\alpha)$ holds for any K . The ratio bound $\frac{1-J_{GU}}{1-J_{P.Rlx d}^*}$ (shown by a dotted line) is at least 90% as $K \geq 90$ | 110 |
| 3.7 | CPU run times (s) in 4 schemes. The Interior Point Method takes approximately 0.21 s per iteration. | 111 |
| | | |
| 4.1 | Network example 2. Self weights are not shown. | 136 |
| | | |
| 5.1 | Network topology in example 1. | 181 |
| 5.2 | Network responses for example 1 with convex cost functions having Lipschitz continuous gradient using Algorithm (5.32) and (5.37). <i>Left:</i> For any $i \in \mathcal{V}$, $s_i(t)$ (solid lines) converges to optimal solution (dash-line) and $x_i(t)$ reaches a limit cycle of period κ . In the top-left figure, \circ represents $\bar{s}(k\kappa)$ of the centralized subgradient method implemented as (5.19). <i>Right:</i> Objective error comparisons with DPS using step size $\gamma(t) = \frac{a}{tb}$, where (blue) solid lines correspond to $a = 0.01$, (green) dashed lines $a = 0.05$, (black) dotted lines $a = 0.1$, and (cyan) dash-dotted ones $a = 0.2$. For each a , we plot the results for $b = 0.5$ and 1. The results from our algorithm are shown in red circles \circ . The algorithm terminates locally for all the agents at $t = 186$ with relative error of the global cost function guaranteed to be less than $\epsilon = 10^{-6}$ | 186 |

| | | |
|-----|--|-----|
| 5.3 | Network responses for example 1 with quadratic cost functions when using Algorithm 5.3 with $\kappa = 7$, $\mathbf{x}(0) = \mathbf{c}$, and with 4 values of γ | 187 |
| 5.4 | Responses of the network in example 2. Dashed line: optimal solution. (a)-(b): Algorithm (5.32)-(5.33), where sub-figure within (a) is a zoom-in of period [400, 800]; (c): Algorithm by Olshevsky (2016) with a constant step size $\beta = \frac{1}{L_0\sqrt{NT}}$; and (d): Distributed Subgradient Method (5.4) with $\gamma(t) = \frac{1}{\sqrt{t}}$ | 189 |
| 6.1 | Directed communication graph of the network example. | 228 |
| 6.2 | Performances of Algorithms 6.1, 6.2, and DSP methods with and without reweighting technique. Reweighting means for each $i \in \mathcal{V}$, π_i is known to agent i in advance and $z_{ii}(t) = \pi_i, \forall t \geq 0$. Here, $\mathbf{s}(t) = P_X(\bar{\mathbf{x}}(t))$ | 230 |

List of Symbols and Abbreviations

| | |
|----------------------------|---|
| \mathcal{E} | Edge set of a graph |
| \mathcal{G} | Communication Graph |
| L | Laplacian matrix |
| N | Number of agents in the network |
| \mathbb{N}_0 | Set of all nonnegative integer numbers |
| $\mathbb{R}(\mathbb{R}_+)$ | Set of all (nonnegative) real numbers |
| t | Discrete time |
| \mathcal{V} | Node set of a graph |
| $W, [w_{ij}]$ | Weight matrix |
| $x_i(t), \mathbf{x}_i(t)$ | Opinion or state vector of agent i at time t |
| $\mathbb{Z}(\mathbb{Z}_+)$ | Set of all (nonnegative) integer numbers |
| $\boldsymbol{\pi}$ | The normalized left Perron eigenvector of the weight matrix W |
| | |
| AEP | Almost Equitable Partition |
| DPS | Distributed Projected Subgradient |
| DSM | Distributed Subgradient Method |
| IPM | Interior Point Method |
| LCM | Least Common Multiple |
| LTI | Linear Time-Invariant |
| PGM | Projected Gradient Method |

Chapter 1: Introduction

Networks are ubiquitous in physical, biological, and engineered systems. Depending on the particular domain and on the network nodes and their interconnections, networks can display interesting characteristics and can achieve a variety of functions. Networks research has seen significantly increasing interest over the past several decades, owing mainly to the realization that applications are wide-ranging and that these applications can prove to be both practical and valuable for society. While a large body of literature has arisen, our understanding of the characteristics, features and dynamics of networks is still in the early stages of development.

In this thesis, we aim to contribute to this important and growing field by pursuing several directions in the general area of network consensus. Among the topics we pursue is the development of new conditions and algorithms for reaching and/or predicting agreement among agents in a network. To this end, a number of theories will be brought to bear on several problems of interest under various scenarios with regard to network connectivity. We also introduce and study problems of network monitoring and consensus prediction, and apply our results to distributed optimization.

1.1 Motivation and Thesis Objectives

1.1.1 Consensus and Information Sharing Model

There has been much interest in problems of distributed computation and cooperative control, where a group of agents aims to achieve a global objective without resorting to a centralized coordination entity and possibly in the presence of limited computing capability and/or energy resources. In this realm, network consensus is a basic problem, which concerns processes by which a collection of agents through their local interactions tries to reach a common goal or decision. This problem has been studied extensively in recent years and has found applications in many areas such as opinion dynamics and learning in social networks, distributed optimization, multi-vehicle rendezvous, formation control and sensor fusion, to name a few (see, e.g., [1–13]). Extensive surveys and tutorials can be found in [14–16].

Historically, consensus in one form or another has long been observed in both natural and social networks and systems. For example, a flock of birds flies in a certain shape with a common velocity, a swarm of fireflies blinks in unison, a group of people achieves agreement after repeatedly exchanging opinions with one another. The discovery of network consensus can be traced to circa 1665 when Christiaan Huygens observed the synchrony of two pendulum clocks mounted next to each other on the same support, a phenomenon now referred to as *coupled oscillations* [17]. Thus, studying mechanisms for network consensus is key to understanding collective behaviors of both natural and social systems, and to building engineering

networks as well.

Many recent efforts have also been devoted to the case of networks with more than one type of agents, including, e.g., leaders and followers, stubborn or even adversarial agents, which appear naturally in real world networks and systems (see, e.g., [5, 12, 13, 18–23] and references therein). The notion of leader is also useful in the study of control of networked systems, where leaders serve as agents that directly receive control inputs and network connections are paths for control action propagating to the other agents [24]. Network controllability and its dual notion, observability, have also been studied extensively in recent years [25–31]. However, a closely related problem, namely network consensus prediction has scarcely been considered. In this thesis, we will investigate this problem in detail.

The idea behind consensus can also serve as a mechanism for information sharing/diffusing in the design of many distributed multi-agents algorithms, including distributed optimization where a group of agents with limited communication tries to solve a global optimization problem. This problem arises in many applications such as distributed estimation in sensor networks [32–34], distributed resource allocation [35, 36], and large-scale machine learning and statistical inference [37–39], and is becoming more urgent in the new era of “Big Data”.

1.1.2 Network Asymmetry

For distributed systems, communication is vital to system performance as it is the backbone for information to flow from one agent to another, and hence, the only

platform for each agent to contribute to or get involved in the global objective of the network. As a result, special attention for communications is needed in the study and design of distributed systems; in fact, it is one of the main aspects that is different from centralized ones.

In general, communications can be categorized as undirected or directed. In many applications, it is possible that inter-agent communications are undirected, i.e., when two individuals communicate, each receives information from the other, and moreover, they can have some agreement on how to use that information. However, there are many other scenarios where communications are directed due to, e.g., communication constraints arising from various sources, including physical network connections or hardware capability of system components. This is clearly the case when we consider the effect of a leader or stubborn agent in the network. Another practical example is an *ad hoc* Wireless Sensor Network, where there may not be a pre-existing communication infrastructure at the time of deployment and directed communications can arise as a consequence of the geometric network layout or nonuniform transmission power limits, or even sensor mobility. We identify directed communication as a type of network asymmetry; see Figure 1.1 for an example of a directed network with 5 agents.

Although directed communication schemes include undirected schemes as a special case (and thus apply to a much larger range of situations), a large body of literature in the field of consensus and distributed computation and optimization focuses on undirected networks. Such networks are more amenable to mathematical analysis, especially when using tools that assume network symmetry (such as sym-

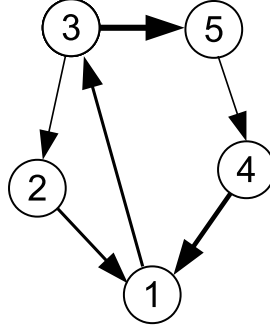


Figure 1.1: A directed network of 5 agents.

metric nonnegative Laplacian matrices, for instance). Analysis of algorithms and problems with directed communications is generally very involved and often requires new tools and techniques. *We seek to develop algorithms that apply to networks with directed communication schemes as the prime subject of this thesis.* In particular, we are interested in scenarios where network asymmetry poses challenges in the analysis and may hinder system performance.

Among various models in the literature, the DeGroot model [1] is one of the most often used for its simplicity and ability to exhibit consensus behaviors. Here, each agent in the network repeatedly updates its opinion as a weighted average of the opinions of its immediate neighbors, including itself. This update scheme simply gives rise to a row stochastic weight matrix, which under mild conditions guarantees consensus in the network. The agreement value depends on this weight matrix and the agents' initial opinions. However, in many situations including various distributed optimization algorithms, an averaging scheme is desired instead; and to this end, the weight matrix is often required to satisfy a balancedness condition, namely, double stochasticity. In such a case, this condition means any agent needs

to know to whom it sends information and/or regulates the way the recipient uses the information. This can be ensured fairly easily and locally in undirected networks but is very hard and costly to ensure in many distributed systems such as wireless sensor/ad-hoc networks and especially networks with broadcast-based communications. A few algorithms proposed recently for directed networks employ column stochastic matrices by requiring that each agent at any time is aware of which neighbors receive the information sent to them by the agent. A row stochastic matrix is much easier to implement in these applications/communication environments, but yields unsatisfactory performance when used with those algorithms. Therefore, in this thesis, we will mostly deal with row-stochastic matrices as the main source of network asymmetry.

1.2 Main Problems and Thesis Contributions

In this research, we consider the following three main topics and the associated problems identified below. Unlike most existing work in the literature where network symmetry or balancedness is assumed, we address the general case of weighted directed graphs. Thus, one of the main technical contributions that we emphasize is a set of tools and techniques developed to overcome network asymmetry in various problems and applications of consensus.

1.2.1 Network with Leaders

In the first part of the thesis, we consider a DeGroot model with the presence of external media nodes, representing leaders, or sources of news often having constant opinion values. See Figure 1.2 for an illustrative example.

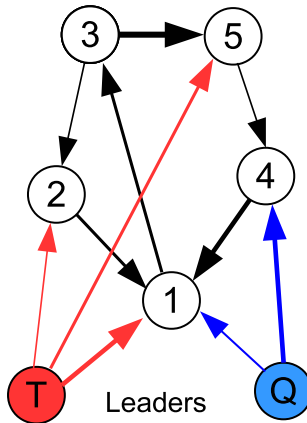


Figure 1.2: A network in the presence of 2 leaders T and Q .

First, when consensus is the main goal of a leader, we are interested in finding conditions under which the whole network will eventually agree with that leader's opinion for any initial opinions of the agents. Indeed, we will determine how strong the connections between the leaders and the followers as well as those among the followers should be to ensure that this agreement can be achieved asymptotically.

- When there is only one leader, we derive new sufficient conditions for guaranteeing consensus with the leader for both fixed and switching topologies. These conditions emphasize the persistence of the connectivity between the leader and the followers and are the mildest so far, covering many existing results in the literature.

- In the presence of more than one leader, we show that only those that are persistent matter. In particular, when only one leader is persistent, we provide conditions under which the network converges to the state of this leader.
- A technical contribution lies in the tool that we develop to prove the results above; namely, a result on the convergence of a infinite product of nonnegative substochastic matrices.

Second, we study the problem of a leader that aims to influence the opinions of agents in a directed network through connecting with a limited number of the agents. The leader's goal is to select this set of agents, referred to as direct followers, to achieve the greatest possible influence on the opinions of agents throughout the network. Here, when there is only one leader and consensus is guaranteed a priori, the influence of that leader is characterized through the transient error of the network, and thus is able to take into account both the network structure and the opinion dynamics evolving on it. When, on the other hand, there is a second leader (or a stubborn agent) with a competing opinion and consensus is not achievable, the influence of the first leader is measured in terms of the steady state error of the network. Compared to existing work, not only are our problem settings and formulations more natural (and thus likely to be of more value for practical applications) but our technical results are also much stronger. In particular,

- We prove the supermodularity property of the objective function capturing the leader influence in both cases and the convexity of its continuous relaxation for general directed networks. Here, the convexity result is novel; the super-

modularity result generalizes existing results in the literature but is proved using a different technique.

- We then develop greedy algorithms that are theoretically guaranteed to have a lower bound on the approximation ratio. The new convex result allows us to benefit from efficient (customized) numerical solvers to obtain practically comparable solutions. We demonstrate through numerical examples that the two approaches can be combined to provide effective tools and better analysis for optimal design of influence spreading in diffusive networks.

1.2.2 Consensus Prediction

In this part of the dissertation, we introduce and study the problem of consensus prediction in a network whose dynamics is described by a DeGroot model. In particular, we assume that there is an observer who can monitor the states of a group of agents, but might not have accurate knowledge of the underlying communication graph and the associated weight matrix; see Figure 1.3.

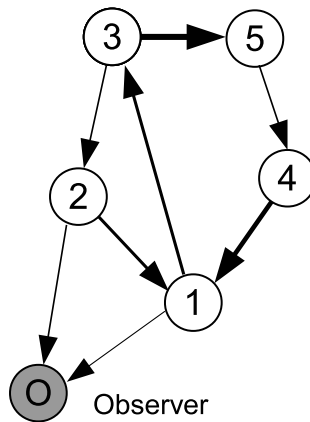


Figure 1.3: A network with an observer.

We want to answer the following questions: For any initial opinions of the agents, how can the observer determine the consensus value, if it exists, by using a finite number of observations? what is the minimum number of observations needed? And, if the observer has more information about the network structure, how to minimize observation time over possible choices of observed nodes. Our main contributions in this topic are as follows:

- We reveal an intrinsic relation between the consensus value and network data, namely, if the consensus value can be computed at a particular time for any initial opinions, then it can be expressed as a linear combination of available observation data with associated coefficients depending on the weight matrix.
- We derive a fundamental limit on the monitoring time for the case of a single observed node, below which the observer with limited knowledge about the network is not able to determine the consensus value regardless of the method used. We provide sufficient conditions for achieving this limit.
- We provide a conjecture and analysis for the case of multiple observed nodes and develop algorithms toward achieving the conjectured bounds through local observations and computations. We show that with certain knowledge about the network structure, the observer can answer a few questions regarding the optimal monitoring time.

1.2.3 Distributed Optimization

In our work on distributed optimization, we seek interaction rules for a network of agents which result in the network collaboratively solving a global optimization problem. The interactions among agents must be local, without a central coordination unit, and the objective function is the sum of local costs of all the agents; see Figure 1.4.

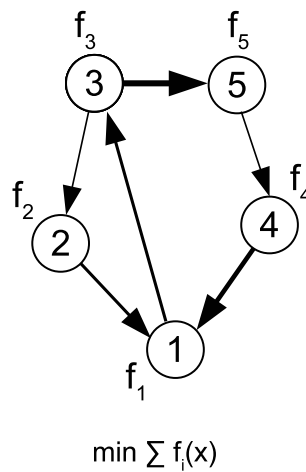


Figure 1.4: A network of five agents that try to minimize the sum of individual cost functions.

Under the conditions that the underlying communication graph is directed and the weight matrix is only row stochastic, we design algorithms for the agents to collaboratively solve this problem in a distributed manner and/or with fast convergence rates.

- We first study the use of consensus prediction for enhancing convergence of distributed optimization algorithms. The resulting algorithms are the first

that possess the following useful features: (i) they are distributed but behave similarly to the centralized gradient methods except on a slower time-scale (including finite time convergence for quadratic cost functions), (ii) all the agents are able to locally stop updating at the same time with the same estimate of optimal solution, and (iii) the theoretical convergence scales at most linearly with a network size in general (and thus is the best so far).

- We provide a unified analysis for distributed projected subgradient methods with nonidentical local constraint sets. To deal with network asymmetry, we introduce a rescaling technique to the original distributed projected subgradient methods by incorporating an additional consensus step which aims to provide each agent with an estimate of the corresponding element of the left Perron eigenvector of the weight matrix.
- We present another algorithm that also uses the rescaling technique above but is able to achieve linear convergence under a stronger assumption on the local objective cost functions.

1.3 Literature Survey

One of the very first mathematical models used for studying network consensus is the DeGroot model [1], described as follows. Consider a group of N agents and let $x_i(t) \in [0, 1]$ denote the opinion of agent i at time t ; here time is discrete, i.e., $t \in \mathbb{Z}_+$. Each agent has an initial opinion $x_i(0)$. At any time $t \geq 0$, each agent observes the opinions of its neighbors and *naively* updates its own opinion according

to

$$x_i(t+1) = \sum_{j=1}^N w_{ij} x_j(t), \quad i = 1, \dots, N, \quad (1.1)$$

where w_{ij} indicates the weight that agent i places on agent j 's opinion. Here $w_{ij} \neq 0$ implies that agent i is able to obtain the opinion of agent j at time t . Thus, $W = [w_{ij}] \in \mathbb{R}^{N \times N}$ is often called the weight matrix (or trust matrix). In this regard, it is natural to represent the communication network among the agents using graphs, where x_i is the state of node i and w_{ij} the weight or strength of link (i, j) ; and for this reason, the terms *agent* and *node* are used interchangeably. The network achieves consensus if for any initial opinion $x_i(0)$, it holds that $\lim_{t \rightarrow \infty} |x_i(t) - x_j(t)| = 0, \forall i, j$. Thus, the network following (1.1) is often called a *consensus network*.

This model was introduced in [1] for the synchronous and time-invariant case, where the author studied the process of reaching agreement among a group of experts, used in [40] for studying coordination of a group of particles, and extended in [2, 41–43] for the case of asynchronous and time-varying network in the context of distributed decision making and parallel computing. Since then, a vast literature on network consensus has developed. Models that generalize or are related to (1.1) are also numerous, including, for example, agents with high order linear dynamics [44, 45], nonlinear dynamics (possibly with nonlinear coupling among agents) [40, 46–51], where consensus (in terms of the agents' outputs) is also known as *synchronization*. Convergence under various assumptions on the communication graph are also studied, including, for example, directed information flow [6, 8], link/node failure and noises [33], communication time-delays [47, 52], fixed or switching net-

work topology [5, 9], and quantization [53, 54].

In this thesis, however, we will mostly focus on the first-order linear model (1.1), which is relatively simple but instructive enough for basic study of consensus dynamics and particularly suitable for various distributed computation and optimization algorithms.

1.3.1 Consensus in Networks with Leaders

Many recent efforts have also studied networks with more than one type of agent, including, e.g., leaders and followers, stubborn or even adversarial agents (see, e.g., [5, 12, 13, 18–23]). In general, for the DeGroot model, consensus can still be achieved asymptotically with a single leader, but not if there are multiple leaders such that at least two of them are uncooperative.

In this subsection, we review known conditions for consensus in the literature, where a leader is included in the network as a special agent with constant opinion.

For the case of a fixed network with a constant weight matrix, a necessary and sufficient condition for consensus is that the graph is rooted [9]. However, such a condition is still an open question for a time-varying interaction topology. In this case, conditions for consensus are those ensuring the convergence to a rank-one matrix of an infinite product

$$\lim_{t \rightarrow \infty} W(t)W(t-1) \dots W(0),$$

where each $W(t)$ is a stochastic matrix. This is also a well studied problem in the theory of non-homogeneous Markov chains. Therefore, many results and tools in

matrix theory and Markov chain theory can be appealed to. It has been shown [9,55] that a necessary condition is that the union graph over an infinite interval is rooted. This condition though is far from being sufficient; a counterexample can be found in [8]. Therefore, to derive sufficient conditions, more assumptions on the network connectivity and the weight matrix are required. For example, the authors in [5] rely on Wolfowitz’s theorem [56] on the convergence of infinite products of stochastic matrices $W(t)$ belonging to a finite set. This condition is relaxed in [8,55,57,58] so that the matrices can belong to an infinite set. However, these works require that all the self-weights and other nonzero link weights are uniformly bounded below by a positive number and that either the weight matrix at any time has a symmetric zero/nonzero structure (i.e., the graph is undirected) or the union of the interaction graphs over any period of some fixed length is strongly connected. Similarly, the limiting behavior of products of random stochastic matrices is also studied in [59] assuming the cut-balanced property of the sequence of the matrices, which is in the same spirit of having symmetric zero/nonzero structures.

We will show that for the DeGroot model in the presence of a leader, the strong connectivity and symmetric structure condition of the weight matrix can be relaxed. However, we take a different approach, in that we develop consensus conditions directly for the model with leaders.

Besides deriving condition for guaranteeing consensus, the problem of selecting an optimal subset of agents in the network to influence is also of interest in many practical applications, often known as leader selection and optimal stubborn agent placement [20,60–67]. By considering different measures of influence or centrality in

the network, there have been various approaches to solve the associated optimization problem. For example, [68] considers the problem of minimizing the convergence rate of a consensus network. The authors show the connection between the convergence rate and the maximum distance from the leader to the followers and then apply a combinatorial optimization method to solve the problem approximately. In [69], the authors consider the problem of minimizing the total system error in a noisy network and derive systematic solutions for some special cases. In [62] the authors consider a characterization quantifying both the transient and the steady states of the agents' opinions, assuming that all the regular agents have the same initial opinions and that any direct follower replaces its own opinion by that of the leader. In [70], the authors use a continuous-time model and consider the problem of leader selection in order to minimize the convergence error, defined as the ℓ_p -norm of the distance between the followers' states and the convex hull of the leader states. By replacing the convergence error by an upper bound that is independent of the initial states of the network, [70] is able to employ a supermodular optimization approach.

Our work in this topic departs from this literature in many respects. First, we drop the assumption that the network is undirected, and only ask that the underlying network be directed. Second, we allow selected direct follower nodes (i.e., agents directly connected to the leader) to follow inter-agent dynamics like any other agents, rather than forcing them to adopt the leader's opinion instantaneously. Third, we allow the agents in the network to have different initial opinions, and the leader can assign different weights to the network agents. Finally, and more importantly, although continuous relaxation and greedy heuristics have been employed in deal-

ing with influence maximization problems, our theoretical results on convexity and supermodularity are considerably stronger than existing results. We achieve this without assuming any symmetry or resorting to the random walk theory. This not only provides a deeper understanding of diffusive processes but also can be used for a broad range of applications.

1.3.2 Consensus Prediction

The topic of consensus prediction is useful in network monitoring and security but has not received much attention. The existing literature mostly focuses on the observability problem for networked multi-agent systems. The problem we consider differs from the observability problem in the sense that we are concerned with the final value instead of trying to recover the initial conditions of all the agents. Moreover, here, the observer might not have know the network structure or the weight matrix. Our analysis builds on a recent method for reaching consensus in finite time by employing the minimal polynomial of each agent [71, 72]. This method is concerned only with consensus predictability, and does not concern optimality.

Although predicting the agreement value of a consensus network is the main goal here, our approach makes a contribution to the topic of network identification through application of realization theory [73–75] to distributed networked systems.

1.3.3 Distributed Optimization

Consensus also plays an important role in distributed optimization, where a group of agents with limited communication tries to solve a global optimization problem, where the objective function is the sum of (possibly nonsmooth) local objectives of the agents and the global constraint set is the intersection of local constraints. Tsitsiklis [2] among others pioneered research on distributed computation over networks and the interplay between network dynamics and performance of decentralized algorithms in the context of networked control. Specifically, in [2] the problem of achieving consensus in system (1.1) was studied and then used as a subroutine for performing estimation and solving a class of optimization problem in a distributed manner. In this connection, the consensus step is utilized to deal with the fact that the agents have incomplete knowledge about the optimization problem. It was this idea that has triggered the development of many consensus-based distributed algorithms, see e.g., [32, 42, 76–85] and references therein.

Well known among those is the class of distributed (sub)gradients methods, which possesses many practically desirable characteristics including its simplicity in implementation and generally weak assumptions on the local cost functions as well as the network topology. Major limitations of algorithms in this category are also well studied. First, the convergence of many algorithms depends on the choice of step size sequences. When a constant step size is used, both Distributed Gradient Descent and Distributed Subgradient methods only yield convergence to a neighborhood of the optimal solution and of the optimal value [78, 86]. This occurs even under

stronger assumptions on the local objective functions such as strong convexity and Lipschitz continuous gradients, and is thus one of the main differences between these methods and their centralized counterparts. This motivates the use of particular diminishing or adaptive step sizes to achieve asymptotic convergence. However, the convergence rate can be very slow (compared to that of the centralized method), depending on the step size sequence, whose appropriate selection is not trivial. Nesterov’s acceleration technique can be employed [87] to speed up the convergence. Second, many incremental subgradient methods require all the agents to construct a closed cycle in order to pass an estimate of the solution around the network; see e.g., [32, 88, 89]. Third, even when asymptotic convergence is guaranteed, it is not obvious how each agent can locally decide when to stop the algorithm without affecting other agents’ estimates. Put differently, there are no simple criteria for all the agents to stop at the same time while also sharing the same estimate of an optimal solution. This is also true for most (if not all) other distributed optimization methods.

When all the local cost functions are quadratic, many other consensus-based algorithms can outperform those in the subgradient class. For example, the ratio consensus method can be used to solve problem without constraints and converges exponentially [34, 90]. Based on this method, [91] proposed a Newton-Raphson-like algorithm which also converges asymptotically for a class of functions having continuous, strictly positive and bounded second derivatives, assuming a sufficiently small discretization step. Recently, much attention has also been given to decentralized Alternating Direction Method of Multipliers (ADMM) type methods with fast convergence in both theory and practice [39, 43, 92].

Most existing methods in distributed optimization (including those mentioned above) require the network to be undirected so that neighboring agents exchange information in both directions, increasing the possibility of reaching some agreement. Many methods employing subgradient and consensus steps require the weight matrix associated with the network to be column stochastic or even doubly stochastic, which may be hard to arrange in directed networks, especially in a broadcast-based communication environment. Recent reweighting technique introduced in [93, 94] allows row stochastic matrices but assumes knowledge of the graph, that is the stationary distribution of the weight matrix and the number of agents in the network. Thus, a fully distributed algorithm employing only row stochastic weight matrices has not been available in the field of distributed optimization thus far. In our work, we will develop such an algorithm. Moreover, known convergence analysis of distributed subgradient methods varies according to whether the problem is unconstrained or constrained, and whether the local constraint sets, usually compact, are identical or nonidentical. Thus, there is a lack of a unified convergence analysis for those scenarios, and hence developing such a analysis is also one of the goals of this thesis.

1.4 Thesis Organization

The remainder of the thesis is organized as follows. This introductory chapter will end with notations, definitions and mathematical background, including well known results for the DeGroot model. Our main results are presented in three parts

corresponding to the three topics.

Part I of the thesis is concerned with the influence of a leader on the opinions of the agents in a directed network whose dynamics follow the DeGroot model. Specifically, in Chapter 2, we develop various sufficient conditions for guaranteeing consensus of all the network agents to the leader opinions in many scenarios: static and dynamic network topologies, with one leader or two competing leaders. Then in Chapter 3, we are concerned with the problem of optimizing the influence of a leader on the opinions of the agents in the case of a fixed network topology. We derive various joint centrality measures for a group of followers in different settings, and then develop theory and approximation algorithms for obtaining suboptimal solutions in large networks.

Part II, which consists of Chapter 4, deals with problems related to an observer that seeks to predict the consensus value of a network by monitoring the opinions of a group of agents. This setting can be seen as the dual to that in Part I, where the leader injects information/control into the network. We make use of a central tool in functional analysis, namely the Hahn-Banach Theorem, to prove the optimality aspect of the degree of the minimal polynomial of a node as a tight lower bound of the observation time over all possible approaches to determine the consensus value if only that node is monitored. We then develop analysis and distributed algorithms for the case of multiple observed nodes. We also discuss optimal selection of observed nodes using graph theory.

Part III of the thesis is concerned with distributed optimization. In Chapter 5, we employ the consensus prediction method presented in Chapter 4 as an acceler-

ation technique for enhancing convergence of the distributed gradient method in terms of correctness and speed. In the special case where the local objective functions are quadratic, we show that finite time convergence can be achieved. We also discuss a performance limit of distributed optimization and compare it with our algorithms. In Chapter 6, we introduce new technique that enables many distributed optimization algorithms to work with directed networks and row stochastic weight matrices. We then develop a unified analysis for convergence as well as convergence rate of a distributed subgradient algorithm and its variation that can be applied to both unconstrained problems and constrained ones possibly with nonidentical and unbounded local constraint sets.

Finally, conclusions and directions for future research are given in Chapter 7.

1.5 Notation and Mathematical Background

1.5.1 Notation and Definitions

Notation:

We use boldface characters and symbols to denote vectors, for example, $\mathbf{x} = [x_1, x_2, \dots, x_m]^T \in \mathbb{R}^m$, $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T \in \mathbb{R}^m$, $\mathbf{1} = [1, 1, \dots, 1]^T$ and $\mathbf{e}_i = [0, \dots, 0, 1_i, 0, \dots, 0]^T$. For a vector \mathbf{x} , $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$ (or just $\|\mathbf{x}\|$) and $\|\mathbf{x}\|_\infty$ denote its 1-norm, 2-norm, and ∞ -norm, respectively. We also denote by $\text{diag}(\mathbf{x})$ the diagonal matrix whose diagonal elements are the elements of vector \mathbf{x} .

For a matrix A , A^T denotes its transpose, A^\dagger its pseudo-inverse, $[A]_{ij}$ or A_{ij} the ij^{th} element, $\text{rank}(A)$ the rank, $\text{tr}(A)$ the trace, $\rho(A)$ its spectral radius, $\|A\|$ the

(induced) 2-norm of A , and $|A|$ the matrix composed of absolute value of elements of A , i.e., $[|A|]_{ij} = |[A]_{ij}|, \forall i, j$. We also use $A^{(i)}$ and $A_{(j)}$, respectively, to denote the i -th row and j -th column of A .

Sets are denoted by calligraphic upper case letters. For a given set \mathcal{A} , $|\mathcal{A}|$ or $\text{card}(\mathcal{A})$ denotes its cardinality, and $\chi_{\mathcal{A}}$ denotes the associated indicator function. The degree of a polynomial q is denoted by $\text{deg}(q)$.

Basic Notion:

A matrix $A = [a_{ij}]$ is nonnegative (positive) if $a_{ij} \geq 0$ ($a_{ij} > 0$), $\forall i, j$. If $A - B$ is a nonnegative matrix, we write $A \geq B$. A nonnegative square matrix A is row stochastic (or simply stochastic) if $A\mathbf{1} = \mathbf{1}$, (row) substochastic if $A\mathbf{1} \leq \mathbf{1}$, column stochastic if $A^T\mathbf{1} = \mathbf{1}$, and doubly stochastic if it is both row and column stochastic. A square matrix A is called an M-matrix if (i) all the off-diagonal elements are nonpositive, i.e., $a_{ij} \leq 0, \forall i \neq j$, and (ii) it can be expressed as $A = sI - B$ where B is a nonnegative matrix such that $\rho(B) \leq s$.

A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a finite set of nodes $\mathcal{V} = \{1, 2, \dots, N\}$ and a set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of edges, where an ordered pair $(i, j) \in \mathcal{E}$ indicates that agent i receives information on the state of agent j . A directed path is a sequence of edges in the form $(i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)$. A simple path is a path without any node repeated. Node i is said to be reachable from node j if there exists a path from j to i . Each node is reachable from itself (i.e., self-loop is permitted). For node i , $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ is the set of in-neighbors (or neighbors for short), and $|\mathcal{N}_i|$ is the degree (also in-degree) of node i . Graph \mathcal{G} is connected (or weakly connected) if it cannot be partitioned into 2 separate groups that have no paths connecting

them. Graph \mathcal{G} is strongly connected if each node is reachable from any other node. A tree is a graph that has a node called root from which all the other nodes are reachable. The diameter of a connected graph \mathcal{G} , denoted by $\text{diam}(\mathcal{G})$, is the length of the longest path among all simple paths.

Let $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be a convex function. The domain of f is denoted by $\text{dom}(f)$. We denote by $\partial f(\mathbf{x})$ the subdifferential of f at $\mathbf{x} \in \text{dom}(f)$, i.e., the set of all subgradients of f at \mathbf{x} :

$$\partial f(\mathbf{x}) = \{\mathbf{g} \in \mathbb{R}^m : f(\mathbf{y}) - f(\mathbf{x}) \geq \mathbf{g}^T(\mathbf{y} - \mathbf{x}), \forall \mathbf{y} \in \text{dom}(f)\}. \quad (1.2)$$

A differentiable function f is called strongly convex with parameter $\mu > 0$ if for any $\mathbf{x}, \mathbf{y} \in \text{dom}(f)$,

$$f(\mathbf{y}) - f(\mathbf{x}) \geq \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|. \quad (1.3)$$

1.5.2 Convergence of DeGroot Model

This subsection presents convergence conditions for the DeGroot model (1.1) that will serve as a basic result for our development in the sequel. Consider a leaderless network consisting of N agents denoted by $\mathcal{V} = \{1, 2, \dots, N\}$. The underlying communication is characterized by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The update of each agent i 's opinion at any time $t \geq 0$ (here t denotes time, which can take any nonnegative integer value) can also be given as follows:

$$x_i(t+1) = \frac{\sum_{j \in \mathcal{N}_i} w_{ij} x_j(t)}{\sum_{j \in \mathcal{N}_i} w_{ij}}, \quad x_i(0) = x_{0i} \in \mathbb{R}, \quad (1.4)$$

where $w_{ij} \in [0, \infty)$ quantifies the unnormalized weight that agent i places on agent j 's opinion and recall that \mathcal{N}_i denotes the set of node i 's immediate neighbors

(including itself). The weight matrix of the network is denoted $W := [w_{ij}] \in \mathbb{R}^{N \times N}$, with the inter-nodal influence parameters $w_{ij} > 0$ when there is a direct link from agent j to agent i and $w_{ij} = 0$ if no such link exists.

Definition 1.5.1. Consider the DeGroot model (1.1). The network achieves consensus if for any initial opinions, it holds that $\lim_{t \rightarrow \infty} |x_i(t) - x_j(t)| = 0, \forall i, j = 1, \dots, N$.

In most chapters (except Chapter 2), we will make the following assumptions on the communication graph \mathcal{G} and the weight matrix W , which are usually imposed to ensure consensus of the network.

Assumption 1.5.2. *The network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a fixed and strongly connected directed graph.*

Assumption 1.5.3. *The matrix $W = [w_{ij}]$ is row stochastic and satisfies $w_{ij} > 0$ if $(i, j) \in \mathcal{E}$, $w_{ii} > 0$ for some $i \in \mathcal{V}$, and $w_{ij} = 0$ otherwise.*

Assumption 1.5.3 means that the zero-nonzero structure of the weight matrix W reflects the network structure. Moreover, W is now a normalized weight matrix. Thus, we can also express (1.4) compactly as

$$\mathbf{x}(t+1) = W\mathbf{x}(t). \tag{1.5}$$

It is well known (see, e.g., [1, 5, 14, 15, 49]) that under the assumptions above, the network achieves consensus. In fact, W is irreducible and represents an ergodic Markov chain. Let $\boldsymbol{\pi}$ denote the stationary distribution of W , i.e., $\boldsymbol{\pi}$ is the left eigenvector of W corresponding to the eigenvalue 1 and satisfying the condition $\mathbf{1}^T \boldsymbol{\pi} = 1$. The following result is the well known Perron-Frobenius theorem for

irreducible matrices (see, e.g. [95]), which lays the foundation for theories of Markov chains and network consensus (see, e.g., [14, 15] and references therein).

Theorem 1.5.4. (see, e.g., [95]) *If W is row stochastic and irreducible, then*

1. W has spectral radius $\rho(W) = 1$, which is also a simple eigenvalue.
2. $\boldsymbol{\pi}^T W = \boldsymbol{\pi}^T$ and $\boldsymbol{\pi}$ is a strictly positive vector.
3. $\exists \lim_{t \rightarrow \infty} W^t = \mathbf{1}\boldsymbol{\pi}^T$. The convergence rate is geometric and determined by the second largest eigenvalue of W .

Corollary 1.5.5. *Under Assumptions 1.5.2 and 1.5.3, the network (1.4) achieves consensus and*

$$\lim_{t \rightarrow \infty} x_i(t) = \boldsymbol{\pi}^T \mathbf{x}(0), \quad \forall i \in \mathcal{V}, \quad (1.6)$$

Moreover, the convergence rate is geometric.

Clearly, the consensus value depends on the weight matrix W and the initial opinions $\mathbf{x}(0)$.

We also often define a weighted Laplacian matrix $L = [l_{ij}] \in \mathbb{R}^{N \times N}$ satisfying

$$\epsilon L = I - W$$

for some $\epsilon > 0$. The following are obvious.

$$\boldsymbol{\pi}^T L = \mathbf{0}^T, \quad L \mathbf{1} = \mathbf{0}.$$

Moreover, all the eigenvalues of L are positively stable except only one at the origin.

Part I: Consensus Network with Leaders

Chapter 2: Opinion Dynamics with Persistent Leaders

Abstract: This chapter revisits the problem of agreement seeking in a network of agents under the influence of leaders. The persistence of the effect of the leader (or leaders) on the opinions of the network agents is characterized by the total weight that they place on the leader's information over time. If this weight is infinite, then the leader is called *persistent*. We will describe the asymptotic behavior of network opinions towards the state of a persistent leader in both cases of fixed and switching network topologies. We also show that only persistent leaders are able to drive the network to the leader's constant state.

2.1 Introduction

It is widely known that both the communication graph and the influence structure of a network play important roles in reaching consensus. The former indicates whom an agent interacts with while the latter determines the weights on the information that he receives from others. In most existing work on network consensus with or without leaders, the weights are usually assumed to be constant or varying in a compact set bounded away from zero (see, e.g., [1,5,13,15] and references therein). In practice, however, the connections between agents may be transient and the weights

may fluctuate in a broad range and even diminish with time. Recently, the notion of *persistent graphs* was studied in [55], where it was shown that *persistent links* are crucial in seeking an agreement. However, the work [55] considered networks without a leader and required stronger conditions than just the persistence of the agent interconnections. In this chapter, we study networks in the presence of leaders and show that consensus can still be achieved under milder requirements.

A great number of recent efforts have also been devoted to the consensus problem in networks with leaders (see, e.g., [5, 12, 13, 20] and references therein). A general conclusion is that consensus cannot be achieved when the leaders have competing opinions. This is mainly because only persistent leaders were considered. In this chapter, we show that network agreement can still be reached in the presence of competing leaders provided that only one of them is persistent. This also distinguishes our results from existing works in this setting.

The main contributions of this work are as follows. First, we derive new sufficient conditions for guaranteeing agreement in networks with a leader for both fixed and switching topologies. These conditions emphasize the persistence of the connectivity between the leader and the followers and, to the best of our knowledge, are the mildest, covering many existing results in the literature. Second, we show that in a network with more than one leader, only those that are persistent matter. In particular, when there is only one persistent leader, we provide conditions under which the network converges to the state of this leader. Most of the results in this chapter were first presented in [96].

The rest of the chapter is organized as follows. In Section 2.2, we describe

the problem of interest in detail. Sections 2.3 and 2.4 present the main results for networks with a single leader and two leaders, respectively. Finally, discussion and future work are given in Section 2.5.

2.2 Problem Formulation

Consider a set of N agents or nodes interacting over a communication network. The topology of the network at time $t \in \mathbb{Z}_+$ is described by a graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. Let $x_i(t) \in [0, 1]$ denote the opinion of node i at time t . At the initial time $t = 0$, each agent has an initial opinion $x_i(0)$. Suppose that at each time t , every agent synchronously obtains opinions of his neighbors and *naively* updates his opinion following the DeGroot discrete-time model [1]

$$x_i(t+1) = \sum_{j \in \mathcal{V}} w_{ij}(t)x_j(t), \quad \forall i \in \mathcal{V}, \quad (2.1)$$

where $w_{ij}(t) \geq 0$ indicates the weight that agent i puts on agent j 's opinion at time t . Here $W(t) = [w_{ij}(t)] \in \mathbb{R}^{N \times N}$ represents the weight matrix (or trust matrix) at time t . We will assume that $W(t)$ is a row stochastic matrix for any t , i.e., $W(t)$ is nonnegative and $W(t)\mathbf{1} = \mathbf{1}$.

Now we consider to the above network under the effect of an external media node, representing a leader or a source of news with a constant opinion value $T \in [0, 1]$. Although it is often thought of as a stubborn node indistinguishable from the others, we consider it separately from that context as we want to look at the network from the point of view of a *leader* and investigate its effect on the opinions of other regular agents, conventionally called *followers*. To this end, assume that

the leader can connect to some of the followers and persuade them to trust in its opinion T with trust levels $\alpha_i(t) \in [0, 1], \forall i \in \mathcal{V}$. Here, $\alpha_i(t) = 0$ means distrust or unawareness of the leader and $\alpha_i(t) = 1$ implies absolute trust. The update rule is then given by

$$x_i(t+1) = \alpha_i(t)T + (1 - \alpha_i(t)) \sum_{j \in \mathcal{V}} w_{ij}(t)x_j(t), \quad \forall i \in \mathcal{V}. \quad (2.2)$$

In the matrix form, (2.2) reads

$$\mathbf{x}(t+1) = \boldsymbol{\alpha}(t)T + \Gamma(t)W(t)\mathbf{x}(t), \quad (2.3)$$

where $\boldsymbol{\alpha}(t) = [\alpha_1(t), \dots, \alpha_N(t)]^T$ and $\Gamma(t) = I - \text{diag}(\boldsymbol{\alpha}(t))$ with $I \in \mathbb{R}^{N \times N}$ being the identity matrix. We are interested in finding conditions under which the network can finally agree with the leader opinion. Indeed, we will determine how strong the connections between the leader and the followers should be to ensure that this agreement can be achieved asymptotically. We will also extend these results to the case of the network with more than one leader.

The following notions will be used in this chapter. See, e.g., [8, 55].

Definition 2.2.1. Consider a time-varying graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ with an associated time-dependent weight matrix $W(t)$.

- Link (i, j) is called persistent if $\sum_{t \geq 0} w_{ij}(t) = \infty$.
- Node i is persistent if $\sum_{t \geq 0} \sum_{k=1}^N w_{ki}(t) = \infty$.
- The persistent graph \mathcal{G}_∞ induced by $\{\mathcal{G}(t), W(t), \forall t \in \mathbb{Z}_+\}$ is the graph containing all the persistent links.

2.3 Opinion Dynamics with One Leader

This section studies the convergence of opinions under different assumptions on the connection topology of the followers' network and the connectivity of the leader with the followers.

First, it is easy to see that if there exists $t^* \in \mathbb{Z}_+$ such that $\mathbf{x}(t^*) = T\mathbf{1}$ (e.g., $\boldsymbol{\alpha}(t^* - 1) = \mathbf{1}$), then $\mathbf{x}(t) = T\mathbf{1}, \forall t \geq t^*$, i.e., the network converges to T in finite-time (at most t^* steps) for any initial opinions $\mathbf{x}(0)$.

Second, it is also known that when W and $\boldsymbol{\alpha}$ are constant, then $\mathbf{x}(t) \rightarrow T\mathbf{1}$ for any $\mathbf{x}(0)$ if the extended graph including the leader and the followers is a spanning tree whose root is at the leader (see, e.g., [9]).

Third, consider the case when the leader's effect lasts for an interval $[0, t_0]$, e.g., a campaign period with $\boldsymbol{\alpha}(t) = \mathbf{0}, \forall t > t_0$. Suppose that W is fixed and the network is strongly connected. Then there exists $\boldsymbol{\pi} \in \mathbb{R}^N$ such that $\lim_{t \rightarrow \infty} W^t = \mathbf{1}\boldsymbol{\pi}^T$ (cf. Section 1.5). As a result, $\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{1}\boldsymbol{\pi}^T \mathbf{x}(t_0)$, i.e., the consensus value may differ from T . If $\boldsymbol{\alpha}(t) \equiv \boldsymbol{\alpha}, \forall t \in [0, t_0]$, then it can be verified that

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{1} (T - \boldsymbol{\pi}^T (\Gamma W)^{t_0+1} (\mathbf{1}T - \mathbf{x}(0))). \quad (2.4)$$

Since ΓW is strictly substochastic and irreducible, it follows that $\rho(\Gamma W) < 1$ (see, e.g., [97, Thm 1.1, p. 24]), and thus $\lim_{t_0 \rightarrow \infty} (\Gamma W)^{t_0} = 0$. Therefore, $\mathbf{x}(\infty) = T\mathbf{1}$ when $t_0 \rightarrow \infty$, i.e., consensus to the leader's state when the leader is persistent.

Next, we allow $\boldsymbol{\alpha}(t)$ to be time-varying and the limit $\lim_{t \rightarrow \infty} \boldsymbol{\alpha}(t)$ need not exist. Some related works are [5, 57] and [59]. Notice that the results in [5] rely

on Wolfowitz's theorem [56] on the convergence of infinite products of stochastic matrices belonging to a finite set. This condition is relaxed in [57] so that the matrices can belong to an infinite set. However, the work [57] requires the symmetry of the zero/nonzero structure of these matrices. Similarly, the limiting behavior of products of random stochastic matrices is also studied in [59] assuming the cut-balanced property of the sequence of these matrices. This property is in the same spirit of having symmetric zero/nonzero structures. Here, we need not impose those conditions.

The following condition suffices to ensure the asymptotic convergence of the network to the leader's opinion.

Theorem 2.3.1. (One Leader, Arbitrary Graph) *Consider system (2.2) and suppose that the weights on the leader satisfy*

$$\sum_{t \geq 0} \min_{i \in \mathcal{V}} \alpha_i(t) = \infty. \quad (2.5)$$

Then $\mathbf{x}(t) \rightarrow T\mathbf{1}$ as $t \rightarrow \infty$ for any initial opinion $\mathbf{x}(0)$.

Proof. The proof follows the method presented in [46]. Let $\tilde{\mathbf{x}}(t) = [\mathbf{x}(t)^T \ T]^T$.

Equation (2.3) is equivalent to

$$\tilde{\mathbf{x}}(t+1) = \tilde{W}(t)\tilde{\mathbf{x}}(t), \quad \tilde{W}(t) \triangleq \begin{bmatrix} \Gamma(t)W & \boldsymbol{\alpha}(t) \\ 0 & 1 \end{bmatrix}. \quad (2.6)$$

Let $h(t) = \max_{1 \leq i, j \leq N+1} (\tilde{x}_i(t) - \tilde{x}_j(t))$. Obviously, $h(t) \geq 0, \forall t \geq 0$. Now

$$\begin{aligned} h(t+1) &= \max_{1 \leq i, j \leq N+1} (\tilde{x}_i(t+1) - \tilde{x}_j(t+1)) \\ &= \max_{1 \leq i, j \leq N+1} \left(\sum_{1 \leq k \leq N+1} \tilde{w}_{ik} \tilde{x}_k(t) - \sum_{1 \leq k \leq N+1} \tilde{w}_{jk} \tilde{x}_k(t) \right) \\ &= \max_{1 \leq i, j \leq N+1} \sum_{1 \leq k \leq N+1} (\tilde{w}_{ik} - \tilde{w}_{jk}) \tilde{x}_k(t). \end{aligned}$$

Denote $\tilde{h}_{ij}(t) = \sum_{k=1}^{N+1} (\tilde{w}_{ik} - \tilde{w}_{jk}) \tilde{x}_k(t)$. Then

$$\begin{aligned} \tilde{h}_{ij}(t) &= \sum_{1 \leq k \leq N+1} (\tilde{w}_{ik} - \min(\tilde{w}_{ik}, \tilde{w}_{jk})) \tilde{x}_k(t) - \sum_{1 \leq k \leq N+1} (\tilde{w}_{jk} - \min(\tilde{w}_{ik}, \tilde{w}_{jk})) \tilde{x}_k(t) \\ &\leq \sum_{1 \leq k \leq N+1} (\tilde{w}_{ik} - \min(\tilde{w}_{ik}, \tilde{w}_{jk})) \max_l \tilde{x}_l(t) - \sum_{1 \leq k \leq N+1} (\tilde{w}_{jk} - \min(\tilde{w}_{ik}, \tilde{w}_{jk})) \min_l \tilde{x}_l(t). \end{aligned}$$

Rearranging terms and using the fact that $\sum_{k=1}^{N+1} \tilde{w}_{ik} = 1, i = 1, \dots, N+1$ yield

$$\begin{aligned} \tilde{h}_{ij}(t) &\leq \max_l \tilde{x}_l(t) - \min_l \tilde{x}_l(t) - \sum_{1 \leq k \leq N+1} \min(\tilde{w}_{ik}, \tilde{w}_{jk}) (\max_l \tilde{x}_l(t) - \min_l \tilde{x}_l(t)) \\ &= (\max_l \tilde{x}_l(t) - \min_l \tilde{x}_l(t)) \left(1 - \sum_{1 \leq k \leq N+1} \min(\tilde{w}_{ik}, \tilde{w}_{jk}) \right). \end{aligned}$$

Therefore,

$$\begin{aligned} h(t+1) &\leq \max_{1 \leq i, j \leq N+1} h(t) \left(1 - \sum_{1 \leq k \leq N+1} \min(\tilde{w}_{ik}, \tilde{w}_{jk}) \right) \\ &= h(t) \left(1 - \min_{1 \leq i, j \leq N+1} \sum_{1 \leq k \leq N+1} \min(\tilde{w}_{ik}, \tilde{w}_{jk}) \right). \end{aligned} \quad (2.7)$$

Now for any $i, j \in \mathcal{V}$

$$\sum_{1 \leq k \leq N+1} \min(\tilde{w}_{ik}, \tilde{w}_{jk}) \geq \min(\tilde{w}_{i, N+1}, \tilde{w}_{j, N+1}) = \min(\alpha_i(t), \alpha_j(t)) \geq \min_{k \in \mathcal{V}} \alpha_k(t) \quad (2.8)$$

Define $\underline{\alpha}(t) := \min_{k \in \mathcal{V}} \alpha_k(t)$. It follows immediately from (2.7) that, for any $t \geq 0$

$$h(t+1) \leq h(t)(1 - \underline{\alpha}(t)) \leq h(t)e^{-\underline{\alpha}(t)} \leq h(0)e^{-\sum_{s=0}^t \underline{\alpha}(s)}. \quad (2.9)$$

The second inequality follows from the fact that $1 - z \leq e^{-z}$, $\forall z \geq 0$. The assumption that $\sum_{s=0}^{\infty} \underline{\alpha}(s) = \infty$ implies that $\lim_{t \rightarrow \infty} h(t) \rightarrow 0$, hence $x_i(t) \rightarrow T, \forall i \in \mathcal{V}$. \square

Clearly, the result above holds for any structure of the network and weight matrix (even in the time-varying case) as it merely relies on the condition (2.5), which means that every node in the network *persistently* trusts the leader (in the sense that $\sum_{t \geq 0} \alpha_i(t) = \infty, \forall i \in \mathcal{V}$). Note that the notion of persistent graphs, that is a graph that consists of links satisfying $\sum_{t \geq 0} w_{ij}(t) = \infty$, was also studied in [55]. However, to guarantee the global agreement, [55] requires further that there exist $a_* > 0$ and $T_* > 0$ such that $\sum_{s=t}^{t+T_*-1} w_{ij}(s) \geq a_*, \forall t \geq 0$ and for all persistent links (i, j) . This condition is stronger than the condition of being a persistent link. Therefore, the results in [55] cannot be applied in this case.

One can notice that condition (2.5) is rather strong. In practice, there are many situations where ensuring this condition may be costly since the leader needs to directly approach every agent in the network for an infinite number of times. This is usually not the most practical advertising strategy either. In fact, the leader should make use of the connections between the followers to advertise for its opinion. Therefore, below, we relax this condition by imposing requirements on the network structure.

Assumption 2.3.2. *The graph \mathcal{G} is fixed and strongly connected. The weight matrix W is fixed and has positive diagonal elements, that is $w_{ii} > 0, \forall i \in \mathcal{V}$.*

Theorem 2.3.3. (One Leader, Strongly Connected Graph, Fixed Weight) *Consider*

system (2.2) and let Assumption 2.3.2 hold. Suppose the weights α_i satisfy

$$\sum_{t \geq 0} \max_{i \in \mathcal{V}} \alpha_i(t) = \infty. \quad (2.10)$$

Then $\mathbf{x}(t) \rightarrow T\mathbf{1}$ as $t \rightarrow \infty$ for any $\mathbf{x}(0)$.

Before giving the proof, we make a few remarks. First, Theorems 2.3.3 and 2.3.1 show the importance of persistent links (including constant weights as a special case) in shaping the final opinion. Second, since the network size is finite, (2.10) is equivalent to the condition that at least one follower persistently trusts the leader even if its trust level fades away. This condition holds for many plausible specifications of α_i , e.g., $\alpha_i(t) = \frac{c}{(t+1)^\gamma}$ for any $c \in (0, 1]$ and $\gamma \in [0, 1]$.

It is tempting to follow the proof of Theorem 2.3.1 which is primarily based on inequalities (2.7) and (2.8). However, under condition (2.10) this technique is no longer applicable. Consider, e.g., a connected undirected network with $N = 5$ and

$$W = \begin{bmatrix} w_{11} & w_{12} & 0 & 0 & 0 \\ w_{21} & w_{22} & w_{23} & 0 & 0 \\ 0 & w_{32} & w_{33} & w_{34} & 0 \\ 0 & 0 & w_{43} & w_{44} & w_{45} \\ 0 & 0 & 0 & w_{54} & w_{55} \end{bmatrix}.$$

Assume that $\alpha_i(t) = 0, \forall i \in \mathcal{V} \setminus \{3\}, \forall t \geq 0$ and $\sum_{t \geq 0} \alpha_3(t) = \infty$. It can be seen that

$$\min_{1 \leq i, j \leq N+1} \sum_{1 \leq k \leq N+1} \min(\tilde{w}_{ik}, \tilde{w}_{jk}) = 0.$$

Thus, from (2.7), we can only obtain that $h(t+1) \leq h(t), \forall t \geq 0$, which is not enough to ensure the convergence of $h(t)$ to 0.

Moreover, since $\alpha(t)$ is not restricted to belong to a finite set, the results in [5] cannot be used. Further, neither W nor $\tilde{W}(t)$ (see (2.6)) is required to have a symmetric zero/nonzero structure, the results in [57, 59] and [98] are not applicable. The following proof uses the results presented in [99] on the convergence of infinite products of substochastic matrices. Notice that [99, Theorem 6.2] requires that the smallest row-sums of all the matrices be uniformly bounded away and below 1. Here, we require milder conditions.

The following results are needed to proceed the proof of Theorem 2.3.3.

Lemma 2.3.4. [99] *Let $M_i \in \mathbb{R}^{n \times n}, i = 1, \dots, m$ be any m substochastic matrices, then the product $P = \prod_{i=1}^m M_i$ is also a substochastic matrix.*

Lemma 2.3.5. [99] *Let $M_i \in \mathbb{R}^{n \times n}, i = 1, \dots, n - 1$, be irreducible substochastic matrices with positive diagonals, then the product $P = \prod_{i=1}^{n-1} M_i$ is a strictly positive matrix, i.e., $P_{ij} > 0, \forall i, j$. Further, let*

$$\underline{m} = \min\{[M_k]_{ij} | i, j \in [1, n], k \in [1, n - 1], [M_k]_{ij} > 0\},$$

then $\min_{i,j} P_{ij} \geq \underline{m}^{n-1}$.

For any matrix M , let $r_i(M) \triangleq \sum_j M_{ij}$, i.e., the sum of i^{th} -row of the matrix.

Lemma 2.3.6. *Suppose that $M_i \in \mathbb{R}^{n \times n}, i = 1, \dots, m$ satisfy $\min_i r_i(M_1) \leq \underline{r}_1$ and $\max_i r_i(M_k) \leq \bar{r}_k, k = 2, \dots, m$. Then it holds that $\min_i r_i(M_1 \dots M_m) \leq \underline{r}_1 \bar{r}_2 \dots \bar{r}_m$.*

Proof. If $\min_i r_i(M_1) \leq \underline{r}_1$ and $\max_i r_i(M_2) \leq \bar{r}_2$, then $r_i(M_1 M_2) = \sum_j [M_1]_{ij} r_j[M_2] \leq \sum_j [M_1]_{ij} \bar{r}_2 = r_i[M_1] \bar{r}_2$. Thus, $\min_i r_i(M_1 M_2) \leq \underline{r}_1 \bar{r}_2$. By induction, it holds that $\min_i r_i(M_1 \dots M_m) \leq \underline{r}_1 \bar{r}_2 \dots \bar{r}_m$ provided $\max_i r_i(M_k) \leq \bar{r}_k, k = 2, \dots, m$. \square

Lemma 2.3.7. *Let U, V, D_1 and D_2 be nonnegative matrices with appropriate dimensions. If $D_1 \leq D_2$, then $\|UD_1V\|_\infty \leq \|UD_2V\|_\infty$.*

Proof. Let $U^{(i)}$ denote the i -th row of U and $V_{(j)}$ the j -th column of V . Since U, V, D_1 and D_2 are nonnegative, it follows that $0 \leq U^{(i)}D_1V_{(j)} \leq U^{(i)}D_2V_{(j)}$ for $\forall i, j$. Thus $\mathbf{0} \leq UD_1V \leq UD_2V$, and hence, $\|UD_1V\|_\infty \leq \|UD_2V\|_\infty$. \square

The proof of Theorem 2.3.3 is presented next.

Proof. Defining $\boldsymbol{\xi}(t) \triangleq \mathbf{x}(t) - T\mathbf{1}$, the update rule (2.3) can be expressed as follows

$$\boldsymbol{\xi}(t+1) = A(t)\boldsymbol{\xi}(t), \quad A(t) \triangleq \Gamma(t)W, \quad (2.11)$$

where $\Gamma(t) = I - \text{diag}(\boldsymbol{\alpha}(t))$. We need to show that $\lim_{t \rightarrow \infty} \boldsymbol{\xi}(t) = \mathbf{0}$ for any $\boldsymbol{\xi}(0)$, or equivalently,

$$\lim_{s \rightarrow \infty} \left\| \prod_{0 \leq t \leq s} A(t) \right\|_\infty = 0, \quad (2.12)$$

where $\prod_{0 \leq t \leq s} A(t) \triangleq A(s) \dots A(0)$. Note that although $A(t)$ is substochastic for all $t \geq 0$ (hence $\rho(A(t)) \leq 1$), it does not automatically imply that $\prod_{t \geq 0} A(t) = 0$;

this is true even if $A(t)$ is strictly substochastic. (For example, the sequence $a_i =$

$\frac{1}{2} \sqrt{\underbrace{2 + \sqrt{2 + \dots + \sqrt{2}}}_{i \text{ times}}}$ satisfies $a_i \in (0, 1), \forall i \geq 1$, but $\prod_{i=1}^\infty a_i = \frac{2}{\pi}$.)

Take any $\eta \in (0, 1)$ and define

$$\tilde{A}(t) = \tilde{\Gamma}(t)W, \quad \tilde{\Gamma}(t) = I - \text{diag}(\eta\boldsymbol{\alpha}(t)).$$

Obviously, $\Gamma(t) \leq \tilde{\Gamma}(t), \forall t \geq 0$. Applying Lemma 2.3.7, we have

$$\left\| \prod_{0 \leq t \leq s} A(t) \right\|_\infty \leq \left\| \prod_{0 \leq t \leq s} \tilde{A}(t) \right\|_\infty, \quad \forall s \geq 0.$$

Thus, the following condition suffices for (2.12)

$$\lim_{s \rightarrow \infty} \left\| \prod_{0 \leq t \leq s} \tilde{A}(t) \right\|_{\infty} = 0. \quad (2.13)$$

Next, define

$$B(s) := \prod_{sN_1 \leq t \leq (s+1)N_1 - 1} \tilde{A}(t), \quad \text{with } N_1 := N - 1, \quad (2.14)$$

and note the following.

- (i) $\exists \underline{b} > 0$ such that $\min_{i,j} B_{ij}(s) \geq \underline{b}, \forall s \geq 0$ and $\forall i, j \in \mathcal{V}$. This can be shown as follows. Let

$$\underline{w} = \min\{w_{ij} \mid i, j \in \mathcal{V}, w_{ij} > 0\}. \quad (2.15)$$

Then, $\min\{\tilde{A}_{ij}(t) \mid i, j \in \mathcal{V}, \tilde{A}_{ij}(t) > 0\} \geq (1 - \eta)\underline{w}$. Note that $\tilde{A}(t)$ is irreducible since the network is strongly connected (by assumption). Thus, by Lemma 2.3.5, we have

$$\min_{i,j} B_{ij}(s) \geq ((1 - \eta)\underline{w})^{N_1} =: \underline{b}. \quad (2.16)$$

- (ii) $\max_i r_i(B(s)) \leq 1$, since $B(s)$ is substochastic any s (cf. Lemma 2.3.4).

- (iii) $\min_i r_i(B(s)) \leq 1 - \delta((s+1)N_1 - 1)$ where

$$\delta(t) := \eta \max_{i \in \mathcal{V}} \alpha_i(t), \quad \forall t \geq 0.$$

This can be obtained by using Lemma 2.3.6 with

$$\min_i r_i \left(\tilde{A}((s+1)N_1 - 1) \right) \leq 1 - \delta((s+1)N_1 - 1)$$

and $\max_i r_i(\tilde{A}(t)) \leq 1$ for $t = sN_1, \dots, (s+1)N_1 - 2$.

Now, let $r_{j^*}(B(s))$ denote the smallest row sum of $B(s)$. Using the above results yields

$$\begin{aligned}
r_i(B(s+1)B(s)) &= \sum_{1 \leq j \leq N} B_{ij}(s+1)r_j(B(s)) \\
&= B_{ij^*}(s+1)r_{j^*}(B(s)) + \sum_{1 \leq j \leq N, j \neq j^*} B_{ij}(s+1)r_j(B(s)) \\
&\stackrel{\text{(ii)-(iii)}}{\leq} B_{ij^*}(s+1)[1 - \delta((s+1)N_1 - 1)] + \sum_{1 \leq j \leq N, j \neq j^*} B_{ij}(s+1) \\
&\stackrel{\text{(ii)}}{\leq} B_{ij^*}(s+1)[1 - \delta((s+1)N_1 - 1)] + 1 - B_{ij^*}(s+1) \\
&\stackrel{\text{(i)}}{\leq} 1 - \underline{b}\delta((s+1)N_1 - 1) \\
&\leq e^{-b\delta((s+1)N_1 - 1)} \quad \forall i \in \mathcal{V},
\end{aligned}$$

where the last inequality follows from the fact that $1 - z \leq e^{-z}, \forall z \geq 0$. Thus

$$\begin{aligned}
\left\| \prod_{0 \leq t \leq (2m+2)N_1 - 1} \tilde{A}(t) \right\|_\infty &= \left\| \prod_{0 \leq s \leq m} B(2s+1)B(2s) \right\|_\infty \\
&\leq \prod_{0 \leq s \leq m} \|B(2s+1)B(2s)\|_\infty \\
&\leq e^{-\underline{b}\sum_{s=0}^m \delta((2s+1)N_1 - 1)}. \tag{2.17}
\end{aligned}$$

If $\sum_{s=0}^\infty \delta((2s+1)N_1 - 1) = \infty$, then the right side of (2.17) decays to 0 as $m \rightarrow \infty$, thus (2.13) follows immediately. Therefore, it remains to show that this is also the case when (2.10) holds, or equivalently $\sum_{t \geq 0} \delta(t) = \infty$. To this end, let

$$\Delta_i^m := \sum_{0 \leq j \leq m} \delta(i + j2N_1), \quad \forall i \in [0, 2N_1 - 1]$$

and note that

$$\sum_{0 \leq t \leq 2(m+1)N_1 - 1} \delta(t) = \sum_{0 \leq s \leq 2N_1 - 1} \Delta_s^m. \tag{2.18}$$

Now, let (2.10) hold. We claim that there must exist $k \in [0, 2N_1 - 1]$ such that $\Delta_k^\infty = \infty$. This can be shown by contradiction, i.e., if $\Delta_i^\infty < \infty, \forall i \in [0, 2N_1 - 1]$,

then $\sum_{s=0}^{2N_1-1} \Delta_s^\infty < \infty$, thus taking the limit of both sides of (2.18) as $m \rightarrow \infty$ yields $\sum_{t \geq 0} \delta(t) < \infty$, which contradicts (2.10). Thus the claim holds.

Now, if $k = N_1 - 1$, i.e., $\Delta_{N_1-1}^\infty = \infty$, then we obtain the desired result, otherwise we can redefine $B(s) \triangleq \prod_{t=sN_1+k}^{(s+1)N_1+k-1} \tilde{A}(t)$ and follow the same steps as above to show that $\prod_{t=k}^\infty \tilde{A}(t) = 0$ and thus $\prod_{t \geq 0} \tilde{A}(t) = 0$. This completes the proof. \square

Note that we can take $N_1 = d_0$ in (2.14), where d_0 denotes the diameter of the graph \mathcal{G} , and then repeat the above proof using a slight modification of Lemma 2.3.5 applied to $\tilde{A}(t) = \tilde{\Gamma}(t)W$. In fact,

$$P = \prod_{k \leq i \leq k+d_0-1} \tilde{A}(i)$$

is strictly positive for any k . The proof of this is not much different from that of Lemma 2.3.5, thus skipped here. Note also that $d_0 = N - 1$ in the worst case.

The above proof also allows us to estimate the ϵ -convergence time for some cases of $\alpha_i(\cdot)$ as follows.

Corollary 2.3.8. (ϵ -Convergence Time) *Let d_0 be the diameter of the graph \mathcal{G} and \underline{w} be defined as in (2.15). Given any number $\epsilon > 0$, it holds that*

$$\|\mathbf{x}(t) - T\mathbf{1}\|_\infty \leq \epsilon \|\mathbf{x}(0) - T\mathbf{1}\|_\infty$$

if $t > 2d_0m$ where

$$(i) \quad m = \exp\left(\frac{2d_0}{(0.5\underline{w})^{d_0}} \log \epsilon^{-1}\right) \text{ if } \max_i \alpha_i(\tau) = \frac{1}{\tau+1}, \text{ or}$$

$$(ii) \quad m = \frac{1}{\bar{\alpha}((1-\bar{\alpha})\underline{w})^{d_0}} \log \epsilon^{-1} \text{ if } \max_i \alpha_i(\tau) = \bar{\alpha} \in (0, 1) \text{ for all } \tau \geq 0.$$

Proof. The proof follows from (2.17) and the fact that $\sum_{t=1}^k \frac{1}{t} > \ln(k+1)$. \square

The following result is a straight extension of Theorem 2.3.3 to the case of time-varying weight matrices.

Assumption 2.3.9. (Strong Weight) *The weight matrix $W(t)$ is row stochastic and satisfies*

a) $w_{ii}(t) \geq \gamma, \forall i \in \mathcal{V}$ for some $\gamma \in (0, 1)$,

b) $w_{ij}(t) \in \{0\} \cup [\gamma, 1), \forall i, j \in \mathcal{V}, i \neq j$.

Theorem 2.3.10. (One Leader, Strongly Connected Graph, Time-varying Weight)

Consider system (2.2) and suppose that $\mathcal{G}(t)$ is strongly connected for all $t \geq 0$ and that Assumption 2.3.9 holds. If (2.10) holds, then $\mathbf{x}(t) \rightarrow T\mathbf{1}$ for any $\mathbf{x}(0)$.

Proof. The only difference between Theorems 2.3.10 and 2.3.3 is that $A(t) = \Gamma(t)W(t)$.

However, under Assumption 2.3.9, we can choose $\underline{w} = \gamma$ and $\underline{b} = ((1 - \eta)\gamma)^{N-1}$ for (2.15) and (2.16), respectively, then follow the same steps as in the proof of Theorem 2.3.3. \square

Note that if there exists $t^* \in \mathbb{Z}_+$ such that $\prod_{t=0}^{t^*} A(t) = 0$ (e.g., $\boldsymbol{\alpha}(t^*) = \mathbf{1}$), then the network converges to T in at most $t^* + 1$ time steps for any initial opinion $\mathbf{x}(0)$. Thus, condition (2.10) used in Theorems 2.3.3 and 2.3.10 is only sufficient but not necessary.

The strong connectivity requirement can be further relaxed to the existence of a spanning tree provided that a root node believes in T persistently. In the following, we assume that node 1 be always a root node.

Theorem 2.3.11. (One Leader, Spanning Tree Graph, Time-varying Weight) *Suppose that the graph $\mathcal{G}(t)$ is a directed spanning tree whose root is at node 1 for all $t \geq 0$. Let Assumption 2.3.9 hold. It follows that $\mathbf{x}(t) \rightarrow T\mathbf{1}$ for any $\mathbf{x}(0)$ if*

$$\sum_{t \geq 0} \alpha_1(t) = \infty. \quad (2.19)$$

The intuition of the result is as follows. Condition (2.19) means that there is an infinite information flow from the leader into node 1. Since node 1 is the root of the tree, this flow arrives at every node of the network, thus consensus can be achieved. The idea of the proof follows that in Theorem 2.3.3 with some modifications.

Proof. For simplicity, assume that $\alpha_i(t) = 0, \forall t \geq 0, \forall i = 2, \dots, N$. The proof follows the same line of that of Theorem 2.3.3. Recall (2.11)-(2.13) and note that $\tilde{A}(t)$ is a substochastic matrix for any t ; specifically, $r_1(\tilde{A}(t)) \leq 1$ and $r_i(\tilde{A}(t)) = 1, i = 2, \dots, N$. Let d_0 denote the diameter of the tree. It can be proved that if any matrices A_1, \dots, A_{d_0} satisfy the conditions on $\tilde{A}(t)$, then $P \triangleq \prod_{i=1}^{d_0} A_i$ satisfies

$$P_{i1} \geq \underline{b} = ((1 - \eta)\gamma)^{d_0}, \quad i = 1, \dots, N. \quad (2.20)$$

Let $B(s) = \prod_{t=sd_0}^{(s+1)d_0-1} A(t)$. It follows that

$$\begin{aligned} r_i(B(s+1)B(s)) &= B_{i1}(s+1)r_1(B(s)) + \sum_{2 \leq j \leq N} B_{ij}(s+1)r_j(B(s)) \\ &\leq B_{i1}(s+1)(1 - \eta\alpha_1((s+1)d_0 - 1)) + \sum_{2 \leq j \leq N} B_{ij}(s+1) \\ &\leq B_{i1}(s+1)(1 - \eta\alpha_1((s+1)d_0 - 1)) + 1 - B_{i1}(s+1) \\ &\leq 1 - \underline{b}\eta\alpha_1((s+1)d_0 - 1) \\ &\leq e^{-\underline{b}\eta\alpha_1((s+1)d_0 - 1)} \quad \forall i \in \mathcal{V}, \end{aligned}$$

where the second to last inequality follows from (2.20). The rest of the proof follows that of Theorem 2.3.3. \square

We can further relax the condition on the connectivity of the network by employing the notion of bounded connectivity times (see, e.g., [100]). Before stating this result, we need the following lemma.

Lemma 2.3.12. [5] *Let $m \geq 2$ be a positive integer and let $A_1, A_2, \dots, A_m \in \mathbb{R}^{n \times n}$ be nonnegative matrices with positive diagonal elements satisfying $0 < \mu \leq [A_i]_{jj} \leq \rho, \forall i, \forall j$, then*

$$A_1 A_2 \dots A_m \geq \left(\frac{\mu^2}{2\rho} \right)^{m-1} (A_1 + A_2 + \dots + A_m).$$

As a consequence of this lemma, if the union of all graphs associated with A_i is a spanning tree, then the graph associated with the product $A_1 A_2 \dots A_m$ is also a spanning tree. For any integers $t \geq 0$ and $N_0 > 0$, define $\mathcal{G}_{[N_0]}(t) = (\mathcal{V}, \cup_{k=t}^{t+N_0-1} \mathcal{E}(k))$ as the union of a sequence of graphs over interval $[t, t + N_0)$. We state the following result.

Theorem 2.3.13. (One Leader, Periodically Spanning Tree Graph) *Consider system (2.2) and let Assumption 2.3.9-a) hold. Suppose there exists $N_0 > 0$ such that for all t , the graph $\mathcal{G}_{[N_0]}(t)$ admits a spanning tree whose root is at node 1 and edges satisfy*

$$\sum_{t \leq k \leq t+N_0-1} w_{ij}(k) \geq \gamma N_0. \quad (2.21)$$

If (2.19) holds, then $\mathbf{x}(t) \rightarrow T\mathbf{1}$ for any $\mathbf{x}(0)$.

Proof. (Sketch) Let $G_t^{N_0}$ denote a spanning tree in the union graph $\mathcal{G}_{[N_0]}(t)$ whose root is at node 1 and edges satisfy condition (2.21). This condition implies that during any interval of length N_0 and for any $(i, j) \in G_t^{N_0}$ there exists at least one time t_{ij}^* such that $w_{i,j}(t_{ij}^*) \geq \gamma$. Denote d_0 the maximum diameter of $\mathcal{G}_{[N_0]}(t), \forall t$. Since the self-weight w_{ii} of each agent is bounded away from 0 by γ , then one can see that every node in the network is reachable from node 1 in at most $d_0 N_0$ steps. Thus, the first column of the matrix

$$\prod_{t=kd_0 N_0}^{(k+1)d_0 N_0 - 1} \tilde{A}(t)$$

is positive and bounded away from 0 by a positive number $\underline{b} = ((1 - \eta)\gamma)^{d_0 N_0}$. Therefore, one can follow the same steps as in the proof of the Theorem 2.3.11 to conclude this result. \square

A closely related work to this result is Proposition 3.3 in [55]. In that paper, the authors studied the problem of ϵ -agreement in persistent graphs without a leader. Here, we consider the presence of a leader (or a source of news) in the network. It can be seen that under the assumptions of Theorem 2.3.13, the link between the leader and agent 1 and those between the agents in the graph $G_t^{N_0}$ are persistent. However, one cannot utilize the result in [55] to prove Theorem 2.3.13 because it assumes that there exist $a_* > 0$ and $T_* > 0$ such that $\sum_{s=t}^{t+T_*-1} w_{i,j}(s) \geq a_*, \forall t \geq 0$ and for all persistent links (i, j) . This condition is indeed equivalent to (2.21). However, we do not require this condition on the connections between the leader and the followers.

2.4 Opinion Dynamics with Two Leaders

In this section, we investigate the case where there are two leaders (or two sources of news) with different opinions T and Q . Assume that all the nodes in the network can be influenced by the two leaders with trust levels $\alpha_i(t), \beta_i(t) \in [0, 1], \forall i \in \mathcal{V}$.

The update rule is now given by

$$\mathbf{x}(t+1) = \boldsymbol{\alpha}(t)T + \boldsymbol{\beta}(t)Q + \Gamma(t)W(t)\mathbf{x}(t), \quad (2.22)$$

where $\boldsymbol{\alpha}(t) = [\alpha_1(t), \dots, \alpha_N(t)]^T$, $\boldsymbol{\beta}(t) = [\beta_1(t), \dots, \beta_N(t)]^T$ and $\Gamma(t) = I - \text{diag}(\boldsymbol{\alpha}(t) + \boldsymbol{\beta}(t))$.

In general, when both T and Q are persistent and the weight matrix W is time-varying, network agreement need not be achieved and the agents' opinions may not converge. Interesting results on opinion disagreement and fluctuation can be found in, e.g., [12, 13] and [20]. In the case where $\boldsymbol{\alpha}(t) \equiv \boldsymbol{\alpha}$, $\boldsymbol{\beta}(t) \equiv \boldsymbol{\beta}$, and $W(t) \equiv W$, the opinions converge to a fixed vector \mathbf{x}_∞ satisfying

$$(I - \Gamma W)\mathbf{x}_\infty = \boldsymbol{\alpha}T + \boldsymbol{\beta}Q.$$

In the following, we consider the case when only T is persistent.

Assumption 2.4.1. *The weights the weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ satisfy $\sum_{t=0}^{\infty} \max_{i \in \mathcal{V}} \alpha_i(t) = \infty$ and $\sum_{t=0}^{\infty} \max_{i \in \mathcal{V}} \beta_i(t) < \infty$.*

Note also that if there exists $t_\beta \in \mathbb{Z}_+$ such that $\beta_i(t) = 0, \forall i \in \mathcal{V}, \forall t \geq t_\beta$, then we can immediately make use of the results in the previous section since after time t_β there is only one persistent leader. In what follows, we assume that the presence of leader Q can last for infinite time.

Theorem 2.4.2. (Two Leaders, Strongly Connected Graph, Time-varying Weight)

Consider system (2.22) with two leaders. Suppose that $\mathcal{G}(t)$ is strongly connected for all $t \geq 0$ and that Assumptions 2.3.9 and 2.4.1 hold. Then $\mathbf{x}(t) \rightarrow T\mathbf{1}$ for any $\mathbf{x}(0)$.

Proof. Again, let $\boldsymbol{\xi}(t) = \mathbf{x}(t) - T\mathbf{1}$. Then system (2.22) becomes

$$\boldsymbol{\xi}(t+1) = \Gamma(t)W(t)\boldsymbol{\xi}(t) + (Q - T)\boldsymbol{\beta}(t). \quad (2.23)$$

Let $A(t) = \Gamma(t)W(t)$ and $\mathbf{u}(t) = (Q - T)\boldsymbol{\beta}(t)$. We note the following:

- (i) From Theorem 2.3.10, it can be verified that the unforced system $\boldsymbol{\xi}(t+1) = A(t)\boldsymbol{\xi}(t)$ with $\boldsymbol{\xi}(0) = \mathbf{x}(0) - T\mathbf{1}$ is asymptotically stable. In fact, let $\Phi(t, l)$ denote the transition matrix $\Phi(t, l) := A(t-1)A(t-2)\dots A(l+1)$, then it follows that $\lim_{k \rightarrow \infty} \Phi(t, l) = \mathbf{0}, \forall l \in \mathbb{Z}_+$.

- (ii) By Assumption 2.4.1, \mathbf{u} is absolutely summable and hence bounded.

We now show that $\lim_{t \rightarrow \infty} \boldsymbol{\xi}(t) = \mathbf{0}$. The solution to (2.23) is given by

$$\boldsymbol{\xi}(t) = \Phi(t, 0)\boldsymbol{\xi}(0) + \sum_{0 \leq k \leq t-1} \Phi(t, k+1)\mathbf{u}(k). \quad (2.24)$$

Let $\delta > 0$ be given. If $\|\boldsymbol{\xi}(0)\|_\infty = 0$, then $\|\Phi(t, 0)\|_\infty \|\boldsymbol{\xi}(0)\|_\infty = 0, \forall t$. Otherwise, from fact (i) we have $\lim_{k \rightarrow \infty} \Phi(t, 0) = \mathbf{0}$. Thus, $\exists N_1 \in \mathbb{Z}_+$ such that

$$\|\Phi(t, 0)\|_\infty \|\boldsymbol{\xi}(0)\|_\infty \leq \frac{\delta}{3}, \quad \forall t \geq N_1. \quad (2.25)$$

Next, from (ii) we have $\sum_{t \geq 0} \|\mathbf{u}(t)\|_\infty \leq |Q - T| \sum_{t \geq 0} \|\boldsymbol{\beta}(t)\|_\infty < \infty$. Therefore, $\exists N_2 \in \mathbb{Z}_+$ such that $N_2 \geq N_1$ and

$$\sum_{k \geq t} \|\mathbf{u}(k)\|_\infty \leq \frac{\delta}{3}, \quad \forall t \geq N_2. \quad (2.26)$$

Let $\bar{u} = \sup_t \|\mathbf{u}(t)\|_\infty$. From (i) we have $\exists N_3 \in \mathbb{Z}_+$ sufficiently large such that

$$\|\Phi(t, i)\|_\infty \leq \frac{\delta}{3\bar{u}N_2}, \quad \forall i \leq N_2, \quad t \geq N_2 + N_3. \quad (2.27)$$

Now for any $t \geq N_2 + N_3$, it follows from (2.24) that

$$\|\boldsymbol{\xi}(t)\|_\infty \leq \|\Phi(t, 0)\|_\infty \|\boldsymbol{\xi}(0)\|_\infty + \sum_{0 \leq k \leq t-1} \|\Phi(t, k+1)\|_\infty \|\mathbf{u}(k)\|_\infty. \quad (2.28)$$

Using (2.25), (2.26) and (2.27), and noting that $\|\Phi(t, k+1)\|_\infty \leq 1$ for any $k \leq t-1$,

we have the following for $\forall t \geq N_2 + N_3$:

$$\begin{aligned} \|\boldsymbol{\xi}(t)\|_\infty &\leq \frac{\delta}{3} + \sum_{k=0}^{t-1} \|\Phi(t, k+1)\|_\infty \|\mathbf{u}(k)\|_\infty \\ &\leq \frac{\delta}{3} + \sum_{k=0}^{N_2-1} \|\Phi(t, k+1)\|_\infty \|\mathbf{u}(k)\|_\infty + \sum_{k=N_2}^t \|\Phi(t, k+1)\|_\infty \|\mathbf{u}(k)\|_\infty \\ &\leq \frac{\delta}{3} + \sum_{k=0}^{N_2-1} \frac{\delta}{3\bar{u}N_2} \bar{u} + \sum_{k=N_2}^t \|\mathbf{u}(k)\|_\infty \\ &\leq \frac{\delta}{3} + \frac{\delta}{3} + \sum_{k=N_2}^{\infty} \|\mathbf{u}(k)\|_\infty \leq \delta. \end{aligned} \quad (2.29)$$

Since $\delta > 0$ is chosen arbitrarily, (2.29) proves that $\boldsymbol{\xi}(t) \rightarrow 0$ as $t \rightarrow \infty$. \square

It should be noted that the result of Theorem 2.4.2 is still valid when there are two or more nonpersistent leaders in the network. This result implies that only persistent ones matter. It is possible to relax the assumption on the network connectivity, but we do not proceed further in this direction.

2.5 Conclusion and Extensions

This chapter revisited the agreement seeking problem in networks with leaders, which has received a fair amount of recent attention. We developed various new

sufficient conditions for guaranteeing consensus to the persistent leader's opinion. We pointed out the important role of persistent connectivity between the leader and the others in the network. In the following, we discuss possible extensions of the our results.

First, note that model (2.2) does not present any delay explicitly. However, the framework presented in this chapter can be extended to take into account information delays as follows. Consider a generalized version of (2.2), given by

$$x_i(t+1) = \alpha_i(t)T + (1 - \alpha_i(t)) \sum_{j \in \mathcal{V}} w_{ij}(t)x_j(t - \tau_{ij}(t)), \quad \forall i \in \mathcal{V}. \quad (2.30)$$

where the delay functions τ_{ij} are assumed to be uniformly bounded, i.e.,

$$\exists \tau \geq 1 : \quad \tau_{ij}(t) \in [0, \tau - 1], \quad \forall (i, j) \in \mathcal{E}.$$

The idea is to consider an extended network \mathcal{G}^τ , which is composed of the original graph \mathcal{G} and $\tau - 1$ copies of it with each being the 1-delay in time of one another. The state of \mathcal{G}^τ is $[\mathbf{x}(t)^T, \mathbf{x}_{-1}(t)^T, \dots, \mathbf{x}_{-(\tau-1)}(t)^T]^T$ where $\mathbf{x}_{-k}(t) = \mathbf{x}(t - k)$. Note that if $\mathcal{G}(t)$ is strongly connected for any t , then every node $i \in \mathcal{V}$ is a root of a spanning tree in the union graph $(\mathcal{G}^\tau)_{[\tau]}(t)$. Thus, Theorem 2.3.11 can be applied to this union graph to establish consensus reachability of the original network \mathcal{G} . In this connection, we can conclude that consensus to the leader is also robust to bounded delays.

Second, we can also consider the case where the leader's state is time varying:

$$T(t+1) = T(t) + u(t)$$

$$x_i(t+1) = \alpha_i(t)T(t) + (1 - \alpha_i(t)) \sum_{j \in \mathcal{V}} w_{ij}(t)x_j(t), \quad \forall i \in \mathcal{V}.$$

Define the tracking error $\boldsymbol{\xi}(t) := \mathbf{x}(t) - T(t)\mathbf{1}$. Then

$$\boldsymbol{\xi}(t+1) = \Gamma(t)W(t)\boldsymbol{\xi}(t) - \mathbf{1}u(t) \quad (2.31)$$

which is a linear time-varying system with input u . Note that the unforced system $\boldsymbol{\xi}(t+1) = \Gamma(t)W(t)\boldsymbol{\xi}(t)$ is asymptotically stable under suitable conditions as in Section 2.3. Therefore, we can invoke stability results of linear time-varying systems to derive consensus conditions. One such result is the following, whose proof follows the same line of that of Theorem 2.4.2 in Section 2.4 and thus is omitted.

Proposition 2.5.1. *Consider system (2.31) and let Assumption 2.3.2 (or 2.3.9) hold. If $\sum_{t=0}^{\infty} |u(t)| < \infty$, then consensus is achieved, i.e., $\lim_{t \rightarrow \infty} |\mathbf{x}(t) - T(t)\mathbf{1}| = 0$ for any $\mathbf{x}(0)$.*

Remark 2.5.2. Note that an equivalent characterization of consensus is that the size of the convex hull of the states of all the agents (including the leader) has to be 0 in the limit. In [8], the author impose the condition of strict convex hull shrinking. The above result shows that the leader needs not to move into the convex hull of the states of regular agents at any time step in order for achieving consensus. Also, the convex hull needs not to shrink monotonically. This result could also give a hint on reducing the gap between necessary and sufficient conditions for consensus.

Chapter 3: Optimizing Leader Influence in Networks through Selection of Direct Followers

Abstract: The chapter considers the problem of a leader that aims to influence the opinions of agents in a directed network through connecting with a limited number of the agents. The aim is to select this set of agents, referred to as direct followers, to achieve the greatest possible influence on the opinions of agents throughout the network. Direct followers are simply agents that the leader decides to connect to, and the influence then occurs through the network's natural inter-agent dynamics. The problem of optimally influencing a network in the presence of another leader with a competing opinion is also considered. The problems with a single leader and in the presence of a competing leader are unified into a general combinatoric optimization problem, for which two heuristic approaches are developed. The first approach is based on a convex relaxation scheme, possibly in combination with the ℓ_1 -norm regularization technique, and the second is based on a greedy selection strategy. The main technical novelties of this work are in the establishment of supermodularity of the objective function and convexity of its continuous relaxation. As a result, the greedy approach is guaranteed to yield a lower bound on the approximation ratio that is sharper than $(1 - \frac{1}{e})$, while the convex approach can benefit from efficient

(customized) numerical solvers to have practically comparable solutions possibly with faster computation times, especially for large networks. The two approaches can be combined to provide effective tools and better analysis for optimal design of influence spreading in diffusive networks. Numerical examples are given to illustrate the usefulness of the approaches. In these examples, the approximation ratio can be made to reach 90% or higher depending on the number of direct followers.

3.1 Introduction

The notion of a *leader* is introduced in many cases to represent a special agent who has the ability to affect the states (or opinions) of other regular agents, conventionally called *followers*, while its state is uninfluenced by others- in this sense, a leader is also termed elsewhere as a *stubborn* agent). A great number of recent efforts have also been devoted to the consensus problem in networks with leaders (see, e.g., [5, 13, 18, 20, 101, 102] and references therein). In most cases, a leader is usually assumed to have a limited number of connections with other agents, due to restrictions on, e.g., the budget, communication power or channels of the leader. This gives rise to the problem of the leader choosing whom to influence directly so that the overall network can best perform (in some sense) under the restriction on the leader's connectivity.

This chapter deals with problems related to a leader selecting a limited number of agents with which to communicate in a directed network. The aim of the leader is to achieve maximum influence on the opinions of agents throughout the network.

The network agents that the leader selects to communicate with are referred to as *direct followers*. Network agents all update their opinions dynamically based on their current opinions and on opinions of immediate neighbors. Thus, through its connections with the direct followers and the inherent network dynamics, the leader influences the opinions of agents throughout the network. The leader wishes to select the limited group of direct followers so as to maximize its influence on the network, in the sense that the opinions of agents throughout the network approach the opinion of the leader either as rapidly as possible over time or as close as possible in the limit. In particular, we consider the following two problems:

- Problem (P1): Optimize the influence of a leader on the agents in a directed network, whose opinion dynamics follow the well known DeGroot model. Here, the leader's goal is to select a limited number of direct followers to connect to, in order to influence all the agents to converge to its constant opinion as quickly as possible.
- Problem (P2): Optimize the influence of one leader in the presence of another leader (with a competing opinion) over a directed network of followers under a similar connectivity constraint as in (P1). Here, the influence of a leader is measured in terms of the distance between the leader's opinion and a weighted average of the steady state opinions of all the network agents.

We unify the two problems above into a more general combinatoric optimization problem, called (P), and develop two heuristics for approximately solving problem (P), namely:

- *Convex relaxation*: which can be treated effectively by available numerical algorithms and solvers. Here, the *convexity* result is novel.
- *Greedy algorithms*: which can be carried out in polynomial time. Here, the *supermodularity* result is new and can be used to provide provable accuracy guarantees for the greedy solutions.

This chapter is related to a large body of literature on problems of leader selection, stubborn agent placement, and sensor selection (see, e.g., [20, 61–67, 103, 104] and references therein) but departs from this literature in many respects. First, we only ask that the underlying network be directed. Second, we allow selected direct follower nodes to follow inter-agent dynamics like any other agents, rather than forcing them to adopt the leader’s opinion instantaneously. Third, we allow the agents in the network to have different initial opinions (which are taken into account explicitly in the context of problem (P1)), and the agents can be weighted differently by the leader. Finally, and more importantly, although continuous relaxation and greedy heuristics have been employed in dealing with influence maximization problems, our theoretical results on convexity and supermodularity are considerably stronger than existing results, without assuming any symmetry or resorting to the random walk theory. This not only provides a deeper understanding of diffusive processes but also can be used for a broad range of applications. More detailed comparisons will be given in Section 3.2 after our problem formulations.

Other by-products of our analysis include: (i) a dynamic centrality measure in the context of problem (P1) (i.e., one in which the measure of effectiveness of the set

of chosen agents can vary with time); (ii) straightforward application to Friedkin’s model [3] (where each agent is allowed to have stubbornness in retaining its initial opinion) in the context of problem (P2); (iii) an affirmative answer to a conjecture recently proposed in [105] on optimization of on-chip thermoelectric cooling systems; and (iv) a convexity result for the state trajectory of a class of bilinear discrete-time systems.

The remainder of the chapter proceeds as follows. In Section 3.2, we introduce our network models and associated optimization problems of interest. Related works are also reviewed. Our main results are given in Sections 3.3, 3.4 and 3.5. In Section 3.3 we provide exact solutions to the problems (P1) and (P2) for the case of selecting one agent or two. The general case of multiple agents selection is treated in Sections 3.4 and 3.5. Specifically, in Section 3.4, we establish the convexity of the relaxed and approximate problems and discuss associated numerical issues in applying convex solvers to these problems. In Section 3.5, we prove the supermodularity property of the original objective functions and present two greedy algorithms that admit provable approximation ratios. Next, a few simulation results are reported in Section 3.6 for two example networks; one of small size and the other much larger. Finally, further (convexity) results and applications to another opinion dynamic model are discussed in Section 3.7.

3.2 Problem Formulation and Related Works

This section proceeds as follows. In Subsection 3.2.1 we augment the DeGroot model with a single leader, and formulate an associated problem of optimizing the leader’s influence on the opinion dynamics of the network agents. Finally, in Subsection 3.2.2, we consider a model similar to that of Subsection 3.2.1 except that we further include another leader with a differing (constant) opinion. Here, the influence of a leader is defined differently but the optimization problem shares the same structure with the previous setting.

Consider a leaderless network with N agents denoted by $\mathcal{V} = \{1, 2, \dots, N\}$. The underlying communication is characterized by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The dynamics of each agent is described by the DeGroot model (1.4), which is repeated here for convenience. Let $x_i(t) \in [0, 1]$ denote the state or opinion of node i at time $t \in \mathbb{N}_0$; At the start, each agent has an initial state $x_{0i} \in [0, 1]$. At any other time $t > 0$, each agent observes opinions of its neighbors and updates its opinion as

$$x_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t), \quad x_i(0) = x_{0i}, \quad \forall i \in \mathcal{V}, \quad (3.1)$$

where, recall that, \mathcal{N}_i denotes the set of node i ’s immediate neighbors (including itself) and $W := [w_{ij}] \in \mathbb{R}^{N \times N}$ denotes the normalized weight matrix of the network. We make the following blanket assumption, simply the combination of Assumptions 1.5.2 and 1.5.3 and presented here for convenience.

Assumption 3.2.1. (Network Connectivity and Weight Matrix) *The graph \mathcal{G} is fixed in time and strongly connected. The weight matrix W is fixed, row-stochastic*

and satisfies $w_{ij} > 0$ for $(i, j) \in \mathcal{E}, i \neq j$, and $w_{ij} = 0$ otherwise. Moreover, W has at least one positive diagonal element.

3.2.1 Formulation of Influence Optimization Problem for the Single Leader Case

Given a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with dynamics as described above, we now consider the effect of an external leader, denoted by $T \notin \mathcal{V}$, seeking to connect to the network. The leader is assumed to have a constant opinion $T \in [0, 1]$. The relationship of the leader to the network \mathcal{G} is as follows:

- For any agent $i \in \mathcal{V}$, the weight $\alpha_i \in [0, \infty]$ that it would place on the leader's opinion T if the leader selects to connect to the agent is known.¹ The connection would of course be directed from the leader to the network agent. (The reverse direction, from regular agents to the leader, would be pointless as the leader's opinion is assumed fixed and cannot be influenced.) We refer to $\boldsymbol{\alpha} := [\alpha_1, \dots, \alpha_N]^T$ as the vector of potential trust of network agents in the leader, or simply the trust vector.
- The leader knows $\boldsymbol{\alpha}$ but is only able to directly connect to up to K agents in the network \mathcal{G} . The K agents that the leader elects to connect to are called direct followers, and are cumulatively denoted in the sequel by the set \mathcal{K} . Unless otherwise stated, such connection is established at time $t = 0$ and the set \mathcal{K} remains fixed thereafter.

¹In this chapter, we allow the weight to be ∞ .

Note that $\alpha_i = 0$ indicates lack of trust or that agent i is not accessible to the leader, and $\alpha_i = \infty$ (or $\alpha_i \gg 1 > w_{ij}, \forall j \in \mathcal{N}_i$) indicates the highest possible level of trust of agent i in the leader's opinion. Without loss of generality, we make the following assumption, which means that the leader only connects to followers having nonzero trust level. (Clearly, it would be pointless for the leader to connect to an agent which would have zero trust in its opinion.)

Assumption 3.2.2. (Positive Trust Selection) *The set \mathcal{K} is such that $\mathcal{K} \neq \emptyset$ and $\mathcal{K} \subseteq \mathcal{V}_\alpha := \{i \in \mathcal{V} : \alpha_i > 0\}$.*

For each \mathcal{K} , let the corresponding selection vector $\mathbf{s}_\mathcal{K}$ be

$$[\mathbf{s}_\mathcal{K}]_i := \chi_\mathcal{K}(i), \quad \forall i \in \mathcal{V}.$$

(Recall that $\chi_\mathcal{A}$ is the indicator function of a set \mathcal{A} .) Then the update rule (3.1) for agent i in the presence of the leader becomes

$$x_i(t+1) = \frac{[\mathbf{s}_\mathcal{K}]_i \alpha_i T + \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t)}{[\mathbf{s}_\mathcal{K}]_i \alpha_i + 1}. \quad (3.2)$$

Here, it is understood that $x_i(t+1) = T$ if $[\mathbf{s}_\mathcal{K}]_i = 1$ and $\alpha_i = \infty$. In vector form, (3.2) can be expressed as

$$\mathbf{x}(t+1) = (I + \text{diag}(\boldsymbol{\alpha}_\mathcal{K}))^{-1}(\boldsymbol{\alpha}_\mathcal{K} T + W\mathbf{x}(t)) \quad (3.3)$$

where $\boldsymbol{\alpha}_\mathcal{K} := \mathbf{s}_\mathcal{K} \circ \boldsymbol{\alpha}$. Here \circ denotes the element-wise product (also known as the Hadamard product).

The following result is well known (see, e.g., [5, 20, 55]).

Theorem 3.2.3. (Consensus to Leader’s Opinion) *Let Assumptions 3.2.1 (Network Connectivity and Weight Matrix) and 3.2.2 (Positive Trust Selection) hold. Then for any $\mathbf{x}(0) \in \mathbb{R}^N$, all network agents asymptotically achieve consensus with the leader’s opinion, i.e., $\lim_{t \rightarrow \infty} x_i(t) = T, \forall i \in \mathcal{V}$. Moreover, the rate of convergence is exponential.*

This theorem asserts that all network agents will adopt the leader’s opinion asymptotically, regardless of their initial opinions. Note that asymptotic convergence can be ensured under conditions milder than Assumptions 3.2.1 and 3.2.2 (see, e.g., [5, 96] and Chapter 2).

Although the initial opinion $\mathbf{x}(0)$ and the selection of the set \mathcal{K} play no role in the final consensus value, which is the leader’s state (as long as $\boldsymbol{\alpha}_{\mathcal{K}} \neq \mathbf{0}$), they clearly affect the manner in which the agents approach this agreement, i.e., the transient behavior of system (3.2). Thus, we turn our attention to the problem of choosing K direct followers so as to minimize the transient error and convergence time of agents’ opinions in the network.

To capture this dynamic behavior, we consider the error vector $\boldsymbol{\xi}(t) := \mathbf{x}(t) - T\mathbf{1}$, which follows the dynamics

$$\boldsymbol{\xi}(t+1) = (I + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1}W\boldsymbol{\xi}(t), \quad (3.4)$$

Thus for all $t \geq 0$, $\boldsymbol{\xi}(t)$ is given by

$$\boldsymbol{\xi}(t) = ((I + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1}W)^t\boldsymbol{\xi}_0. \quad (3.5)$$

Consensus regardless of initial condition is clearly equivalent to the global asymptotic stability of the origin for (3.4), and since the system is linear and time-invariant,

consensus is also equivalent to global exponential stability. Let L be the weighted Laplacian matrix given by

$$L := I - W. \quad (3.6)$$

We have the following facts on the spectrum of the state dynamics matrix in (3.4) and the spectrum of the weighted Laplacian matrix.

Lemma 3.2.4. (Spectrum) *If Assumptions 3.2.1 and 3.2.2 hold, then*

$$(i) \ \rho((I + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1}W) < 1, \text{ and}$$

$$(ii) \ \forall \lambda \in \sigma(L + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}})), \Re(\lambda) > 0.$$

Proof. It is well-known that if A is an irreducible row substochastic matrix with the row-sum of at least one row less than one, then $\rho(A) < 1$ (see, e.g., [97, Thm 1.1, p. 24]). Using this result with $A = (I + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1}W$ yields part (i). Part (ii) follows immediately from an application of the Gershgorin Circle Theorem [95, p. 344] and [95, Cor. 6.2.9, p. 356], using the strong connectivity of \mathcal{G} and noticing that at least one diagonal entry of $L + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}})$ is shifted to the right compared to the corresponding entry of L . \square

Remark 3.2.5. Assertion (i) of the lemma is in fact equivalent to the result in Theorem 3.2.3 above (the constant linear system is exponentially stable). Part (ii) will be needed in defining our objective costs in the sequel.

Next, define $\|\xi_i\|_{l_1} := \sum_{t=1}^{\infty} |\xi_i(t)|$ (which is well defined because of exponential convergence of ξ) and consider the cumulative convergence error defined as

$$J_{\mathcal{K}}^{total} = \sum_{i \in \mathcal{V}} b_i \|\xi_i\|_{l_1},$$

where $\mathbf{b} = [b_1, \dots, b_N]^T \in \mathbb{R}_+^N$ is a weight vector chosen by the leader, which we require to satisfy $\mathbf{1}^T \mathbf{b} = 1$. The elements of the vector \mathbf{b} are measures of the relative preferences that the leader places on the opinions of all network agents. Note that we do not include $\xi_i(0)$ in $\|\xi_i\|_{l_1}$ since $\xi_i(0)$ does not depend on the leader's selection of direct followers. We say the selection \mathcal{K}_1 is better than \mathcal{K}_2 if $J_{\mathcal{K}_1}^{total} < J_{\mathcal{K}_2}^{total}$. Roughly speaking, the smaller $J_{\mathcal{K}}^{total}$, the smaller the convergence time, i.e., the faster consensus is achieved. However, since computing $J_{\mathcal{K}}^{total}$ is nontrivial, we will work with an upper bound $J_{\mathcal{K}}^{(1)}$ obtained as follows:

$$\begin{aligned}
J_{\mathcal{K}}^{total} &= \sum_{t \geq 1} \mathbf{b}^T |\xi(t)| \stackrel{(3.5)}{=} \mathbf{b}^T \sum_{t \geq 0} |(I + \text{diag}(\alpha_{\mathcal{K}}))^{-1} W)^t \xi(1)| \\
&\leq \mathbf{b}^T \sum_{t \geq 0} ((I + \text{diag}(\alpha_{\mathcal{K}}))^{-1} W)^t |\xi(1)| \\
&\stackrel{(\text{Lem. 3.2.4})}{=} \mathbf{b}^T (I - (I + \text{diag}(\alpha_{\mathcal{K}}))^{-1} W)^{-1} |\xi(1)| \\
&= \mathbf{b}^T (\text{diag}(W\mathbf{1} + \alpha_{\mathcal{K}}) - W)^{-1} \text{diag}(W\mathbf{1} + \alpha_{\mathcal{K}}) |\xi(1)| \\
&\leq \mathbf{b}^T (L + \text{diag}(\alpha_{\mathcal{K}}))^{-1} |W\xi_0| \quad =: J_{\mathcal{K}}^{(1)}. \tag{3.7}
\end{aligned}$$

Here the last inequality holds since, first, the inverse $(L + \text{diag}(\alpha_{\mathcal{K}}))^{-1}$ exists based on Lemma 3.2.4, part (ii), and, second $|\xi(1)| \leq (I + \text{diag}(\alpha_{\mathcal{K}}))^{-1} |W\xi_0|$. It can be verified that equality holds if either $\xi_0 \geq \mathbf{0}$ or $\xi_0 \leq \mathbf{0}$, i.e., if the leader's opinion T is outside the convex hull of the agents' initial opinions $\{x_i(0), i \in \mathcal{V}\}$. Therefore, $J_{\mathcal{K}}^{(1)}$ is a tight upper bound on $J_{\mathcal{K}}^{total}$. The more influential the direct followers are, the smaller $J_{\mathcal{K}}^{(1)}$ is and thus the faster consensus can be reached. Formally, in this

work we consider the following problem:

$$\begin{array}{l}
 \text{(P1)} \quad \min_{\mathcal{K} \subseteq \mathcal{V}} \quad J_{\mathcal{K}}^{(1)} = \mathbf{b}^T (L + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1} |W \boldsymbol{\xi}_0| \\
 \text{s.t.} \quad |\mathcal{K}| \leq K,
 \end{array} \tag{3.8}$$

Remark 3.2.6. The objective function $J_{\mathcal{K}}^{(1)}$ is defined in such a way that it allows the leader T to (i) weight each agent in the network differently through the weight or preference vector \mathbf{b} , (ii) take into account partial incentives or trust encoded in the vector $\boldsymbol{\alpha}$, and (iii) incorporate the role of initial opinions of all the agents in the network. As a consequence of (iii), it is possible for the leader to view $J_{\mathcal{K}}^{(1)}$ as the cost-to-go at the initial time, when the set of direct followers is first chosen, and to define the cost at any time t as

$$J_{\mathcal{K}}^{(1)}(t) = \mathbf{b}^T (L + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1} |W \boldsymbol{\xi}(t)|.$$

With this time-dependent objective cost, one can imagine a policy that achieves improved performance through re-solving a similar optimization problem at regular intervals for new sets of direct followers. (This would entail having limited term contracts with the direct followers selected at any time.) This is akin to a model predictive control strategy with an infinite horizon cost-to-go $J_{\mathcal{K}}^{(1)}(t)$ and control action being the sequence of sets of direct followers.

Remark 3.2.7. (Dynamic centrality measure for degree of influence of set of direct followers) Note that the reciprocal of $J_{\mathcal{K}}^{(1)}$, denoted by $C_{\mathcal{K}} := 1/J_{\mathcal{K}}^{(1)}$, can be viewed as a measure of the effectiveness of the set \mathcal{K} in spreading the leader's opinion. This can also be viewed in terms of the relative influence of the choice of one set of agents

\mathcal{K} vs. another, or as a centrality measure of a set \mathcal{K} of direct followers. A set \mathcal{K} is more influential than \mathcal{K}' if $C_{\mathcal{K}} > C_{\mathcal{K}'}$. What is new about our centrality measure is that $C_{\mathcal{K}}$ can be taken as a *dynamic centrality measure* through the definition $C_{\mathcal{K}}(t) := 1/J_{\mathcal{K}}^{(1)}(t)$, rather than a fixed quantity as are many existing centrality measures in the literature.

3.2.2 Formulation of Influence Optimization Problem in the Presence of a Competing Leader

Now we consider a similar model as above except that there are two leaders with different opinions T and Q trying to influence opinions of agents in network \mathcal{G} . Let $\mathcal{K}, \mathcal{L} \subseteq \mathcal{V}$ denote the sets of nodes that are directly connected to T and Q . Each node in the network has some potential trust levels $\alpha_i, \beta_i \in [0, \infty], \forall i \in \mathcal{V}$ and updates its opinion as follows:²

$$x_i(t+1) = \frac{[\mathbf{s}_{\mathcal{K}}]_i \alpha_i T + [\mathbf{s}_{\mathcal{L}}]_i \beta_i Q + \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t)}{[\mathbf{s}_{\mathcal{K}}]_i \alpha_i + [\mathbf{s}_{\mathcal{L}}]_i \beta_i + 1} \quad (3.9)$$

where $\mathbf{s}_{\mathcal{K}}$ and $\mathbf{s}_{\mathcal{L}}$ denote the selection vectors of T and Q respectively. In matrix form, (3.9) reads

$$\mathbf{x}(t+1) = (I + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}} + \boldsymbol{\beta}_{\mathcal{L}}))^{-1}(\boldsymbol{\alpha}_{\mathcal{K}} T + \boldsymbol{\beta}_{\mathcal{L}} Q + W \mathbf{x}(t)).$$

where $\boldsymbol{\alpha}_{\mathcal{K}} := \mathbf{s}_{\mathcal{K}} \circ \boldsymbol{\alpha}$ and $\boldsymbol{\beta}_{\mathcal{L}} := \mathbf{s}_{\mathcal{L}} \circ \boldsymbol{\beta}$. In our context, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are associated with the agents in the network and are assumed to be fixed over time. For given choices of \mathcal{K} and \mathcal{L} , the network \mathcal{G} need not (and usually does not) reach consensus even

²We exclude the case where $\alpha_i = \beta_i = \infty$ for some $i \in \mathcal{V}$.

under a strong connectivity assumption. In fact, the opinions converge to a fixed vector $\mathbf{x}(\infty)$ which depends only on $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and W , but not $\mathbf{x}(0)$ (see (3.10) below). Thus in this section, we will measure the influence of each leader by examining the limiting opinion $\mathbf{x}(\infty)$.

As we are interested in designing a competition strategy for one leader (T) in the presence of another (Q), without loss of generality suppose $\boldsymbol{\beta} \neq \mathbf{0}$ and $\mathbf{s}_L = \mathbf{1}$ (i.e., the set of direct followers of Q is known to T).

If $\boldsymbol{\alpha}_K = \mathbf{0}$, it is clear that $x_i(\infty) = Q, \forall i \in \mathcal{V}$ under the strong connectivity assumption on \mathcal{G} , i.e., the whole network will eventually be out of favor with leader T . Therefore we only consider $\boldsymbol{\alpha}_K \neq \mathbf{0}$. Further, we assume that

$$\text{card}(\boldsymbol{\alpha}_K) \leq K < \text{card}(\boldsymbol{\alpha})$$

where K represents the maximum number of connections that T is allowed to establish, accounting for limited communication and/or budget. We are interested in the following problem: *Given knowledge of $\boldsymbol{\alpha}, \boldsymbol{\beta}, W$ and of the largest allowed number of connections K , which nodes should leader T directly connect to in order to achieve the greatest possible influence (in a sense made precise below) on the eventual opinions of the network agents?*

Note that the limiting opinion vector $\mathbf{x}(\infty)$ satisfies

$$\mathbf{x}(\infty) = (I + \text{diag}(\boldsymbol{\alpha}_K + \boldsymbol{\beta}))^{-1}(\boldsymbol{\beta}Q + \boldsymbol{\alpha}_K T + W\mathbf{x}(\infty)).$$

Thus,

$$\mathbf{x}(\infty) = (L_\beta + \text{diag}(\boldsymbol{\alpha}_K))^{-1}(\boldsymbol{\beta}Q + \boldsymbol{\alpha}_K T), \quad (3.10)$$

where $L_\beta := L + \text{diag}(\beta) = I + \text{diag}(\beta) - W$, which is nonsingular under the strong connectivity assumption and the condition that $\alpha_{\mathcal{K}} \neq \mathbf{0}$ and $\beta \neq \mathbf{0}$ (cf. Lemma 3.2.4-ii).

We are interested in the steady state error vector $\xi(\infty) := \mathbf{x}(\infty) - T\mathbf{1}$. Since $(L_\beta + \text{diag}(\alpha_{\mathcal{K}}))^{-1}(\beta + \alpha_{\mathcal{K}}) = \mathbf{1}$, it can be verified that

$$\xi(\infty) = (L_\beta + \text{diag}(\alpha_{\mathcal{K}}))^{-1}\beta(Q - T).$$

To quantify the long term effect of T in the presence of Q , we define the following function operating on the set \mathcal{K} :

$$J_{\mathcal{K}}^{(2)} := \mathbf{b}^T |\xi(\infty)|,$$

where $\mathbf{b} \geq 0$ is a weight or preference vector, indicating the relative importance to the leader T of the final opinion of each agent in the network. Since $(L_\beta + \text{diag}(\alpha_{\mathcal{K}}))$ is a nonsingular M-matrix (cf. Lemma 3.2.4-ii), it follows that $(L_\beta + \text{diag}(\alpha_{\mathcal{K}}))^{-1}$ is a nonnegative matrix (see, e.g., Lemma A.1.3 in Appendix A.1). Thus

$$J_{\mathcal{K}}^{(2)} = \mathbf{b}^T (L_\beta + \text{diag}(\alpha_{\mathcal{K}}))^{-1}\beta|Q - T|.$$

Without loss of generality, let $T = 0$ and $Q = 1$ represent two competing opinions.

We are interested in the following problem:

Given $\alpha, \beta, \mathbf{b}, W$ and an integer $K > 0$, select \mathcal{K} such that $|\mathcal{K}| \leq K$ and the effect of T is maximized, i.e.,

$$\begin{array}{ll} \text{(P2)} & \min_{\mathcal{K} \subseteq \mathcal{V}} J_{\mathcal{K}}^{(2)} = \mathbf{b}^T (L_\beta + \text{diag}(\alpha_{\mathcal{K}}))^{-1}\beta \\ & \text{s.t. } |\mathcal{K}| \leq K \end{array} \quad (3.11)$$

This is a link creation problem (namely selection of \mathcal{K}), where partial incentives are allowed (i.e., $\boldsymbol{\alpha}, \boldsymbol{\beta} \in [0, \infty]^N$) and each agent can be weighted differently (through \mathbf{b}). In the limiting case when α_i, β_i are all either 0 or ∞ , this problem reduces to the previously studied optimal stubborn placement or leader selection problems in the literature, which we recall below. First, we give a general problem formulation that covers the cases without and with a competing leader.

Remark 3.2.8. (A unified problem formulation) Except for some minor differences, problems (P1) and (P2) described in (3.8) and (3.11) are almost the same. Our aim is thus to develop methods that can be applied to both. To this end, we embed these two problems in the following general one:

$$\boxed{\begin{array}{ll} \text{(P)} & \min_{\mathcal{K} \subseteq \mathcal{V}} J_{\mathcal{K}} = \mathbf{b}^T (L_{\boldsymbol{\beta}} + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1} \mathbf{c} \\ & \text{s.t. } |\mathcal{K}| \leq K \end{array}} \quad (3.12)$$

where \mathbf{b}, \mathbf{c} and $\boldsymbol{\beta}$ are nonnegative vectors. The optimal value will be denoted by J^* .

3.2.3 Comparison to Previous Work

3.2.3.1 Single leader case

The following model is widely used in the literature (see, e.g., [15, 62, 67, 106, 107]):

$$x_i(t+1) = \begin{cases} \tilde{\alpha}_i T + (1 - \tilde{\alpha}_i) \sum_{j \in \mathcal{V}} w_{ij} x_j(t), & i \in \mathcal{K} \\ \sum_{j \in \mathcal{V}} w_{ij} x_j(t), & i \in \mathcal{V} \setminus \mathcal{K} \end{cases} \quad (3.13)$$

which is equivalent to the one described in (3.2) with

$$\tilde{\alpha}_i = \frac{\alpha_i}{\alpha_i + 1}. \quad (3.14)$$

Based on this model, the works [62, 67, 106] consider the following associated problem:

$$\min_{\mathcal{K} \subseteq \mathcal{V}, |\mathcal{K}| \leq K} \tilde{f}(\mathcal{K}) := \mathbf{1}^T (I - D_{\mathcal{K}} W)^{-1} \mathbf{1}, \quad (3.15)$$

where $D_{\mathcal{K}} = I - \text{diag}(\tilde{\boldsymbol{\alpha}}_{\mathcal{K}})$, $\tilde{\boldsymbol{\alpha}} = \mathbf{1}$, and $\tilde{f}(\mathcal{K})$ represents the cumulative errors over time of all the agents. Note that $\tilde{f}(\mathcal{K})$ in (3.15) clearly corresponds to a special case of $J_{\mathcal{K}}^{(1)}$ with $\mathbf{b} = \mathbf{1}$ and $\boldsymbol{\xi}_0 = \mathbf{1}$. Thus, one may wonder why we use model (3.2) instead of (3.13). The main reason is that using the former model allows us to obtain a much stronger convexity result than using the latter. This is also one of the main contributions of our work.

To deal with problem (3.15), [62] uses a continuous relaxation of \tilde{f} and ℓ_1 -norm regularization technique and proves *element-wise convexity* of the so-obtained objective function. This allows the authors to employ the coordinate descent approach. However, it is important to point out that the relaxed problem formulated in [62] is not necessarily convex; see Remark 3.4.1 below for an example. In [67], supermodularity property of $\tilde{f}(\mathcal{K})$ in (3.15) is proved a greedy heuristic [108] is used to yield approximate solutions with provable accuracy.

In [65], the authors use a continuous-time version of the DeGroot model and consider the problem of selecting a set of nodes to become leaders (instantaneously) so as to minimize the convergence error, defined as the l_p -norm of the distance between the followers' states and the convex hull of the leader states. By replacing the convergence error with an upper bound that is *independent* of the initial states of the network (and is loose in general), [65] proves the supermodularity property

of so-obtained bound based on a connection with the random walk theory, and then employs the greedy approach in [108].

Kempe et al. [60] also formulate the problem of finding the influential nodes in a network as a discrete optimization problem with a submodular cost function and apply the greedy algorithm to obtain a $(1 - \frac{1}{e})$ approximate solution. However, the diffusion model in [60], called Independent Cascade, is basically different from the opinion model considered here.

3.2.3.2 Multiple leaders case

In [64], the authors consider a linear stochastic model the mean behavior of which is equivalent to the following deterministic model:

$$x_i(t) = \begin{cases} 0, & \text{if } i \in \mathcal{V}_0 \\ 1, & \text{if } i \in \mathcal{V}_1 \\ \sum_{j \in \mathcal{V}} w_{ij} x_j(t-1) & \text{else} \end{cases} \quad (3.16)$$

where $\mathcal{V}_0, \mathcal{V}_1 \subset \mathcal{V}$ are two disjoint sets of stubborn agents. This model is a limiting case of (3.9) with $\alpha_i, \beta_i \in \{0, \infty\}$ (i.e., an agent becomes stubborn if directly connected to a leader). The optimal stubborn agent placement problem studied in [64] is defined as follows: For a given set \mathcal{V}_0 with known locations in the network, choose K nodes from $\mathcal{V} \setminus \mathcal{V}_0$ to form the set \mathcal{V}_1 so that the network bias toward \mathcal{V}_1 in the limit is maximized, i.e.,

$$\max_{\mathcal{V}_1 \subset \mathcal{V}} \left\{ \sum_{i \in \mathcal{V}} x_i(\infty) : |\mathcal{V}_1| = K, \quad \mathcal{V}_0 \text{ fixed} \right\}.$$

This problem is in fact similar to a special case of (3.11) with $\mathbf{b} = \mathbf{1}$ and $\alpha_i, \beta_i \in \{0, \infty\}$. The authors prove submodularity of the objective function based on connection with a random walk and then use the greedy algorithm [108] to (approximately) solve the problem.

The work [66] considers a similar model as in [64] and defines a measure of node centrality for a given set \mathcal{V}_0 as $H(l) = \sum_{i \in \mathcal{V}} x_i(\infty | \mathcal{V}_1 = \{l\})$. The authors introduce a distributed message passing algorithm that enables each node $l \in \mathcal{V} \setminus \mathcal{V}_0$ to compute its own $H(l)$. One of our optimality criteria is also able to subsume this centrality measure as a special case. More importantly, it is considered in a more general setting and practical (centralized) algorithms are developed for the benefit of network designers or market competitors.

In [63] the following model proposed by Friedkin and Johnsen [3] is considered:

$$x_i(t) = (1 - \sigma_i) \sum_{j \in \mathcal{V}} w_{ij} x_j(t-1) + \sigma_i x_i(0).$$

Here $\sigma_i \in [0, 1]$ reflects the level of stubbornness of each agent $i \in \mathcal{V}$ regarding its initial opinion. The paper deals with the problem of selecting K nodes so that if they become fully stubborn and their opinions are set to 1, then the limiting opinions of all the agents, on average, are as positive as possible, i.e.,

$$\max_{\mathcal{V}_1 \subset \mathcal{V}} \left\{ \sum_{i \in \mathcal{V}} x_i(\infty) : |\mathcal{V}_1| = K, x_i(t) = 1, \forall t \geq 0, \forall i \in \mathcal{V}_1 \right\}.$$

The authors exploit a connection between this model and absorbing random walks to establish the submodularity of the cost function, and then rely on the greedy algorithm in [108] to approximate the optimal solution within factor $(1 - \frac{1}{e})$.

3.2.3.3 Our Contributions

Our work greatly generalizes and differs from the aforementioned works both in problem formulation and solution.

Regarding problem formulation, it should be noted that our direct followers can have dynamics like any other network node, unlike the forceful/stubborn agents in those papers. This can be viewed in terms of trust levels of the direct followers with respect to the leader's state being arbitrary in our work. Moreover, within the context of problem (P1), the agents' initial opinions need not be the same and are taken into account explicitly in the cost $J_{\mathcal{K}}^{(1)}$, which is a tight upper bound on the cumulative error of all the agents over time. This also allows us to consider a time-varying objective cost $J_{\mathcal{K}}^{(1)}(t)$ and update the set of direct followers \mathcal{K} repeatedly to further improve the network performance. Furthermore, the agents can be weighted differently by the leader in contributing to the cost $J_{\mathcal{K}}$. We believe that these settings are more natural and thus likely to be of more value for practical applications. Finally, the models considered here, i.e., (3.2) and (3.9), allow us to establish the convexity of a relaxed problem of (P), while neither (3.16) nor (3.13) does so; see also Remark 3.4.1 below.

Regarding problem solving, although we adopt two well known heuristic approaches, namely, convex relaxation/approximation technique and the greedy selection strategy, the theoretical results presented here are much more general and stronger. In particular, our technical contributions include establishment of the supermodularity property of the objective function in problem (P) and the convexity

of its continuous relaxation; both results are based on the M-matrix theory, which is completely different from tools used in [63–65, 67]. First, we prove the convexity of our relaxed problem (in the usual sense instead of just element-wise) and without assuming any kind of symmetry, which is of great benefit since it allows us to use much more effective numerical algorithms (e.g., gradient descent and Interior Point Methods) compared to the coordinate descent approach employed in [62]. Second, we derive a general matrix supermodularity inequality that can be used to prove supermodularity of $J_{\mathcal{K}}$ as well as another type of cost function encountered in the literature (see Remark 3.5.6 below). Combining the supermodularity result with the notion of curvature of a submodular function [109], we prove that the well known greedy algorithm [108] applied to our problem admits a theoretical approximation guarantee that is sharper than $(1 - \frac{1}{e})$. In addition, we develop an improved version of this algorithm that is able to achieve better accuracy. Finally, in both approaches, we derive upper and lower bounds on the optimal value, which, when combined together, provide a better analysis of the obtained approximate solutions. As will be demonstrated in our numerical examples, the approximation ratio can be ensured to be ranging from 70% to 100% depending on the value K .

3.3 Special Cases $K = 1, 2$: Optimal Solutions

For any matrix A , let $A_{(i)}$ and $A^{(j)}$ denote the i -th column and j -th row of A , respectively. Moreover, we will use both A_{ij} , $[A]_{ij}$ and a_{ij} to refer to the (ij) -th element of A .

3.3.1 Single Agent Selection

Because W is an irreducible row stochastic matrix, 0 is a simple eigenvalue of $L = I - W$ associated with right eigenvector $\mathbf{1}$. Let $\boldsymbol{\pi} \in \mathbb{R}^N$ denote the left normalized eigenvector corresponding to this eigenvalue such that $\boldsymbol{\pi}^T \mathbf{1} = 1$. It is known by Perron theorem (see [95, Thm. 8.4.4]) that $\boldsymbol{\pi}$ is strictly positive under the strong connectivity assumption on the underlying communication graph.

Now let \mathcal{K} be a singleton, i.e., $K = 1$. Then, there are at most N possible choices that leader T can take. For problem (3.8), we have the following result (where we recall that for a matrix A , $A^{(k)}$ and $A_{(k)}$ denote the k -th row and k -th column, respectively).

Theorem 3.3.1. (Single agent selection for problem (P1) in (3.8)) *Suppose \mathbf{b} satisfies the normalized condition that $\mathbf{1}^T \mathbf{b} = 1$. For any $k \in \mathcal{V}$, we have*

$$J_{\{k\}}^{(1)} = \mathbf{p}_k^T |\boldsymbol{\xi}_0|, \quad \text{with} \quad (3.17)$$

$$\mathbf{p}_k^T := (\mathbf{b}^T L^\dagger - L^{\dagger(k)}) - (\alpha_k^{-1} - L_{kk}^\dagger - \mathbf{b}^T L_{(k)}^\dagger) \frac{\boldsymbol{\pi}^T}{\pi_k}. \quad (3.18)$$

Moreover, if $\mathbf{b} = \mathbf{1}/N$, then we have

$$\mathbf{p}_k^T = (\alpha_k^{-1} + L_{kk}^\dagger) \frac{\boldsymbol{\pi}^T}{\pi_k} - L^{\dagger(k)}. \quad (3.19)$$

Proof. See Appendix A.2.1. □

Our next result characterizes the cost function corresponding to a single agent selection for problem (3.11).

Theorem 3.3.2. (Single agent selection for problem (P2) in (3.11)) *Let $P = L_\beta^{-1}$.*

For any $k \in \mathcal{V}$, we have

$$J_{\{k\}}^{(2)} = 1 - \frac{\mathbf{b}^T P_{(k)}}{\alpha_k^{-1} + P_{kk}}. \quad (3.20)$$

Proof. See Appendix A.2.2. □

As a result, when $K = 1$, optimal solutions to both problems (3.8) and (3.11) are given by

$$k^* = \arg \min_{i \in \mathcal{V}} J_{\{i\}}.$$

It should be noted that one only needs to evaluate L^\dagger and $\boldsymbol{\pi}$ or L_β^{-1} once (which requires $O(N^3)$ operations), then uses (3.18) or (3.20) to compute the influence corresponding to each follower being selected. When N is large, this is less computationally expensive than inverting matrix $(L + \text{diag}(\boldsymbol{\alpha}_\mathcal{K}))$ multiple times (each costs $O(N^3)$ operations) for different choices of \mathcal{K} .

The cost $J_{\{k\}}$ is inversely proportional to the trust level α_k . This has a practical meaning as follows. In a social network, an agent who is strongly influenced by his neighbors but is quite skeptical about new information (from the leader) may be less important in spreading the leader's opinion than one of his friends, who is easier to persuade.

Furthermore, $J_{\{k\}}^{(1)}$ also depends linearly in $|\boldsymbol{\xi}_0|$, the initial error of the whole network. As noted earlier in Remark 3.2.7, we can view $J_{\{k\}}^{(1)}$ as the cost-to-go at initial time, i.e., $J_{\{k\}}^{(1)}(0)$. In this connection, it is easy to see that the cost at any time t is given by

$$J_{\{k\}}^{(1)}(t) = \mathbf{p}_{\{k\}}^T |\boldsymbol{\xi}(t)|.$$

This suggests that the centrality of each agent should be *dynamic*. That is, an agent may be the most important at some time but may not be at the other times, depending not only on its position in the network structure but also on how it behaves over time. The significance of this is that if the leader is able to repeatedly compute $J_{\{k\}}^{(1)}(t)$, then it can further improve the performance of the network by repeatedly selecting the informed agent.

Remark 3.3.3. (Connection of $J_{\{k\}}^{(1)}$ with other centrality measures) Consider again when the graph \mathcal{G} is undirected and L is symmetric, $\mathbf{b} = \mathbf{1}/N$, and $\boldsymbol{\xi}_0 = \mathbf{1}/N$. Note that $\boldsymbol{\pi} = \mathbf{1}/N$ and that $L^\dagger \boldsymbol{\pi} = \mathbf{0}$ (see Lemma A.1.5). Hence,

$$J_{\{k\}}^{(1)} = \alpha_k^{-1} + L_{kk}^\dagger. \quad (3.21)$$

Thus $C_{\{k\}}^{(1)}$ is proportional to $1/L_{kk}^\dagger$. It is interesting to note that in [110] the authors define the *topological centrality* of a node to be $TC_k = 1/L_{kk}^\dagger$ where L^\dagger is the pseudo-inverse of a Laplacian matrix L ; see [110] for further details. Additionally, the notions of *information centrality* [111] and *node certainty* [112] can also be shown to be proportional to $1/L_{kk}^\dagger$. Notice that [110, 111] only define these notions for undirected graphs where L is symmetric. Thus when the graph \mathcal{G} is undirected and L is symmetric, these centrality indices and our $C_{\{k\}}$ are equivalent in ranking the importance of nodes in the network.

Moreover, for undirected networks, the pseudo-inverse of a Laplacian matrix also has a nice connection with the notions of *resistance distance*, that is,

$$L_{kk}^\dagger = \frac{1}{IC_k} - \frac{K_f}{N^2}$$

where $K_f = \text{tr}(L^\dagger)$ denotes the *Kirchhoff index* of the network and IC_k the *information centrality* [111] of node k given by

$$\frac{1}{IC_k} = \frac{1}{N} \sum_j r_{kj}$$

with r_{kj} being the topological distance between k and j . Therefore,

$$J_{\{k\}}^{(1)} = \frac{N}{\alpha_k} + \frac{N}{IC_k} - \frac{K_f}{N} = \frac{N}{\alpha_k} + \sum_j r_{kj} - \frac{K_f}{N}.$$

As a consequence, if $\alpha_k = \alpha_j, \forall k, j \in \mathcal{V}$, then the centrality $C_{\{k\}}^{(1)}$ agrees with the information centrality. In particular, nodes with smaller total distance to all the others will have higher centrality measures, thus more important.

It is, however, important to note that our measure $C_{\{k\}}^{(1)}$ also depends proportionally on α_k , which makes more practical sense since α_k represents the proclivity towards the leader's opinion of agent k . Moreover, $C_{\{k\}}^{(1)}$ is not restrictively defined for undirected graphs and symmetric L .

3.3.2 Two-Agent Selection

In this subsection, we derive an explicit expression to the joint centrality of any pair of agents. Let $\mathcal{K} = \{i, j\} \subset V, i \neq j$. For problem (3.8), we have the following result.

Theorem 3.3.4. (Two-agent selection for problem (P1) in (3.8)) *Let $\mathbf{b} = \mathbf{1}/N$. We have*

$$J_{\{ij\}}^{(1)} = \mathbf{p}_{ij}^T |\boldsymbol{\xi}_0|, \tag{3.22}$$

where

$$\mathbf{p}_{ij}^T = \frac{\gamma_{ii}\gamma_{jj} - \gamma_{ij}\gamma_{ji}}{\sum \gamma_{ij}} \boldsymbol{\pi}^T - \frac{\gamma_{jj} + \gamma_{ji}}{\sum \gamma_{ij}} L^{\dagger(i)} - \frac{\gamma_{ii} + \gamma_{ij}}{\sum \gamma_{ij}} L^{\dagger(j)}, \quad (3.23)$$

and $\sum \gamma_{ij} := \gamma_{jj} + \gamma_{ij} + \gamma_{ii} + \gamma_{ji}$ with

$$\begin{aligned} \gamma_{ii} &= \frac{1}{\pi_i} \left(L_{ii}^{\dagger} + \frac{1}{\alpha_i} \right), & \gamma_{ji} &= -\frac{L_{ji}^{\dagger}}{\pi_i} \\ \gamma_{jj} &= \frac{1}{\pi_j} \left(L_{jj}^{\dagger} + \frac{1}{\alpha_j} \right), & \gamma_{ij} &= -\frac{L_{ij}^{\dagger}}{\pi_j}. \end{aligned}$$

Proof. See Appendix A.2.3. □

Note that \mathbf{p}_{ij} can also be expressed as

$$\frac{\gamma_{jj} + \gamma_{ji}}{\sum \gamma_{ij}} \mathbf{p}_i + \frac{\gamma_{ii} + \gamma_{ij}}{\sum \gamma_{ij}} \mathbf{p}_j - \frac{(\gamma_{ii} + \gamma_{ij})(\gamma_{jj} + \gamma_{ji})}{\sum \gamma_{ij}} \boldsymbol{\pi},$$

where \mathbf{p}_i is given by (3.19), which is proportionate to $J_{\{i\}}^{(1)}$.

As a consequence of Theorem 3.3.4, we can determine the optimal pair at any time t as

$$\mathcal{K}^*(t) = \arg \min_{i,j \in \mathcal{V}, i \neq j} \mathbf{p}_{ij}^T |\boldsymbol{\xi}(t)|. \quad (3.24)$$

Remark 3.3.5. (A special case) Consider the case where L is symmetric and let $\boldsymbol{\xi}_0 = \mathbf{1}, \boldsymbol{\alpha} = \mathbf{1}\infty$. Note that $\boldsymbol{\pi} \in \text{span}(\mathbf{1})$ and $L^{\dagger} \boldsymbol{\pi} = \mathbf{0}$. Thus the cost $J_{\{ij\}}^{(1)}$ reduces to

$$J_{\{ij\}}^{(1)} = \frac{L_{ii}^{\dagger} L_{jj}^{\dagger} - L_{ij}^{\dagger} L_{ji}^{\dagger}}{L_{ii}^{\dagger} + L_{jj}^{\dagger} - L_{ij}^{\dagger} - L_{ji}^{\dagger}}. \quad (3.25)$$

Notice that the term in the denominator $L_{ii}^{\dagger} + L_{jj}^{\dagger} - L_{ij}^{\dagger} - L_{ji}^{\dagger} =: r_{ij}$ is usually referred to as the resistance distance of the network measured at nodes i and j , which is

identical to the topological distance between them. The term in the numerator can be expressed as

$$L_{ii}^\dagger L_{jj}^\dagger - L_{ij}^\dagger L_{ji}^\dagger = L_{ii}^\dagger L_{jj}^\dagger (1 - (\cos^\dagger(i, j))^2),$$

where

$$\cos^\dagger(i, j) = \frac{L_{ij}^\dagger}{\sqrt{L_{ii}^\dagger L_{jj}^\dagger}}.$$

Here, by following [113], we use $\cos^\dagger(i, j)$ to measure how structurally similar the roles of i and j are. The cost now reads

$$J_{\{ij\}}^{(1)} = L_{ii}^\dagger L_{jj}^\dagger \frac{(1 - (\cos^\dagger(i, j))^2)}{r_{ji}}.$$

By (3.21) and $\alpha = \mathbf{1}\infty$, we have

$$C_{\{ij\}}^{(1)} = C_{\{i\}}^{(1)} C_{\{j\}}^{(1)} \frac{r_{ji}}{1 - (\cos^\dagger(i, j))^2}.$$

Obviously, the cost $C_{\{ij\}}^{(1)}$ depends on individual centrality $C_{\{i\}}^{(1)}, C_{\{j\}}^{(1)}$, resistance distance r_{ij} and $\cos^\dagger(i, j)$ in a nonlinear fashion. However, we can loosely infer that to minimize $J_{\{ij\}}$, the optimal selection should satisfy the following

- *Self-centrality:* $C_{\{i\}}^{(1)}$ and $C_{\{j\}}^{(1)}$ should be large.
- *Relative distance:* r_{ij} should be large, i.e., i and j should be far apart.
- *Topological similarity:* $\cos^\dagger(i, j)$ should be large, i.e., i and j should have similar roles in the network.

Consider, for example, an unweighted undirected cycle graph where α_i is identical for any node. An optimal choice (i^*, j^*) is any two nodes that are of the farthest

distance.³ Consider a network which consists of two communities as another example. A reasonable candidate for the optimal solution would be i^* and j^* where each node is the most influential in each community.

Remark 3.3.6. In connection with other centrality measures (e.g., information centrality, topological centrality), the cost $J_{\{ij\}}^{(1)}$ can be described by

$$\begin{aligned} J_{\{ij\}}^{(1)} &= L_{ii}^\dagger L_{jj}^\dagger \frac{(1 - (\cos^\dagger(i, j))^2)}{r_{ji}} \\ &= \left(\frac{1}{IC_i} - \frac{K_f}{N^2}\right) \left(\frac{1}{IC_j} - \frac{K_f}{N^2}\right) \frac{(1 - (\cos^\dagger(i, j))^2)}{r_{ji}} \\ &= \left(\frac{R_i}{N} - \frac{K_f}{N^2}\right) \left(\frac{R_j}{N} - \frac{K_f}{N^2}\right) \frac{(1 - (\cos^\dagger(i, j))^2)}{r_{ji}} \end{aligned}$$

where $R_i = \sum_k r_{ik}$ is the sum of all resistance distances from node i to all the others, which is reciprocal to the information centrality of node i .

The derivations of $C_{\{ij\}}^{(1)}$ in previous remarks are valid only under special assumptions about the trust vector α , the initial condition $\xi(0)$ and the structure of the network as well as the Laplacian matrix L . More importantly, it only indicates the importance of nodes at initial time 0. As the agents' opinions evolve with time, so do their influence measures with respect to the whole performance of the network.

This can be captured by our time-varying objective function $J_{\mathcal{K}}^{(1)}(t)$ or $C_{\mathcal{K}}^{(1)}(t)$, which

³This can be seen as follows. Let N denote the number of nodes in the cycle. For any 2 nodes in the cycle, there are exactly 2 disjoint paths connecting them. Let x, y denote the lengths of the 2 paths, which obviously satisfy $x+y = N$. Since all the nodes are identical to each other, so the joint centrality only depends on the relative distance r_{ij} where $r_{ij} = (x^{-1} + y^{-1})^{-1} = \frac{xy}{N} \leq \frac{(x+y)^2}{2N} = \frac{N}{2}$. Therefore, r_{ij} is maximized when $x = y = \frac{N}{2}$ for even N or $(x, y) = (\frac{N-1}{2}, \frac{N+1}{2})$ for N odd. This proves the claim.

is defined for any initial condition and any trust and bias vectors. This is one of the main differences between our work and others.

In a similar fashion, we can compute the cost function associated with any pair of agents for the case of problem (P2) as follows.

Theorem 3.3.7. (Two-agent selection for problem (P2) in (3.11)) *Let $P = L_\beta^{-1}$ and*

$$\begin{aligned} \nu_{ii} &= \alpha_i^{-1} + P_{ii}, & \nu_{ji} &= -P_{ji} \\ \nu_{jj} &= \alpha_j^{-1} + P_{jj}, & \nu_{ij} &= -P_{ij}. \end{aligned}$$

We have

$$J_{\{ij\}}^{(2)} = \frac{\mathbf{b}^T P_{(i)}(\nu_{jj} + \nu_{ij}) + \mathbf{b}^T P_{(j)}(\nu_{ii} + \nu_{ji})}{\nu_{ii}\nu_{jj} - \nu_{ij}\nu_{ji}}. \quad (3.26)$$

Proof. The proof is based on the rank-2 update of matrix inversion by the Woodbury identity (A.1). □

Of course, this objective function depends on the network structure and weight matrix (encoded in L) as well as the trust vectors α, β . Although connections with other notions of centrality may not be inferred easily, there is a close relation between this cost function and the average voltage of a network of resistors. In particular, assume the graph \mathcal{G} is undirected and consider the network of $|\mathcal{E}|$ resistors corresponding to the graph with w_{ij} representing the conductance between nodes i and j . Let Q and T denote two voltage sources and let α_i (β_i) denote the conductance between node i and T (Q) when there is a link, that is, when node i is selected by T (or Q). Then it can be seen that $J_{\{ij\}}^{(2)}$ is the weighted average (corresponding to \mathbf{b}) of the node voltages in the resistor network.

We close this section with the following remark. As we have seen in this section, the joint-centrality measure of a set becomes more complicated to express as K increases. Moreover, for large networks, finding the optimal solution by sweeping through all the possible combinations is a challenging or even impractical task. Therefore, we content ourselves with approximate solutions whenever they are attainable with certain quality. In this connection, we now develop two practical approaches for the general problem-(P) where lower and upper bounds on the optimal value can be obtained and used to assess approximate solutions.

3.4 General Case: Convexification Approach

In this section, we study the convexity property of the continuous relation defined by $J_{\mathcal{K}}$ and discuss numerical methods that can be used to solve the relaxed or approximate problem. We emphasize that no symmetry conditions on the Laplacian matrix L (even on its structure).

3.4.1 Convexity of Relaxation

Consider problem (3.12), equivalently put as follows:

$$\begin{aligned}
 & \min_{\mathbf{s} \in \mathbb{R}^N} f(\mathbf{s}) := \mathbf{b}^T (L_{\beta} + \text{diag}(\mathbf{s} \circ \boldsymbol{\alpha}))^{-1} \mathbf{c} \\
 \text{(P)} \quad & \text{s.t. } s_i \in \{0, 1\} \quad \forall i = 1, \dots, N, \\
 & \text{card}(\mathbf{s}) \leq K
 \end{aligned} \tag{3.27}$$

where $\mathbf{b}, \mathbf{c} \in \mathbb{R}_+^N \setminus \{\mathbf{0}\}$. Recall that $L_{\beta} = L + \boldsymbol{\beta}$. We will also use $L_0 := L$ to signify the case of problem (3.8) i.e., $\boldsymbol{\beta} = \mathbf{0}$. The optimal value of this problem is denoted

by f_P^* .

First, this problem is clearly combinatoric in nature (hence nonconvex) and generally hard to solve especially for large networks. We defer our discussion on the convexity of the objective function f for now and discuss techniques to handle the cardinality constraint instead. The first idea is to consider instead of (P) a relaxed version (P_Rlxd) defined as follows:

$$\begin{aligned}
 \text{(P_Rlxd)} \quad & \min_{\mathbf{y} \in \mathbb{R}^N} f(\mathbf{y}) \\
 & \text{s.t. } \mathbf{y} \in [0, 1]^N \text{ and } \mathbf{1}^T \mathbf{y} \leq K.
 \end{aligned} \tag{3.28}$$

This is a continuous relaxation of (P). The optimal value for (P_Rlxd), denoted by $f_{P_Rlxd}^*$, is clearly a lower bound for that of (P), i.e., $f_{P_Rlxd}^* \leq f_P^*$. Of course this lower bound is useful if an optimal solution \mathbf{y}_{P_Rlxd} is computable. In that case, if \mathbf{y}_{P_Rlxd} is a binary vector, then it is also the optimal one for (P). However, a binary solution is not to be expected as \mathbf{y}_{P_Rlxd} tends to be dense. In general, we can use a simple projection onto the feasible set of problem (P) to obtain an approximation (e.g., rounding up to 1 the K largest elements of \mathbf{y}_{P_Rlxd} and zeroing out the rest), resulting in an upper bound on f_P^* , which we denote by \bar{f}_{P_Rlxd} .

Next, we consider another practical approximation using the well known ℓ_1 -norm regularization technique. Here, we consider the problem

$$\begin{aligned}
 \text{(P_Aprx)} \quad & \min_{\mathbf{y} \in \mathbb{R}^N} g(\mathbf{y}) := f(\mathbf{y}) + \mu \mathbf{1}^T \mathbf{y} \\
 & \text{s.t. } \mathbf{y} \in [0, 1]^N =: \Omega
 \end{aligned} \tag{3.29}$$

where μ is a positive parameter the role of which is to promote sparsity of the solution. (Note that if $\mu = 0$, then clearly $\mathbf{y} = \mathbf{1}$ is the global solution to this ap-

proximate problem (see also Theorem 3.4.3 below); increasing μ is a way to penalize the number of nonzero elements in the solution.) Let $\mathbf{s}_{P_Aprx}^*$ be the binary vector corresponding to the K largest elements of a solution to problem (P_Aprx). Then $f_{P_Aprx} := f(\mathbf{s}_{P_Aprx}^*)$ is clearly an upper bound on the optimal value of the original problem (P). As a result, the gap $(f_{P_Aprx} - f_{P_Rlxd}^*)$ can also be used to evaluate the quality of our approximations.

We now discuss convexity of the function f , which would clearly be pertinent for problem (P_Rlxd) as well as problem (P_Aprx). Note that we do not assume any symmetry conditions on the Laplacian matrix L (even on its structure), or on the nonnegative vectors \mathbf{b} and \mathbf{c} (trivial cases such as $\mathbf{b} = \mathbf{0}$ or $\mathbf{c} = \mathbf{0}$ are excluded). For somewhat similar cost functions that are convex under symmetry of L , see, e.g., [61,104,105]. As noted earlier, the functions \tilde{f} in (3.15) and $J_{\mathcal{K}}^{(1)}$ defined for problem (P1) with $\mathbf{b} = \mathbf{1}$ and $\boldsymbol{\xi}_0 = \mathbf{1}$ are equivalent through the change of variables (3.14), one may wonder whether the continuous relaxation of \tilde{f} is convex. The following remark provides a negative answer for this question.

Remark 3.4.1. (Nonconvexity of continuous relaxation of \tilde{f} of (3.15)) We use a simple example with $N = 2$ to show that the continuous relaxation of \tilde{f} of (3.15) using (3.28). By abuse of notation, consider

$$\tilde{f}(\mathbf{s}) = \mathbf{1}^T (I - (I - \text{diag}(\mathbf{s} \circ \tilde{\boldsymbol{\alpha}}))W)^{-1} \mathbf{1}, \quad \mathbf{s} \in \Omega = [0, 1]^N.$$

Suppose $W = \begin{bmatrix} 0.1 & 0.9 \\ 0.5 & 0.5 \end{bmatrix}$ and $\tilde{\alpha} = \begin{bmatrix} 0.8 \\ 0.8 \end{bmatrix}$. We have

$$\nabla^2 \tilde{f}(\mathbf{1}/2) = \begin{bmatrix} 6.9101 & 16.6656 \\ 16.6656 & 22.2587 \end{bmatrix}$$

which is not positive definite as it has a negative eigenvalue, namely $\lambda = -3.7632$.

Thus, \tilde{f} is nonconvex on Ω .

In contrast to \tilde{f} , the continuous relaxation of $J_{\mathcal{K}}^{(1)}$ is convex on Ω . In fact, more is true; we establish below the convexity of f in (3.28), which is the main result of this subsection. The convexity proof relies on the following technical lemma.⁴

Lemma 3.4.2. *Let $A \in \mathbb{R}^{N \times N}$ be nonnegative and $V \in \mathbb{R}^{N \times N}$ be diagonal. Then for each $m \geq 0$,*

$$\sum_{i+j+k=m} A^i V A^j V A^k$$

is a nonnegative matrix, where i, j, k are nonnegative integers.

Proof. By change of variables, we have

$$\sum_{i+j+k=m} A^i V A^j V A^k = \sum_{0 \leq q \leq r \leq m} A^q V A^{r-q} V A^{m-r}.$$

Writing $V = \text{diag}(v_1, \dots, v_n)$, the st -th coefficient of the matrix above is

$$\sum_{0 \leq q \leq r \leq m} \sum_{1 \leq i, j \leq n} [A^q]_{si} [A^{r-q}]_{ij} [A^{m-r}]_{jt} v_i v_j \quad (3.30)$$

To simplify this expression, let us consider the graph generated by matrix A , where a_{ij} denotes the weight of the directed edge $i \rightarrow j$.⁵ Let \mathcal{P}_m denote the set of all

⁴We thank Prof. Terrence Tao for the idea for the proof of Lemma 3.4.2.

⁵The edge direction defined within this proof is in reverse order to our usual notation.

walks of length m from node $s_0 = s$ to $s_m = t$, i.e., those of the form

$$s = s_0 \xrightarrow{e_1} s_1 \xrightarrow{e_2} \dots \xrightarrow{e_m} s_m = t,$$

where $e_i = (s_{i-1}s_i)$ denotes the directed edge from node s_{i-1} to s_i . Now for a fixed tuple $(qirj)$, consider a collection $P_{(qirj)} \subset \mathcal{P}_m$ that satisfies the conditions $s_q = i$ and $s_r = j$ (i.e., fixing positions q and r). Then the term under the double summation in (3.30) represents the total weight of all the walks⁶ in $P_{(qirj)}$ multiplied by $v_{s_q}v_{s_r}$, i.e.,

$$[A^q]_{si}[A^{r-q}]_{ij}[A^{m-r}]_{jt}v_iv_j = \sum_{\{e_k\}_1^m \in P_{(qirj)}} a_{e_1}a_{e_2} \dots a_{e_m}v_{s_q}v_{s_r},$$

where we have assumed $A = [a_{kl}]_{1 \leq k, l \leq n}$. Summing the right side of this relation over $1 \leq i, j \leq n$ yields the total weight of all the walks in \mathcal{P}_m (each being scaled by $v_{s_q}v_{s_r}$), namely,

$$\sum_{\{e_k\}_1^m \in \mathcal{P}_m} a_{e_1}a_{e_2} \dots a_{e_m}v_{s_q}v_{s_r}.$$

As a result, (3.30) becomes

$$\sum_{0 \leq q \leq r \leq m} \sum_{\{e_k\}_1^m \in \mathcal{P}_m} a_{e_1}a_{e_2} \dots a_{e_m}v_{s_q}v_{s_r} = \sum_{\{e_k\}_1^m \in \mathcal{P}_m} a_{e_1}a_{e_2} \dots a_{e_m} \sum_{0 \leq q \leq r \leq m} v_{s_q}v_{s_r} \quad (3.31)$$

Note that $a_{e_i} \geq 0$ for any i and that

$$2 \sum_{0 \leq q \leq r \leq m} v_{s_q}v_{s_r} = \left(\sum_{0 \leq i \leq m} v_{s_i} \right)^2 + \sum_{0 \leq i \leq m} v_{s_i}^2 \geq 0.$$

It then follows that the right side of (3.31) is nonnegative, thereby completing the proof. □

⁶A walk's weight is defined as the product of the weights of all the edges along the walk.

We are now ready to establish the convexity as well as other important properties of our objective functions.

Theorem 3.4.3. (Properties of objective function in (3.27)) *For any $\mathbf{b}, \mathbf{c}, \boldsymbol{\alpha} \in \mathbb{R}_+^N \setminus \{\mathbf{0}\}$ and $\boldsymbol{\beta} \in \mathbb{R}_+^N$, let Ω be given as in (3.29) and consider $f : \mathbb{R}_+^N \rightarrow \mathbb{R} \cup \{\infty\}$ defined in (3.27), i.e.,*

$$f(\mathbf{y}) := \mathbf{b}^T (L_\beta + \text{diag}(\mathbf{y} \circ \boldsymbol{\alpha}))^{-1} \mathbf{c}. \quad (3.32)$$

Then f is positive, convex and decreasing on Ω . It is smooth on $\Omega \setminus \{\mathbf{0}\}$ with gradient ∇f and Hessian \mathbf{H} given by

$$\nabla f(\mathbf{y}) = -(Y^{-T} \mathbf{b}) \circ \boldsymbol{\alpha} \circ (Y^{-1} \mathbf{c}) \quad \text{with} \quad (3.33)$$

$$Y := L_\beta + \text{diag}(\mathbf{y} \circ \boldsymbol{\alpha})$$

and

$$\mathbf{H}(\mathbf{y}) = H(\mathbf{y}) + H^T(\mathbf{y}) \quad \text{with} \quad (3.34)$$

$$H(\mathbf{y}) := \text{diag}(\boldsymbol{\alpha} \circ (Y^{-T} \mathbf{b})) Y^{-1} \text{diag}(\boldsymbol{\alpha} \circ (Y^{-1} \mathbf{c})).$$

Moreover, $\mathbf{H}(\mathbf{y})$ is a nonnegative matrix and

$$0 \preceq \mathbf{H}(\mathbf{y}) \preceq L_f I, \quad \text{with} \quad L_f := \rho(\mathbf{H}(\mathbf{0})). \quad (3.35)$$

Furthermore, $L_f \leq N \max_{ij} [\mathbf{H}(\mathbf{0})]_{ij}$.

Proof. Smoothness of f follows from its definition. Positiveness follows from assumptions $\mathbf{b}, \mathbf{c}, \boldsymbol{\beta} \geq \mathbf{0}$ and the fact that $Y = L_\beta + \text{diag}(\mathbf{y} \circ \boldsymbol{\alpha})$ is a nonsingular M-matrix whenever $\mathbf{y} \in \Omega$ and $\boldsymbol{\beta}$ are not both equal $\mathbf{0}$, which ensures that Y^{-1} is a nonnegative matrix (see Lemma A.1.3 in Appendix A.1). Hence $f(\mathbf{y}) = \mathbf{b}^T Y^{-1} \mathbf{c} \geq 0$

for all $\mathbf{y} \in \Omega$. Next, we find the first differential of f , namely,

$$\begin{aligned}
df(\mathbf{y}) &= \mathbf{b}^T dY^{-1} \mathbf{c} \\
&= -\mathbf{b}^T Y^{-1} \text{diag}(\boldsymbol{\alpha}) \text{diag}(Y^{-1} \mathbf{c}) d\mathbf{y}, \\
&= -\left[(Y^{-T} \mathbf{b}) \circ \boldsymbol{\alpha} \circ (Y^{-1} \mathbf{c}) \right]^T d\mathbf{y}, \tag{3.36}
\end{aligned}$$

where we have used the fact that $dY^{-1} = -Y^{-1}(dY)Y^{-1}$, $dY = d(L_\beta + \text{diag}(\mathbf{y} \circ \boldsymbol{\alpha})) = \text{diag}(d\mathbf{y} \circ \boldsymbol{\alpha})$, and $\text{diag}(\mathbf{x})\mathbf{y} = \text{diag}(\mathbf{y})\mathbf{x} = \mathbf{x} \circ \mathbf{y}$. Therefore,

$$\nabla f(\mathbf{y}) = -\left(Y^{-T} \mathbf{b} \right) \circ \boldsymbol{\alpha} \circ \left(Y^{-1} \mathbf{c} \right).$$

Since $Y^{-1} \geq \mathbf{0}_{N \times N}$, we have $\nabla f(\mathbf{y}) \leq \mathbf{0}$, which implies that f is decreasing in \mathbf{y} . In fact, a stronger statement holds, that is, $Y^{-1} = (L_\beta + \text{diag}(\mathbf{y} \circ \boldsymbol{\alpha}))^{-1}$ is nonnegative and decreasing in \mathbf{y} . As a result, $\|\nabla f(\mathbf{y})\|_2 \leq \|\nabla f(\mathbf{0})\|_2, \forall \mathbf{y} \in \Omega$. When $\boldsymbol{\beta} \neq \mathbf{0}$, $\|\nabla f(\mathbf{0})\|_2 < \infty$, thus f is Lipschitz continuous with parameter $\|\nabla f(\mathbf{0})\|_2$ on Ω .

Next, we find the second differential of f as follows:

$$\begin{aligned}
d^2 f(\mathbf{y}) &= 2\mathbf{b}^T Y^{-1} \text{diag}(d\mathbf{y} \circ \boldsymbol{\alpha}) Y^{-1} \text{diag}(d\mathbf{y} \circ \boldsymbol{\alpha}) Y^{-1} \mathbf{c}, \tag{3.37} \\
&= 2d\mathbf{y}^T \text{diag}(\boldsymbol{\alpha} \circ (Y^{-T} \mathbf{b})) Y^{-1} \text{diag}(\boldsymbol{\alpha} \circ (Y^{-1} \mathbf{c})) d\mathbf{y}, \\
&= d\mathbf{y}^T (H + H^T) d\mathbf{y} \tag{3.38}
\end{aligned}$$

with H defined as in (3.34). Thus, $\mathbf{H} = (H + H^T)$ is the Hessian of f . Clearly, H is nonnegative since $Y^{-1}, W, \mathbf{b}, \mathbf{c}$ are so. This proves that \mathbf{H} is also nonnegative.

For convexity, it suffices to show that $d^2 f$ given by (3.37) is positive semidefinite on $\Omega \setminus \{\mathbf{0}\}$. Indeed, since \mathbf{b} and \mathbf{c} are nonnegative, we will prove that

$$Y^{-1} V Y^{-1} V Y^{-1} \geq \mathbf{0}_{N \times N},$$

where $V = \text{diag}(\mathbf{d}\mathbf{y} \circ \boldsymbol{\alpha})$. Note that Y is a nonsingular M-matrix. Thus, by definition, $Y = s(I - A)$ for some positive s and some nonnegative matrix A with $\rho(A) < 1$. Then we have $Y^{-1} = s^{-1} \sum_{i=0}^{\infty} A^i$ and hence

$$\begin{aligned} Y^{-1}VY^{-1}VY^{-1} &= s^{-3} \sum_{i \geq 0} \sum_{j \geq 0} \sum_{k \geq 0} A^i V A^j V A^k \\ &= s^{-3} \sum_{m \geq 0} \sum_{i+j+k=m} A^i V A^j V A^k. \end{aligned} \quad (3.39)$$

Now by Lemma 3.4.2, $\sum_{i+j+k=m} A^i V A^j V A^k \geq \mathbf{0}_{N \times N}$ for each $m \geq 0$. Therefore, $Y^{-1}VY^{-1}VY^{-1} \geq \mathbf{0}$, thereby proving convexity of f .

Next, to prove (3.35), we use the inequality

$$\mathbf{x}^T \mathbf{H}(\mathbf{y}) \mathbf{x} \leq \rho(\mathbf{H}(\mathbf{y})) \mathbf{x}^T \mathbf{x}, \quad \forall \mathbf{x} \in \mathbb{R}^N, \mathbf{y} \in \Omega, \quad (3.40)$$

which holds since $\rho(\mathbf{H}(\mathbf{y}))$ is the largest eigenvalue of the nonnegative (and symmetric) matrix $\mathbf{H}(\mathbf{y})$ (see Theorem A.1.1 in Appendix A.1). Note also that $H(\mathbf{y})$ is decreasing in $\mathbf{y} \in \Omega$. Thus we have

$$\mathbf{0}_{N \times N} \leq \mathbf{H}(\mathbf{y}) \leq \mathbf{H}(\mathbf{0}) \leq \max_{ij} [\mathbf{H}(\mathbf{0})]_{ij} \mathbf{1} \mathbf{1}^T.$$

Finally, by Theorem A.1.2 in Appendix A.1, we have $\rho(\mathbf{H}(\mathbf{y})) \leq \rho(\mathbf{H}(\mathbf{0})) = L_f \leq \max_{ij} [\mathbf{H}(\mathbf{0})]_{ij} \rho(\mathbf{1} \mathbf{1}^T) = N \max_{ij} [\mathbf{H}(\mathbf{0})]_{ij}$. \square

Consider again the example in Remark 3.4.1 and choose W and $\boldsymbol{\alpha}$ satisfying (3.14), e.g., $W = \tilde{W}$ and $\boldsymbol{\alpha} = 4 \times \mathbf{1}$. With $\boldsymbol{\beta} = \mathbf{0}$, $\mathbf{b} = \mathbf{c} = \mathbf{1}$, we have $f(\mathbf{y}) = \mathbf{1}^T (L + 4 \text{diag}(\mathbf{y}))^{-1} \mathbf{1}$ and

$$\nabla^2 f(\mathbf{1}/2) = \mathbf{H}(\mathbf{1}/2) = \begin{bmatrix} 2.5952 & 0.7958 \\ 0.7958 & 3.8131 \end{bmatrix} \succ 0.$$

Remark 3.4.4. (Lipschitz constant for problem (3.8)) When $\beta = \mathbf{0}$ we have $L_\beta = L$, which is singular. As a result, the Lipschitz constant $L_f = \rho(\mathbf{H}(\mathbf{0})) = \infty$. Indeed, when $\alpha = \beta = \mathbf{0}$, the agents' opinions converge to a consensus value that is unaffected by either T or Q .

The following result is an immediate consequence, whose proof is omitted.

Corollary 3.4.5. (Properties of g in (3.29)) *The function g is smooth and convex over constraint set Ω with gradient*

$$\nabla g(\mathbf{y}) = \nabla f(\mathbf{y}) + \mu \mathbf{1}, \quad (3.41)$$

which is Lipschitz continuous with Lipschitz constant $L_g = L_f$. Moreover, if $\eta := \min_{\mathbf{y} \in \Omega} \lambda_{\min}(\mathbf{H}(\mathbf{y})) > 0$, then g is strongly convex with parameter η .

It now becomes obvious that both problems (P_Rlxd) and (P_Aprx) are convex with a (possibly strongly) convex smooth cost function. Thus, they can be solved by various algorithms, including Interior Point Methods (IPMs), or the Projected Gradient Method (PGM) (see e.g., [114–118]), provided that $\nabla f(\mathbf{y})$ can be evaluated efficiently (see Remark 3.4.11 below).

We now remark on how to deal with the original problem (P) in connection with tuning the parameter μ in (P_Aprx).

Remark 3.4.6. (On selecting regularization parameter μ) From the optimal solution $\tilde{\mathbf{y}}^*$ of problem (3.29) for a particular μ , we can obtain an approximate solution to the original problem (3.27) by choosing nodes corresponding to the K largest entries of $\tilde{\mathbf{y}}^*$. As μ increases, there (usually) exists $\bar{\mu}$ such that $\text{card}(\tilde{\mathbf{y}}^*) \leq K$. Once this

value is found (which can be done fairly easily), μ can be tuned manually within the interval $[0, \bar{\mu}]$ to find the best approximation.

We conclude this subsection with the following remark, showing an application of our analysis developed above.

Remark 3.4.7. (A proof of a conjecture in [105]) In a less apparently related context, the authors in [105] study an on-chip active cooling system (based on super-lattice thin-film thermoelectric coolers) and the problem of minimizing the steady state temperature profile. The following conjecture was posed and supported by extensive simulations.

Conjecture 3.4.8. ([105]) Suppose $H^{-1} \in \mathbb{R}^{N \times N}$ is a Stieljes matrix.⁷ Then, for any $1 \leq k, l \leq N$ and $\mathbf{z} \in \mathbb{R}^N$, the following holds: $\mathbf{z}^T \text{diag}(H^{(k)})H \text{diag}(H_{(l)})\mathbf{z} \geq 0$.

Assuming this conjecture to be valid, the paper then shows the convexity of each element h_{kl} of matrix $H(x) = (G - xD)^{-1}$ as a function of $x \in [0, x_m]$, where D is a diagonal matrix with at least one positive entry, G is an irreducible Stieljes matrix, and $x_m > 0$ such that $G - xD$ is positive definite for all $x \in [0, x_m]$. This convexity result was proved in a later work [119] by using results on the convexity of parameterized linear equations [120] but the conjecture has not yet been proved. We will prove the conjecture next.

Although our cost function does not resemble $H(x)$ in the mentioned papers, our analysis provided above can be used to give an affirmative answer to the conjecture, even under a weaker assumption, namely that H^{-1} is an M-matrix. Indeed

⁷A Stieljes matrix is a real symmetric positive definite M-matrix.

the proof below does not require a symmetry assumption.

Proof of Conjecture 3.4.8. Let $V = \text{diag}(\mathbf{z})$. We have

$$\begin{aligned} \mathbf{z}^T \text{diag}(H^{(k)})H\text{diag}(H_{(l)})\mathbf{z} &= H^{(k)} \text{diag}(\mathbf{z})H\text{diag}(\mathbf{z})H_{(l)} \\ &= \mathbf{e}_k^T HVHVH\mathbf{e}_l. \end{aligned}$$

Since H^{-1} is an M-matrix, it follows that $H^{-1} = s(I - A)$ for some $s > 0$ and some $A \geq \mathbf{0}_{N \times N}$ with $\rho(A) < 1$. Thus $H = s^{-1} \sum_{i \geq 0} A^i$, and hence using the same expansion as in (3.39) yields $s^3 HVHVH = \sum_{m \geq 0} \sum_{i+j+k=m} A^i V A^j V A^k$, which is nonnegative by Lemma 3.4.2. Therefore, $\mathbf{e}_k^T HVHVH\mathbf{e}_l \geq 0$ as desired. \square

The foregoing proof suggests that the convexity results in our work can be useful in studying various applications, such as those considered in [105, 119].

3.4.2 Numerical Methods

We now discuss two numerical algorithms that can be used to solve problem (P_Aprx), namely the Projected Gradient Method and Interior Point Methods. Problem (P_Rlxd) can be treated similarly.

Theorem 3.4.9 (PGM). *Consider the Projected Gradient Method applied to problem (3.29):*

$$\mathbf{y}^{(t+1)} = P_\Omega \left[\mathbf{y}^{(t)} - \gamma^{(t)} (\nabla f(\mathbf{y}^{(t)}) + \mu \mathbf{1}) \right] \quad (3.42)$$

where P_Ω denotes the projection operator onto the constraint set Ω of (3.29), and $\gamma^{(t)}$ step size. If $\gamma^{(t)}$ is chosen by the Armijo rule, then every limit point of $\{\mathbf{y}^{(t)}\}$ is an optimal solution to problem (3.29). If $\beta \neq \mathbf{0}$, we can use any constant step size

$\gamma^{(t)} \equiv \gamma \in (0, L_f)$. If $\eta > 0$, then for $\gamma = \frac{1}{L_f}$, $\mathbf{y}^{(t)}$ converges linearly to the unique solution \mathbf{y}^* with rate $(1 - \frac{\eta}{L_f})^{\frac{1}{2}}$.

Proof. The theorem follows from Propositions 2.3.1 and 2.3.2 in [115], and Theorem 2.2.8 in [117]. \square

Remark 3.4.10. (On implementation of PGM when $\boldsymbol{\beta} = \mathbf{0}$) This corresponds to Problem (3.8). We have that f is well-defined and smooth on $\Omega \setminus \{\mathbf{0}\}$ ($f(\mathbf{0}) = \infty$). As a result, given $\mathbf{y}^{(0)} \neq \mathbf{0}$, the level set $\Omega^0 = \{\mathbf{y} \in \Omega | g(\mathbf{y}) \leq g(\mathbf{y}^{(0)})\}$ is convex compact set excluding $\mathbf{0}$, over which g , ∇g and $\nabla^2 g = \mathbf{H}$ are continuous. In particular, ∇g is Lipschitz continuous on Ω^0 with coefficient $L_g^0 = \max_{\mathbf{y} \in \Omega^0} \|\mathbf{H}(\mathbf{y})\|$. Thus, we can replace P_Ω by P_{Ω^0} or choose a step size ensuring that $\mathbf{y}^{(t)} \in \Omega^0$, then the PGM iteration (3.42) still works in this case (i.e., $\boldsymbol{\beta} = \mathbf{0}$).

Remark 3.4.11. (On gradient evaluation) Gradient $\nabla f(\mathbf{y})$ involves inversion of $Y = (L_\beta + \text{diag}(\mathbf{y} \circ \boldsymbol{\alpha}))$, which usually costs $O(N^3)$ operations and $O(N^2)$ memory storage, and thus does not scale well with network size. Moreover, even if the underlying graph is sparse, this inversion can yield a dense matrix and therefore storing it could also be too expensive for very large networks. In such a case, one way to reduce those costs is to exploit the sparsity of the graph and the structure of the cost function. In particular, from (3.33), we have $\nabla f(\mathbf{y}) = -\mathbf{u} \circ \boldsymbol{\alpha} \circ \mathbf{v}$, where

$$\mathbf{u} := Y^{-T} \mathbf{b}, \quad \mathbf{v} := Y^{-1} \mathbf{c}. \quad (3.43)$$

That is, \mathbf{u} and \mathbf{v} are respectively the solutions to the sparse linear equations $Y^T \mathbf{u} = \mathbf{b}$ and $Y \mathbf{v} = \mathbf{c}$, for which many solvers/algorithms are available. For example,

based on the diagonal dominance property of matrix Y , we can employ the power-iteration. Specifically, consider the decomposition $Y = D_y + E$, where D_y and E denote the diagonal and off-diagonal parts of Y . It is clear from the structure of $Y = L_\beta + \text{diag}(\mathbf{y})$ that only D_y depends on \mathbf{y} (hence the subscript y). Now consider \mathbf{u} , which satisfies $\mathbf{b} = Y^T \mathbf{u} = D_y \mathbf{u} + E^T \mathbf{u}$. Since D_y is invertible, we then have $\mathbf{u} = -D_y^{-1} E^T \mathbf{u} + D_y^{-1} \mathbf{b}$, which is a fixed point relation, where under Assumptions 3.2.1 and 3.2.2, the right side defines a contraction mapping with contraction coefficient $\rho(D_y^{-1} E^T) < 1$. Therefore, we can use the following iteration to compute \mathbf{u} :

$$\mathbf{u}_{k+1} = -D_y^{-1}(E^T \mathbf{u}_k - \mathbf{b}). \quad (3.44)$$

It should be noted that (3.44) is highly scalable since (i) E is sparse and can be read off from L (or W), whose storage takes only $O(|\mathcal{E}|)$ where $|\mathcal{E}|$ is the number of directed edges in the graph, and (ii) the computation also takes $O(|\mathcal{E}|)$ operations as it only involves a multiplication of \mathbf{u}_k with E^T and an element-wise scaling (after a subtraction by \mathbf{b}) by diagonal entries of D_y . Moreover, suppose (3.44) terminates in k_u iterations, which yields a convergence error proportional to $\rho^{k_u}(D_y^{-1} E^T)$, then the running time to compute \mathbf{u} is $O(k_u |\mathcal{E}|)$. Finally, \mathbf{v} can be computed in the same manner, i.e., $\mathbf{v}_{k+1} = -D_y^{-1}(E \mathbf{v}_k - \mathbf{c})$.

PGM belongs to the class of first order methods which only requires gradient evaluation (and projection step). Thus, it can be employed to deal with large networks. However, for networks that are not very large, other more efficient and sophisticated algorithms are available such as primal-dual IPMs [114]. Here we note

that each iteration of this method involves computing the Newton direction, which requires $O(N^3)$ operations to evaluate gradient ∇f and Hessian matrix \mathbf{H} , given respectively in (3.33) and (3.34). In practice, the method converges in a very few iterations (say, a few tens), which is often much less than required in PGM.

3.5 Supermodularity and Greedy Algorithms

In this section, we develop an alternative approach to problem (P) based on the greedy strategy where approximation bounds for the suboptimal solutions can be established. To this end, we first prove that $J_{\mathcal{K}}$ in (3.12) is monotone and supermodular in the set-variable \mathcal{K} . In fact, more is true, that is, *function f given by (3.32) is supermodular and monotone on Ω* . For this, we will give two different proofs as each has its own merit. As a result, problem (3.12) admits a $(1 - \frac{1}{e})$ approximation algorithm [108]. We then develop an improved version of this algorithm that is able to achieve better approximate solutions.

3.5.1 Supermodularity Results

We now establish supermodularity of the objective function f , and thus $J_{\mathcal{K}}$. Our first approach relies on the results in the previous subsection and the following known result.

Theorem 3.5.1. (Topkis' Characterization Theorem [121, 122]) *Let $\Omega = [\underline{\mathbf{x}}, \bar{\mathbf{x}}]$ be an interval in \mathbb{R}^N and $h : \mathbb{R}^N \rightarrow \mathbb{R}$ be twice continuously differentiable on (some open set containing) Ω . Then h is supermodular on Ω if and only if for all $\mathbf{x} \in \Omega$*

and all $i \neq j$, $\partial^2 h / \partial x_i \partial x_j \geq 0$. (There are no restrictions on $\partial^2 h / \partial x_i^2$.)

As a consequence, we have the following.

Theorem 3.5.2. (Supermodularity of objective functions) *Consider the function f in problem (P) and the set Ω defined in (P_Aprx). Then f is supermodular and monotone on Ω . Thus, the cost $J_{\mathcal{K}}$ is supermodular and monotone in \mathcal{K} .*

Proof. By Theorem 3.4.3, f in (3.32) is decreasing and its Hessian matrix \mathbf{H} is element-wise nonnegative on Ω . Supermodularity of f then follows from Theorem 3.5.1. Thus, $J_{\mathcal{K}}$ is also supermodular as it is the restriction of f on the vertexes of Ω . □

It should be pointed out that unlike in problem (P2), the function f in (P1) is not defined at $\mathbf{0} \in \Omega$ and thus is not twice continuously differentiable on any open set containing Ω . Therefore, the result above does not apply directly to problem (P1).

Next, we provide a second approach to proving the supermodularity result avoiding the technical problem above. This approach is based on the following two lemmas, the first of which is a *matrix supermodularity inequality* and the second is a *composition property*. These results not only provide us a deeper understanding of the influence process considered here but also are useful in proving the modularity of another related cost function used in the literature.

Lemma 3.5.3. (Matrix supermodularity inequality) *For any $\mathcal{S} \subset \mathcal{V}$, let $\Gamma_{\mathcal{S}} = \text{diag}(\boldsymbol{\alpha}_{\mathcal{S}})$. Then we have $(L_{\beta} + \Gamma_{\mathcal{S}})^{-1} \in \mathbb{R}_+^{N \times N}$ is nonincreasing and supermodular*

in \mathcal{S} , i.e., the following matrix inequalities hold for any $v, k \in \mathcal{V} \setminus \mathcal{S}$

$$(L_\beta + \Gamma_{\mathcal{S}})^{-1} - (L_\beta + \Gamma_{\mathcal{S} \cup \{v\}})^{-1} \geq (L_\beta + \Gamma_{\mathcal{S} \cup \{k\}})^{-1} - (L_\beta + \Gamma_{\mathcal{S} \cup \{k, v\}})^{-1} \geq \mathbf{0}. \quad (3.45)$$

This result also holds true if we replace L_β with L_0 .

Proof. The proof relies on the Woodbury matrix identity and results in M-matrix theory. See Appendix A.2.4 for details. \square

The result seems to suggest that opinion diffusion and influence spreading processes inherently possess monotonicity and supermodularity properties.

Remark 3.5.4. We do not exclude the case $\mathcal{S} = \emptyset$ since it can be seen that $(L_\beta + \Gamma_\emptyset)^{-1} = +\infty$ if $\beta = \mathbf{0}$.

Lemma 3.5.5. (Composition property) *Suppose $F : 2^{\mathcal{V}} \rightarrow \mathbb{R}^{N \times N}$ is decreasing and supermodular, $f : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}$ is increasing and convex. Then the composition $(f \circ F)$ is nonincreasing and supermodular.*⁸

Proof. This result is a straightforward extension of the standard case [121] in which $F : 2^{\mathcal{V}} \rightarrow \mathbb{R}$ and $f : \mathbb{R} \rightarrow \mathbb{R}$. Details are omitted for brevity. \square

Now using Lemmas 3.5.3 and 3.5.5 with $F(\mathcal{K}) = (L_\beta + \Gamma_{\mathcal{K}})^{-1}$, $f_1(X) = \mathbf{b}^T X |\boldsymbol{\xi}_0|$ and $f_2(X) = \mathbf{b}^T X \beta$, we again have that $J_{\mathcal{K}}^{(i)} = (f_i \circ F)(\mathcal{K})$ for $i = 1, 2$ are nonincreasing and supermodular.

Remark 3.5.6. The authors in [104] consider the problem of selecting a number of agents as leaders (in their context) in order to minimize the overall variance in an

⁸Here, \circ denotes the composition operator and should not be confused with the Hadamard product used in Section 3.4.

undirected unweighted network subject to stochastic disturbances. It can be verified that the cost function in that paper is equivalent to $\text{tr}((L + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1})$, which is equal to $(f \circ F)(\mathcal{K})$ with $F(\mathcal{K}) = (L + \Gamma_{\mathcal{K}})^{-1}$ and $f(X) = \text{tr}(X)$. Using the result above, we can immediately conclude the supermodularity property of this cost function; this was not established in [104].

3.5.2 Greedy Algorithms and Ratio Bounds

Having established supermodularity of the objective functions, we now introduce our greedy algorithms and show their ratio bounds. For convenience, $J_{\mathcal{S}}$ and $J(\mathcal{S})$ are used interchangeably. Our first algorithm, whose output is denoted by \mathcal{K}^G , is similar to the greedy algorithm in [108], which we described next.

Algorithm 3.1: *Greedy Adding \mathcal{K}^G*

Data: $W, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{b}$ and K

- 1 Init: $\mathcal{K}^G \leftarrow \emptyset$
- 2 **for** $i = 1 : K$ **do**
- 3 $k_i^* \leftarrow \arg \min \{J(\mathcal{K}^G \cup \{v\}), v \notin \mathcal{K}^G\}$
- 4 $\mathcal{K}^G \leftarrow \mathcal{K}^G \cup \{k_i^*\}$
- 5 Output: \mathcal{K}^G

Description of Algorithm 3.1: The idea is to start with an empty set \mathcal{K}^G (line 1) then greedily find one more node that most decreases the cost $J_{\mathcal{K}}$ to add to the set \mathcal{K}^G (lines 2-4). The algorithm is terminated after K sequential selections.

Remark 3.5.7. (Complexity of Algorithm 3.1) The number of function evaluations is $KN - \frac{K(K-1)}{2}$. In a naive way, without exploiting the structure of the cost function, each evaluation requires $O(N^3)$ operations (due to matrix inversion) and thus the total cost would be $O(KN^4)$. We can use the following tricks to alleviate this

computational burden.

- *Rank-1 updates:* At any iteration, let \mathcal{S} denote the current set \mathcal{K}^G and let $P := (L_\beta + \Gamma_{\mathcal{S}})^{-1}$. By the Woodbury identity (A.1), it can be verified that

$$(L_\beta + \Gamma_{\mathcal{S} \cup \{v\}})^{-1} = P - \frac{P_{(v)}P^{(v)}}{\alpha_v^{-1} + P_{vv}}. \quad (3.46)$$

Let $\Delta J(v, \mathcal{S}) := J(\mathcal{S}) - J(\mathcal{S} \cup \{v\})$. Then

$$\Delta J(v, \mathcal{S}) = \frac{\mathbf{b}^T P_{(v)} P^{(v)} \mathbf{c}}{\alpha_v^{-1} + P_{vv}}.$$

As a result, knowing P , it requires $O(N)$ operations to compute $\Delta J(v, \mathcal{S})$ and hence $O(N(N - |\mathcal{S}|))$ to find $v^* = \arg \max_{v \in \mathcal{V} \setminus \mathcal{S}} \Delta J(v, \mathcal{S})$. The matrix $(L_\beta + \Gamma_{\mathcal{S} \cup \{v^*\}})^{-1}$ is then obtained from P by a rank-1 update (3.46), which is $O(N^2)$. Note that the initial case $\mathcal{S} = \emptyset$ corresponds to $P = L_\beta^{-1}$, which takes $O(N^3)$ operations to compute. To sum up, using this scheme, the algorithm requires $O(KN^2 + N^3)$ operations, reduced from the naive way by a factor of $O(N)$. It also demands $O(N^2)$ of memory space (mainly to store the matrix inverse).

- *Power-iteration method:* For a very large network, it may be too expensive to reserve $O(N^2)$ memory for storing the inversed matrix $(L_\beta + \Gamma_{\mathcal{S}})^{-1}$. In this case, one can exploit the sparsity structure of L_β in connection with the power-iteration method to overcome the memory issue as shown in Remark 3.4.11. In particular, we can write $J(\mathcal{S}) = \mathbf{u}_{\mathcal{S}}^T \mathbf{c}$, where $\mathbf{u}_{\mathcal{S}} = (L_\beta + \Gamma_{\mathcal{S}})^{-T} \mathbf{b}$ can be computed using iteration (3.44). As before, let k_u denote the number of

iterations (on average) of running (3.44) (to achieve certain accuracy of \mathbf{u}).

Then the algorithm takes $O(KNk_u|\mathcal{E}|)$ operations and $O(|\mathcal{E}|)$ memory.

Note that the same greedy algorithm using rank-1 updates has been applied in [107] for the case of problem (3.8). Here, we use this algorithm for (3.12) (which is more general) and provide proofs of the supermodularity of $J_{\mathcal{K}}$ and the ratio bounding the error incurred, which were not included in [107].

Our result on the approximation guarantee of Algorithm 3.1 involves the notion of curvature of a submodular function (see, e.g., [109]). Let $Z(\mathcal{S})$ be nondecreasing submodular in \mathcal{S} . Then

$$\sigma := 1 - \min_{x \in \mathcal{P}} \frac{Z(\mathcal{P} \setminus \{x\}) - Z(\mathcal{P})}{Z(\emptyset) - Z(\{x\})}, \quad (3.47)$$

is called the total curvature of Z with respect to the set \mathcal{P} .

Theorem 3.5.8. ([109, Cor. 5.7]) *Let $Z(\mathcal{S})$ be a nondecreasing submodular function of \mathcal{S} such that $Z(\emptyset) = 0$. Let \mathcal{S}^G and \mathcal{S}^* denote the greedy solution and the optimal solution to the problem $\max\{Z(\mathcal{S}) : \mathcal{S} \subseteq \mathcal{P}, |\mathcal{S}| \leq K\}$. Then*

$$\frac{Z(\mathcal{S}^G)}{Z(\mathcal{S}^*)} \geq \frac{1}{\sigma} \left(1 - \left(1 - \frac{\sigma}{K}\right)^K\right) =: R_{\sigma, K} \quad (3.48)$$

where σ is the curvature of Z with respect to \mathcal{P} .

To use this result, we need to consider the case $\beta = \mathbf{0}$ separately since L_0 is singular and thus $J(\emptyset) = \infty$.

Theorem 3.5.9. (Properties of Alg. 3.1) *Let Assumptions 3.2.1 and 3.2.2 hold. Let \mathcal{K}^* denote an optimal solution to (3.12) and let \mathcal{K}^G be the output of Algorithm 3.1. Let $\mathcal{V}_\alpha = \{i \in \mathcal{V}, \alpha_i \neq 0\}$.*

(i) Let $v^* = \arg \min_{v \in \mathcal{V}_\alpha} J(\{v\})$. If $\beta = \mathbf{0}$, then

$$\frac{J(\{v^*\}) - J(\mathcal{K}^G)}{J(\{v^*\}) - J(\mathcal{K}^*)} \geq R_{\sigma, K-1} \quad (3.49)$$

where $\sigma = 1 - \min_{x \in \mathcal{V}_\alpha \setminus \{v^*\}} \frac{J(\mathcal{V}_\alpha \setminus \{x\}) - J(\mathcal{V}_\alpha)}{J(\{v^*\}) - J(\{v^*, x\})}$.

(ii) If $\beta \neq \mathbf{0}$, then

$$\frac{J(\emptyset) - J(\mathcal{K}^G)}{J(\emptyset) - J(\mathcal{K}^*)} \geq R_{\sigma, K} \quad (3.50)$$

where $\sigma = 1 - \min_{x \in \mathcal{V}_\alpha} \frac{J(\mathcal{V}_\alpha \setminus \{x\}) - J(\mathcal{V}_\alpha)}{J(\emptyset) - J(\{x\})}$.

Proof. (i). Define $Z(\mathcal{S}) := J(\{v^*\}) - J(\mathcal{S} \cup \{v^*\})$ for any $\mathcal{S} \subseteq \mathcal{V}_\alpha \setminus \{v^*\}$. Then it can be verified that Z is nondecreasing, submodular with curvature σ and $Z(\emptyset) = 0$. Thus, applying Theorem 3.5.8 and rearranging terms yield (3.49).

(ii). Similarly, (3.50) follows from Theorem 3.5.8 with σ being the curvature of $Z(\mathcal{S}) := J(\emptyset) - J(\mathcal{S})$ for any $\mathcal{S} \subseteq \mathcal{V}_\alpha$. \square

Note that $R_{\sigma, K} > \frac{1}{\sigma}(1 - e^{-\sigma}) > 1 - e^{-1}$ for any $\alpha \in (0, 1)$ and $K \geq 1$. Thus in general $R_{\sigma, K}$ is tighter than the constant bound $(1 - e^{-1})$ established in [108] (and also [63, 65, 67]).

Remark 3.5.10. (Bounds on $J(\mathcal{K}^*)$ by Alg. 3.1) Clearly, $J(\mathcal{K}^G)$ is an upper bound on $J(\mathcal{K}^*)$ and (3.49) or (3.50) provides a lower bound. We shall denote these bounds by J_{GU} and J_{GL} respectively; e.g., $J_{GL} = J(\emptyset) - \frac{J(\emptyset) - J_{GU}}{R_{\sigma, K}}$ for (3.50). Since $J(\mathcal{K}^*) \geq 0$, the bound J_{GL} is useful only if $J_{GL} \geq 0$, i.e., $J_{GU} \geq (1 - R_{\sigma, K})J(\emptyset)$ or $J_{GU} \geq (1 - R_{\sigma, K-1})J(\{v^*\})$.

In the following, we construct another algorithm (Algorithm 3.2 given and described below), which contains Algorithm 3.1 as a special case and is able to

practically improve accuracy. The idea is still to greedily select one “best” node at a time, but we additionally employ a particular swapping strategy: to repeatedly replace a selected node in \mathcal{K} by another node in $\mathcal{V} \setminus \mathcal{K}$ (or more precisely $\mathcal{V}_\alpha \setminus \mathcal{K}$) if the swapping most decreases the objective function. This strategy is in fact a special case of the Interchange Heuristic [108], which was also employed in [103] and [104] for problems related to sensor placement and leader selection. Our algorithm here differs from the aforementioned ones in that instead of swapping whenever an improvement of the cost function occurs, we carry out swapping in the direction of steepest descent coordinate, which helps avoid exponential number of exchanges. (As a side note, the supermodularity property and approximation bound for the greedy algorithm were not established in [103, 104]. Moreover, the convex analysis in these works is based on the symmetry of Laplacian matrices associated with undirected graphs.)

Algorithm 3.2: *Greedy Swapping* $\mathcal{K}_M^S := \text{GSwap}(\mathcal{K}_0^S, M)$

Data: $W, \alpha, \beta, \mathbf{b}, \mathcal{K}_0^S$, and M

- 1 **for** $m = 1 : M$ **or until** $\mathcal{K}_m^S = \mathcal{K}_{m+1}^S$ **do**
- 2 $\mathcal{S} \leftarrow \emptyset, \mathcal{T} = \{t_1, t_2, \dots\} \leftarrow \mathcal{K}_{m-1}^S$
- 3 **for** $i = 1 : K$ **do**
- 4 $\mathcal{T} \leftarrow \mathcal{T} \setminus \{t_i\}$
- 5 $t_i^* \leftarrow \arg \min_{v \notin \mathcal{S} \cup \mathcal{T}} J(\mathcal{S} \cup \{v\} \cup \mathcal{T})$
- 6 $\mathcal{S} \leftarrow \mathcal{S} \cup \{t_i^*\}$
- 7 $\mathcal{K}_m^S \leftarrow \mathcal{S}$
- 8 **Output:** \mathcal{K}_m^S

Description of Algorithm 3.2: The algorithm starts with an arbitrary set $\mathcal{K}_0^S \subseteq \mathcal{V}_\alpha$ (assuming $|\mathcal{K}_0^S| \in [0, K]$) and works in a cyclic manner for a predetermined number of cycles M or until $\mathcal{K}_{m^*}^S = \mathcal{K}_{m^*-1}^S$ for some m^* (line 1). In the m -th

cycle (lines 2-7), we revise the estimate \mathcal{K}_{m-1}^S from previous cycle by updating each entry one after the other; that is, for $i = 1, \dots, N$, we select $t_i^* \in \mathcal{R}_i := \mathcal{V} \setminus \{t_1^*, \dots, t_{i-1}^*, t_{i+1}, \dots, t_k\}$ that minimizes the cost $J(\mathcal{S} \cup \{v\} \cup \mathcal{T})$ (line 5), i.e.,

$$t_i^* = \arg \min_{v \in \mathcal{R}_i} J(\underbrace{\{t_1^*, \dots, t_{i-1}^*\}}_{\mathcal{S}}, v, \underbrace{t_{i+1}, \dots, t_k}_{\mathcal{T}}),$$

then add t_i^* to \mathcal{S} . We call this a greedy swapping step. Note that if $i > |\mathcal{K}_0^S|$, we allow $\{t_i\} = \emptyset$, in which case greedy swapping reduces to greedy adding (as in Algorithm 3.1). *In essence, this algorithm is based on the cyclic coordinate descent method (also known as the Gauss-Seidel method).*

Remark 3.5.11. (Entry search in Algorithm 3.2) In general, it is not computationally efficient to determine the optimal order in which the elements of the set \mathcal{K}^S are selected to be revised in each cycle (in order to reduce the objective cost to the extent possible). In this work, we use a cyclic selection scheme with the least possible complexity.

Remark 3.5.12. (Complexity of Alg. 3.2 with cyclic search) Each cycle (other than the first one) requires $(KN - K^2)$ function evaluations. That of the first cycle depends on $|\mathcal{K}_0^S|$, but is no more than $KN - \frac{K(K-1)}{2}$. Again, the naive approach takes $O(mKN^4)$ operations and $O(N^2)$ memory; but we can exploit the structure of the cost function to reduce these computational and memory costs, especially for large networks. Using the power-iteration method, we can avoid $O(N^2)$ memory requirement as shown in Remark 3.5.7. For not too large networks where storage is not an issue, we can employ the Woodbury matrix identity (A.1) for rank-2 updates (since swapping involves two nodes). Specifically, suppose we want to check for a

possible swap between $t \in \mathcal{T} \cup \mathcal{S} =: \mathcal{P}$ with some $v \in \mathcal{V} \setminus \mathcal{P}$. Let $P := (L_\beta + \Gamma_{\mathcal{P}})^{-1}$ and $E_{(tv)} := [\mathbf{e}_t, \mathbf{e}_v]$. Then it can be shown that

$$(L_\beta + \Gamma_{\mathcal{P} \setminus \{t\} \cup \{v\}})^{-1} = P - PE_{(tv)} \begin{bmatrix} P_{tt} - \alpha_t^{-1} & P_{tv} \\ P_{vt} & P_{vv} + \alpha_v^{-1} \end{bmatrix}^{-1} E_{(tv)}^T P. \quad (3.51)$$

Thus, $\Delta^2 J(-t, v, \mathcal{P}) := J(\mathcal{P}) - J(\mathcal{P} \cup \{v\} \setminus \{t\})$, the marginal gain of swapping t and v , can be computed as

$$[\mathbf{b}^T P_{(t)}, \mathbf{b}^T P_{(v)}] \begin{bmatrix} P_{tt} - \alpha_t^{-1} & P_{tv} \\ P_{vt} & P_{vv} + \alpha_v^{-1} \end{bmatrix}^{-1} \begin{bmatrix} P^{(t)} \mathbf{c} \\ P^{(v)} \mathbf{c} \end{bmatrix}$$

which takes $O(N)$ operations provided that P is known. Hence, finding $v^* = \arg \max_{v \in \mathcal{V} \setminus \mathcal{P}} \Delta^2 J(-t_i, v, \mathcal{S})$ requires $O(N(N - K))$ operations and if a swap is performed, the matrix $(L_\beta + \Gamma_{\mathcal{P} \setminus \{t_i\} \cup \{v^*\}})^{-1}$ is then obtained from P by a rank-2 update (3.51), which takes $O(N^2)$. (Note that the foregoing calculation resulting in the swapping selection above is also more computationally expensive than finding a possible greedy swap; which is also one of the reasons we opt for the greedy swapping strategy instead of the swapping method used in [103] and [104].) During each cycle, at most K swaps can be carried out, taking $O(KN^2)$ operations. For the initial cycle, if P is not supplied, then its computation costs at most $O(N^3)$. Thus, in general, for M cycles, Algorithm 3.2 takes $O(MKN^2 + N^3)$ operations. However, from our simulations, a good value of M is usually small (say 2-3) and does not scale as $O(N)$.

Theorem 3.5.13. (Properties of Alg. 3.2) *Let $\{\mathcal{K}_m^S\}_0^M$ denote the sequence of approximate solutions generated by Algorithm 3.2.*

(i) If $\mathcal{K}_0^S = \emptyset$, then $\mathcal{K}_1^S \equiv \mathcal{K}^G$, where \mathcal{K}^G denotes the output of Algorithm 3.1.

(ii) For any $m \geq 0$ and $\mathcal{K}_0^S \subseteq \mathcal{V}_\alpha$, $J(\mathcal{K}_{m+1}^S) \leq J(\mathcal{K}_m^S)$. In fact, let m^* denote the smallest index such that $\mathcal{K}_{m^*}^S = \mathcal{K}_{m^*+1}^S$, then

$$J(\mathcal{K}_m^S) < J(\mathcal{K}_{m+1}^S), \quad \forall m < m^*, \quad \text{and}$$

$$J(\mathcal{K}_m^S) = J(\mathcal{K}_{m+1}^S), \quad \forall m \geq m^*.$$

(iii) Let $v^* = \arg \min_{v \in \mathcal{V}_\alpha} J^{(1)}(\{v\})$. For any $\mathcal{K}_0^S \subseteq \mathcal{V}_\alpha$,

$$\frac{J^{(1)}(\{v^*\}) - J^{(1)}(\mathcal{K}_{m^*}^S)}{J^{(1)}(\{v^*\}) - J^{(1)}(\mathcal{K}^*)} > \frac{1}{2} \quad \text{and} \quad \frac{1 - J^{(2)}(\mathcal{K}_{m^*}^S)}{1 - J^{(2)}(\mathcal{K}^*)} > \frac{1}{2}.$$

Proof. (i) Consider $\mathcal{K}_0^S = \emptyset$ and the first cycle, i.e., $m = 1$. So, $\mathcal{T} = \emptyset$ and S is initialized as empty. As a result, line 5 becomes: $t_i^* = \arg \min_{v \in \mathcal{V} \setminus S} J(\mathcal{S} \cup \{v\})$, which together with line 6 is the greedy algorithm 3.1. Therefore, $\mathcal{K}_1^S \equiv \mathcal{K}^G$ as desired.

(ii) Consider the m -th cycle. It follows from the algorithm that $\mathcal{K}_{m-1}^S = \{t_1, t_2, \dots, t_K\}$ (line 3). By the greedy choice of t_i^* (line 5), it can be seen that $J(\mathcal{K}_{m-1}^S) = J(\{t_1, t_2, \dots, t_K\}) \leq J(\{t_1^*, t_2, \dots, t_K\}) \leq \dots \leq J(\{t_1^*, t_2^*, \dots, t_K^*\}) = J(\mathcal{K}_m^S)$. Thus $J(\mathcal{K}_{m-1}^S) = J(\mathcal{K}_m^S)$ if and only if all the inequalities in this relation become equalities, i.e., no further improvement on the objective can be made entry-wise. Hence, if $\mathcal{K}_{m^*}^S = \mathcal{K}_{m^*+1}^S$ for some m^* , then $J(\mathcal{K}_m^S) = J(\mathcal{K}_{m^*}^S), \forall m \geq m^*$. The existence of m^* clearly follows from the fact that the feasible set of \mathcal{K} is finite (which comes from finiteness of the network size).

(iii) For any submodular and nondecreasing function $Z(\mathcal{S})$, it follows from

[108, Thm. 5.1] that

$$\frac{Z(\mathcal{S}^*) - Z(\mathcal{S}^I)}{Z(\mathcal{S}^*) - Z(\emptyset)} \leq \frac{K - 1}{2K - 1} < \frac{1}{2}$$

where \mathcal{S}^* and \mathcal{S}^I denote the optimal solution and an interchange solution (i.e., no more possible local improvement) to the problem $\max\{Z(\mathcal{S}) : \mathcal{S} \subseteq \mathcal{P}, |\mathcal{S}| \leq K\}$. Applying this result to our case, where $Z(\mathcal{S}) := J^{(1)}(\{v^*\}) - J^{(1)}(\mathcal{S} \cup \{v^*\}), \forall \mathcal{S} \subseteq \mathcal{V}_\alpha \setminus \{v^*\}$ for problem (P1) or $Z(\mathcal{S}) := 1 - J^{(2)}(\mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{V}_\alpha$ for problem (P2), yields the desired results. Here, $\mathcal{K}_{m^*}^S$ is an interchange solution for each $\mathcal{K}_0^S \subseteq \mathcal{V}_\alpha$. \square

The ratio bound of $\frac{1}{2}$ in part (iii) is less than the constant $R_{\sigma, K}$ in Theorem 3.5.9 but holds for any initial set \mathcal{K}_0^S . Note also that first part of this proposition asserts that Algorithm 3.1 can be obtained from Algorithm 3.2 by letting $\mathcal{K}_0^S = \emptyset$ and $M = 1$. In this case, the performance of the latter algorithm is ensured to be no worst than the former. In fact, it is clear from part (ii) that better estimates are attained almost surely when $M > 1$.

Corollary 3.5.14. (Approximation accuracy of Alg. 3.2 with $\mathcal{K}_0^S = \emptyset$) *For any $m^* \geq 1$, $J(\mathcal{K}_{m^*}^S) \leq J(\mathcal{K}^G)$. Strict inequality holds if $m^* > 1$.*

Although we are not yet able to quantify this gain rigorously, our simulation results illustrate radical improvement compared to Algorithm 3.1, even with small values of M .

Remark 3.5.15. (On implementation of Alg. 3.2) The following are worth noting.

- *Starting point:* The algorithm works for an arbitrary choice of \mathcal{K}_0^S and thus can be useful in practice to improve upon a good starting set \mathcal{K}_0^S which may

be available from, e.g., the convex relaxation approach or Algorithm 3.1.

- *Local minimizer $\mathcal{K}_{m^*}^S$* : When it is found, there are practical techniques to possibly escape this local minimizer at the expense of more computation time and power; e.g., random swapping of multiple nodes in $\mathcal{K}_{m^*}^S$ with $\mathcal{V} \setminus \mathcal{K}_{m^*}^S$.
- *Termination*: We observed that even with a small M (say 2-3), the algorithm still finds a good approximation, especially from a good starting point. This may be attributable to the “diminishing returns” nature of the objective function resulting in significant improvements only in the first few cycles.

3.6 Numerical Examples

The simulations in this section were carried out in Matlab[®] R2015b on a PC with Intel[®] Core[™]i7 CPU@3.10 GHz and 12 GB of RAM.

3.6.1 Example 1: Small Network with One Leader

Consider the network depicted in Figure 3.1, where at every time step each agent updates its opinion by taking the average of its own opinion and those of its neighbors, i.e.,

$$w_{ij} = \frac{1}{|\mathcal{N}_i|}, \quad \forall j \in \mathcal{N}_i, \quad i \in \mathcal{V}.$$

This network is also studied in [62, 107].

Suppose there is an external leader with constant opinion $T = 0$ who wants to connect to a small number of agents so as to achieve fast consensus to its opinion;

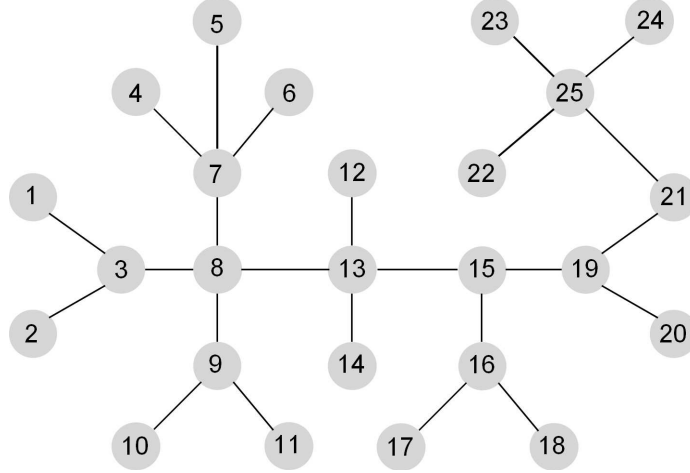


Figure 3.1: Network in example 1.

see Problem (P1). We revisit the problem of direct followers selection (with maximum level of trust) in [62], which corresponds to $\alpha=\infty$, $\beta=\mathbf{0}$, $\mathbf{x}_0=\mathbf{1}$ and $\mathbf{b}=\mathbf{1}/N$. Table 3.1 compares the simulation results of different approaches: (1) exhaustive search, which provides optimal solutions, (2) coordinate descent method [62], (3) Algorithms 3.1 and 3.2, and (4) the convex relaxation (P_Rlxd) solved by an Interior Point Method. In the last case, we approximate $\alpha=10^3\mathbf{1}$ to solve for \mathbf{y}_{P_Rlxd} , and then choose \mathcal{K}^{P_Rlxd} corresponding to the K largest elements of \mathbf{y}_{P_Rlxd} . We further apply the greedy swapping algorithm to \mathcal{K}^{P_Rlxd} ; see the last column, where $\mathcal{K}_1^{P_Rlxd} = \text{GSwap}(\mathcal{K}^{P_Rlxd}, 1)$ and $\mathcal{K}_2^{P_Rlxd} = \text{GSwap}(\mathcal{K}_1^{P_Rlxd}, 1)$.

As observed from Table 3.1, algorithm GSwap takes very few cycles to converge to optimal solutions except for the case $K = 5$, where it falls into a local minimizer. Usually, $M = 2$ is enough to obtain a good approximate solution, which is much improved from what generated by Algorithm 3.1 (and is exact in many cases).

Table 3.1: Comparison results for Network in example 1 (* denotes an optimal value). In the last column, $J_{\mathcal{K}_1^{P_Rlxd}}$ denotes $J_{\mathcal{K}_1^{P_Rlxd}}(J_{\mathcal{K}_2^{P_Rlxd}})$.

| | Exhaustive search | [62] | Alg.3.1 | Alg.3.2 ($\mathcal{K}_0^S = \emptyset$) | | | (P_Rlxd) in (3.28) | | |
|-----|---------------------------|---------------------|-------------------|---|-----------------------|---------------------------|--------------------|-----------------------------|-------------------------------|
| K | \mathcal{K}^* | $J_{\mathcal{K}^*}$ | $J_{\mathcal{K}}$ | $J_{\mathcal{K}^G}$ | $J_{\mathcal{K}_2^S}$ | $J_{\mathcal{K}_{m^*}^S}$ | m^* | $J_{\mathcal{K}^{P_Rlxd}}$ | $J_{\mathcal{K}_1^{P_Rlxd}}$ |
| 1 | 13 | 44.16 | 180.32 | * | * | * | 1 | 180.32 | *(*) |
| 2 | 8, 19 | 13.36 | 28.96 | 23.37 | * | * | 2 | 28.96 | 16.54 (*) |
| 3 | 8, 15, 25 | 6.94 | 10.47 | 9.29 | * | * | 2 | 10.47 | *(*) |
| 4 | 7, 8, 15 25 | 5.18 | 7.03 | 5.85 | * | * | 2 | 7.53 | 5.45 (*) |
| 5 | 3, 7, 9, 15 25 | 3.53 | 3.83 | 4.09 | 4.06 | 4.06 | 2 | 6.57 | 3.82 (3.82) |
| 6 | 3, 7, 9, 13 16, 25 | 2.22 | * | 3.13 | 2.54 | * | 3 | 5.61 | *(*) |
| 7 | 3, 7, 9, 13 16, 19, 25 | 1.36 | * | 2.17 | * | * | 2 | 2.17 | *(*) |

As for the implementation of the convex approach with regularization, there is no optimal rule for selecting μ , the sparsity penalizing coefficient, other than trial-and-error (see also Remark 3.4.6). Note that the computational cost per cycle of the coordinate descent method in [62], which involves the cost function's gradient and Hessian matrix evaluations at each coordinate, is roughly twice as much as that of Algorithm 3.2 (which requires only function evaluations). In addition, Algorithm 3.2 converges after 2-3 cycles with a guaranteed accuracy, while the coordinate descent method could take many more cycles for each trial of μ (with no provable bound on accuracy). Furthermore, although the Interior Point Method applied to (P_Rlxd) also employs the gradient and Hessian matrix, it converges within a few iterations (10-20 in this example).

We simulate the network responses for the case $K = 4$; see Figures 3.2-3.5, where the fastest convergence is when the leader repeatedly applies Algorithm 3.1 every $T_p = 5$ time steps.

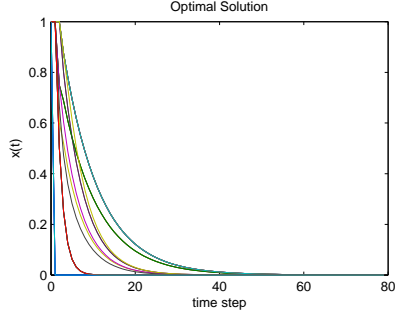


Figure 3.2: $\mathcal{K}^* = \{7, 8, 15, 25\}$

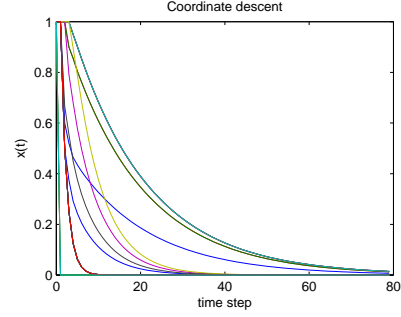


Figure 3.3: $\mathcal{K} = \{7, 13, 16, 25\}$

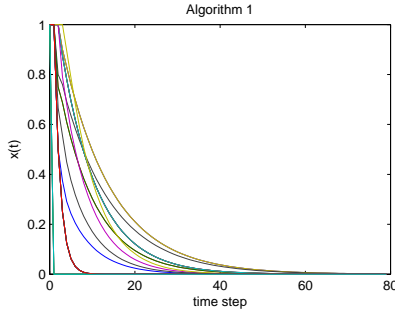


Figure 3.4: $\mathcal{K} = \{8, 13, 16, 25\}$

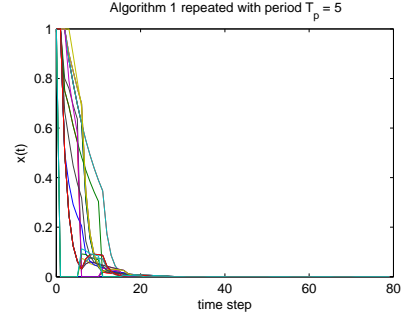


Figure 3.5: Alg. 3.1 every 5 time steps

3.6.2 Example 2: Medium-Size Network with Two Leaders

Consider a directed network based on the largest strongly connected component of the Wikipedia vote network⁹ studied in [123]. Thus, our network has $N = 1300$ nodes and 39456 edges. We generate the weight of each directed edge randomly in the interval $(0, 1)$. Suppose that leader Q has selected the set \mathcal{V}_β containing the first 50 nodes with the highest out-degrees and that $\beta_i = 10^6, \forall i \in \mathcal{V}_\beta$ (thus, they are in full support of Q). Suppose that leader T can connect to up to K nodes in \mathcal{V}_α that contains the first 1000 nodes that are not direct followers of Q (here “the first 1000 nodes is understood in terms of the numbering sequence of the nodes). We

⁹Data available at: <http://snap.stanford.edu/data/wiki-Vote.html>

also assume that $\alpha_i = 10, \forall i \in \mathcal{V}_\alpha$.

In this example, we consider problem (P2) for different values of $K \in [1, 200]$ using various schemes:

- (i) Algorithm 3.1: the greedy algorithm with output \mathcal{K}^G providing $J_{GU} = J(\mathcal{K}^G)$ and J_{GL} as upper and lower bounds on J^* (see Remark 3.5.10).
- (ii) (P_Rlxd)+IPM: the relaxed problem (P_Rlxd) solved by the Interior Point Method in OPTI toolbox [124],¹⁰ which gives \bar{f}_{P_Rlxd} and $f_{P_Rlxd}^*$ as upper and lower bounds.
- (iii) (P_Aprx)+IPM: the regularized problem (P_Aprx) solved by the Interior Point Method (with sparsity threshold set to 0.01). The output, denoted by \mathcal{K}^{P_Aprx} , yields a corresponding cost $f_{P_Aprx} =: J_{P_Aprx}$, an upper bound on J^* .
- (iv) GSwap($\mathcal{K}^{P_Aprx}, 1$): applying one cycle of the greedy swapping algorithm to \mathcal{K}^{P_Aprx} obtained from (iii).

The simulation results are shown in Figures 3.6 and 3.7. Here, the upper bounds by the greedy algorithm, $\text{GSwap}(\mathcal{K}^{P_Aprx}, 1)$ and (P_Rlxd)+IPM schemes are almost the same while the convex relaxation approach gives the best lower bounds, which help evaluate approximation errors. In particular, using these bounds, we are able to conclude that the the approximation ratio of greedy solutions \mathcal{K}^G (as well as that of $\text{GSwap}(\mathcal{K}^{P_Aprx}, 1)$ and (P_Rlxd)+IPM) satisfies

$$\frac{1 - J(\mathcal{K}^G)}{1 - J^*} \geq \frac{1 - J_{GU}}{1 - f_{P_Rlxd}^*}.$$

¹⁰Here, we let $\mathbf{y}^{(0)} = \mathbf{0}$ and stop the algorithm if $\frac{|f_i - f_{i-1}|}{|f_i|} \leq 10^{-6}$.

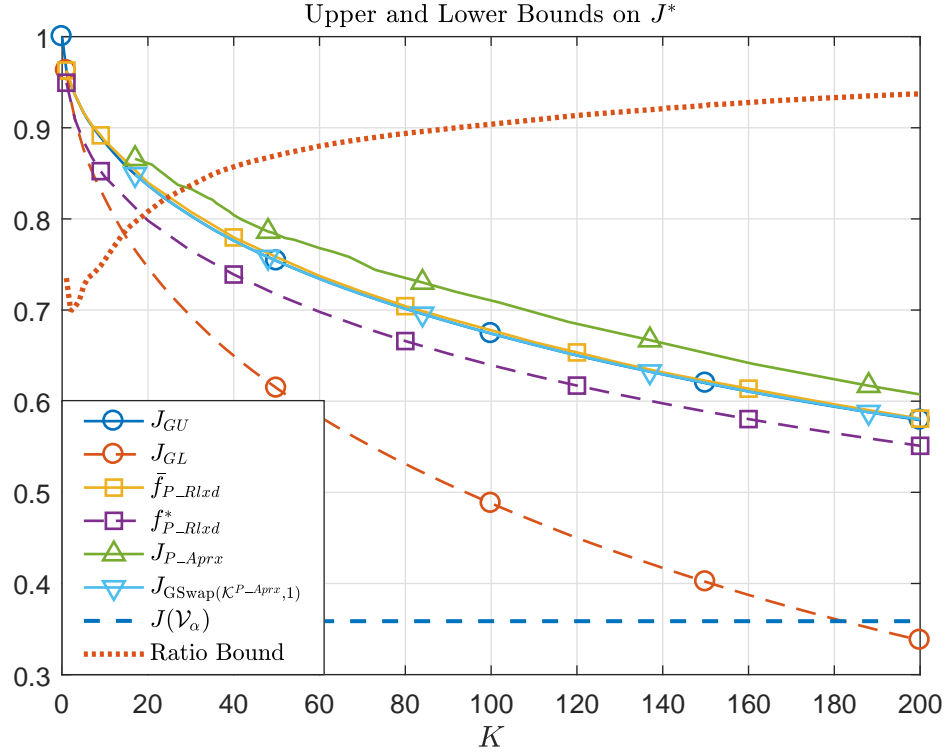


Figure 3.6: Upper bounds (solid lines) and lower bounds (dashed line) on J^* ; The global lower bound $J(\mathcal{V}_\alpha)$ holds for any K . The ratio bound $\frac{1-J_{GU}}{1-f_{P_Rlxd}^*}$ (shown by a dotted line) is at least 90% as $K \geq 90$.

The lower bound, depicted by a dotted line in Figure 3.6, is clearly much higher than $R_{\sigma,K}$ (here $\sigma = 0.99$) and the well-known theoretical ratio of $(1 - \frac{1}{e}) = 63.21\%$ for the greedy algorithm. For example, the ratio bound is at least 90% as $K \geq 90$.

Regarding running time, note that the greedy algorithm scales linearly with K , while the convex approach does not; see Figure 3.7. As μ increases, $|\mathcal{K}^{P-Aprx}|$ reduces, and thus so does the running time of $G\text{Swap}(\mathcal{K}^{P-Aprx}, 1)$.

3.7 Closing Discussion

This section provides further applications and results based on the analysis developed in the previous sections.

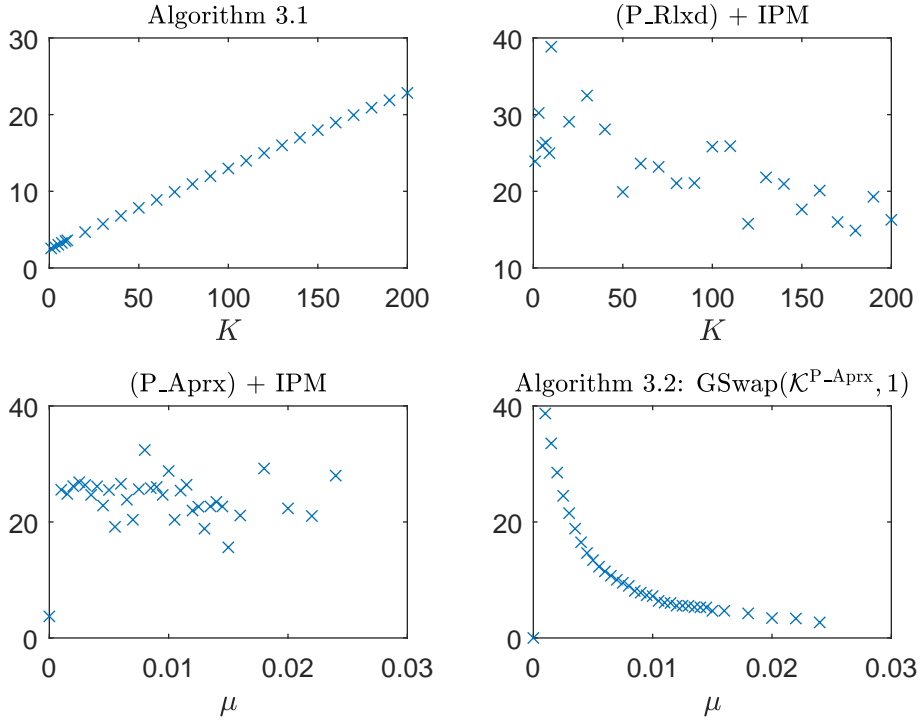


Figure 3.7: CPU run times (s) in 4 schemes. The Interior Point Method takes approximately 0.21 s per iteration.

3.7.1 Application to Friedkin's Model

Consider a Friedkin's model [3] in the presence of two leaders T and Q :

$$x_i(t+1) = \frac{\alpha_i T + \beta_i Q + \sigma_i x_i(0) + \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t)}{\alpha_i + \beta_i + \sigma_i + 1},$$

where, as before, α_i and β_i denote the weights that agent i puts on T and Q , respectively, and σ_i represents the stubbornness of agent i in keeping his initial opinion (or internal belief, see also [20, 63]). In matrix form,

$$\mathbf{x}(t+1) = (I + \text{diag}(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\sigma}))^{-1}(\boldsymbol{\alpha}T + \boldsymbol{\beta}Q + \boldsymbol{\sigma} \circ \mathbf{x}(0) + W\mathbf{x}(t)). \quad (3.52)$$

Again, assuming $Q = 1$ and $T = 0$, the equilibrium of the system is then given by

$$\mathbf{x}(\infty) = (L + \text{diag}(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\sigma}))^{-1}(\boldsymbol{\beta} + \boldsymbol{\sigma} \circ \mathbf{x}(0)).$$

Thus, we can define an associated influence optimization problem of “ T against Q and the stubborn” as follows

$$\min_{\mathcal{K} \subseteq \mathcal{V}} \{\mathbf{b}^T \mathbf{x}(\infty) = \mathbf{b}^T (L_{\beta+\sigma} + \text{diag}(\boldsymbol{\alpha}_{\mathcal{K}}))^{-1} \tilde{\mathbf{c}} : |\mathcal{K}| \leq K\}$$

where $L_{\beta+\sigma} := L + \text{diag}(\boldsymbol{\beta} + \boldsymbol{\sigma})$ and $\tilde{\mathbf{c}} := \boldsymbol{\beta} + \text{diag}(\boldsymbol{\sigma})\mathbf{x}(0)$. Clearly, this problem fits into the general one (P) described in (3.12) and thus can be treated efficiently by the methods developed in this chapter.

3.7.2 Further Convexity Results

The following theorem builds on the convexity analysis in Section 3.4.

Theorem 3.7.1. *Consider systems (i) $\mathbf{x}(t+1) = A_u \mathbf{x}(t)$ where $A_u = A + \text{diag}(\mathbf{u})$ is a nonnegative matrix, and (ii) $\mathbf{x}(t+1) = B_u \mathbf{x}(t)$ where $B_u = (A + \text{diag}(\mathbf{u}))^{-1}B$ and B is a nonnegative matrix and $A + \text{diag}(\mathbf{u})$ is a nonsingular M -matrix. For either system, if $\mathbf{x}(0) \geq 0$, then for $\forall t \geq 0$, $x_i(t)$ is convex in \mathbf{u} .*

Proof. [Sketch] For system (i), the conclusion follows from noting that $dA_u^{t+2} = \sum_{i+j+k=t} A_u^i \text{diag}(d\mathbf{u}) A_u^j \text{diag}(d\mathbf{u}) A_u^k$ and applying Lemma 3.4.2. For (ii), let $b_{ij}(\mathbf{u})$ denote the ij -th element of B_u . Similar to the convexity proof of f in (3.32), it can be shown that $b_{ij}(\mathbf{u})$ is positive, convex and decreasing in \mathbf{u} . Thus, $[B_u^t]_{ij}$, as a summation of products of $b_{ij}(\mathbf{u})$, is also positive, convex and decreasing in \mathbf{u} . \square

It can be verified that result for system (ii) in the statement of Theorem 3.7.1 can be applied to both models (3.2) and (3.9) as well as (3.52). It is also interesting to note that the result for system (i) in the statement of Theorem 3.7.1 is closely related to [125, Lem. 3], which states that for a continuous-time system $\dot{\mathbf{x}}(t) = (M + \text{diag}(\mathbf{u}))\mathbf{x}(t)$ with $\mathbf{x}(0) \geq 0$, the function $\mathbf{u} \mapsto x_i(t) = \mathbf{e}_i^T e^{(M + \text{diag}(\mathbf{u}))t} \mathbf{x}(0)$ is convex if M is a Metzler matrix.¹¹ Thus our result for system (i) can be viewed as the dual applied for discrete-time systems. However, we remark that the proof technique developed here is totally different. Moreover, in general, not every real matrix has a real logarithm, let alone uniqueness. The connection between these results and applications of Theorem 3.7.1 are left for future work.

3.7.3 Towards Relaxing Strong Connectivity Assumption

Consider the case where the network \mathcal{G} is fixed but not strongly connected. Without loss of generality, assume that \mathcal{G} is weakly connected (i.e., not disconnected). Then for each \mathcal{K} , we decompose $\mathcal{V} = \mathcal{V}_{\mathcal{K}} \cup \mathcal{V}_{\bar{\mathcal{K}}}$ where $\mathcal{V}_{\mathcal{K}}$ denotes the set of agents in \mathcal{G} that are reachable from \mathcal{K} . Clearly, $\{T\} \cup \mathcal{V}_{\mathcal{K}}$ forms a spanning tree rooted at T . Moreover, there are directed links from $\mathcal{V}_{\bar{\mathcal{K}}}$ to $\mathcal{V}_{\mathcal{K}}$ but not vice versa. As a result, the agents in $\mathcal{V}_{\bar{\mathcal{K}}}$ evolve independently with those in $\mathcal{V}_{\mathcal{K}}$ and reach an equilibrium. Thus, the opinion of each agent in $\mathcal{V}_{\mathcal{K}}$ also converges to a fixed point, which is a linear combination of T and the final opinions of those in $\mathcal{V}_{\bar{\mathcal{K}}}$. In this regard, $\mathcal{V}_{\bar{\mathcal{K}}}$ can be considered as other leaders to $\mathcal{V}_{\mathcal{K}}$ besides T , thus the analysis in this chapter can also be applied in this scenario.

¹¹i.e., off-diagonal entries are nonnegative.

Part II: Consensus Prediction by Observer

Chapter 4: Consensus Prediction in Minimum Time

Abstract: This chapter studies an observer that seeks to predict in minimal time the asymptotic agreement value of the agents in a network. The network is governed by the DeGroot opinion dynamics model. The observer can monitor the opinions of a group of agents, but might not have accurate knowledge of the underlying communication graph and the associated weight matrix. The work makes use of and builds on previous work on finite time consensus to address this prediction problem. In particular, for the case of a single observed agent, a tight lower bound on the monitoring time is determined below which the observer with limited knowledge about the network is not able to determine the consensus value regardless of the method used. This minimal prediction time can be achieved by employing the minimal polynomial associated with this observed agent. Next, for the general case of an observer with access to multiple agents, a similar bound is conjectured, and we develop algorithms toward achieving this bound through local observations and computations.

4.1 Introduction

In this chapter, we are concerned with the problem of predicting the consensus value of a network implementing a consensus protocol, where the agents exchange information according to the nearest neighbor weighted averaging scheme. This problem is related to the finite-time consensus problem that has been investigated in the literature (see, e.g., [71, 72]). Building on these contributions, we investigate the minimal observation time that enables an observer to determine the consensus value by monitoring a set of agents in the network.

This problem is useful in network monitoring and security. Moreover, the algorithms developed in this work can also be used to allow the agents to possibly reach consensus in a time that is shorter than the best known results in the literature. As an application, in Chapter 5, we will demonstrate the use of consensus prediction in developing distributed optimization algorithms that have many desired features.

The contributions of this work are as follows. First, we reveal an intrinsic relation between the consensus value and available observation data, based on which we (i) derive a fundamental limit on the monitoring time for the case of a single observed node, and (ii) provide a conjecture and analysis for the case of multiple observed nodes. Next, we develop algorithms toward achieving the conjectured bounds through local observations and computations.

The rest of the chapter is organized as follows. In Section 4.2 we describe the problem formulation and provide some background on the finite-time consensus protocol developed in [71]. In Section 4.3, we provide the main results on shortest

time prediction of consensus using the notion of a node’s minimal polynomial as introduced in [71]. Then in Section 4.4 we develop algorithms for computing minimal polynomials in a distributed manner and in (sub)optimal time. We provide numerical examples and discuss problems for future work in Sections 4.5 and 4.7, respectively.

4.2 Problem Statement and Previous Results

4.2.1 Problem Description

Consider a network consisting N agents denoted by $\mathcal{V} = \{1, 2, \dots, N\}$ with the underlying communication characterized by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $x_i(t)$ denote the state or opinion of node i at time $t \geq 0$; $x_i(0)$ represents the initial opinion. At any time t , each agent observes opinions of its neighbors and updates its opinion following the DeGroot model (1.4) as described in Chapter 1, namely:

$$x_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t), \quad \forall t \geq 0, \quad \forall i \in \mathcal{V}, \quad (4.1)$$

where, recall that, \mathcal{N}_i denotes the set of agent i ’s neighbors (including itself) and $W = [w_{ij}]$ the weight matrix. In this chapter, the following is a blanket assumption, which is the combination of Assumptions 1.5.2 and 1.5.3 and presented here for convenience.

Assumption 4.2.1. (Network Connectivity and Weight Matrix) *The graph \mathcal{G} is fixed and strongly connected. The weight matrix W is fixed, row-stochastic and satisfies $w_{ij} > 0$ for $(i, j) \in \mathcal{E}, i \neq j$, and $w_{ij} = 0$ otherwise. Moreover, W has at*

least one positive diagonal element.

Under this assumption, the network asymptotically achieves consensus:

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{1}\boldsymbol{\pi}^T \mathbf{x}(0), \quad (4.2)$$

where we recall that $\boldsymbol{\pi} \in \mathbb{R}^N$ is the normalized left Perron eigenvector of W , i.e., $\boldsymbol{\pi}^T W = \boldsymbol{\pi}^T$ and $\mathbf{1}^T \boldsymbol{\pi} = 1$; see Section 1.5.2 for details. Denote by x^* the consensus value, i.e., $x^* = \boldsymbol{\pi}^T \mathbf{x}(0)$. Our problems of interest are as follows:

Suppose that there is an observer that might not know W but can monitor the states of m nodes in the network starting from initial time $t = 0$. First, for any initial states $\mathbf{x}(0)$, how can the observer predict the consensus value x^ in minimum time? Second, which nodes should be observed to minimize the number of time-steps needed when more information on the network is available?*

Let $\mathcal{O} \subset \mathcal{V}$ denote the set of m nodes selected by the observer. By the *observation* at time t we mean the vector $\mathbf{x}_{\mathcal{O}}(t) \in \mathbb{R}^m$ that includes the states of observed nodes at time t . The number of consecutive observations (starting from $t = 0$) that allows the observer to determine x^* is called the *observation time*.

We find it convenient to introduce the following information model. Define $\Theta(t)$ as the “accumulated information” about the network that the observer possesses at time t . (Note that $\Theta(t)$ is an equivalence class.) Let $\Theta(-1)$ denote the initial knowledge and assume that the information dynamics satisfies $\Theta(t + 1) = \Theta(t) \cup \{\mathbf{x}_{\mathcal{O}}(t + 1)\}$, implying that the observer accumulates information. As a result, at any time $t \geq 0$, the observer knows $\mathbf{x}_{\mathcal{O}}(s), \forall s \in [0, t]$.

4.2.2 Previous Results on Consensus in Finite Time

We now recall the method in [71] that enables the agents to exactly calculate the consensus value after running the iteration (4.1) for only a finitely many steps. The method is based on the concept of an individual node's minimal polynomial, which is given below.

First, recall that for any square matrix $A \in \mathbb{R}^{n \times n}$, its associated minimal polynomial q_A is the monic polynomial of least degree for which $q_A(A) = \mathbf{0}_{n \times n}$.

Definition 4.2.2. (*Minimal polynomial of a node [71]*) Given weight matrix W , the *minimal polynomial of node i* , denoted by q_i , is the monic polynomial of least degree for which $\mathbf{e}_i^T q_i(W) = \mathbf{0}_n^T$, where \mathbf{e}_i is the i^{th} standard unit basis vector.

The existence of q_i follows from the fact that q_W satisfies the condition $\mathbf{e}_i^T q_W(W) = \mathbf{0}_n^T$. Moreover, q_i is easily seen to be unique by virtue of being a monic polynomial of least degree satisfying this condition. Note also that the value $\deg(q_i)$'s may not be the same for different $i \in \mathcal{V}$. However, it always holds that $\deg(q_i) \leq \deg(q_W)$, $i \in \mathcal{V}$. Furthermore, important properties of q_i are given below; see [71] for a proof.

Lemma 4.2.3. (Properties of minimal polynomial of a node) *For each $i \in \mathcal{V}$, q_i divides q_W . Moreover, if μ is a simple eigenvalue of W whose associated eigenvector has all nonzero elements, then μ is a simple root of q_i .*

As a consequence, when Assumption 4.2.1 holds, all the roots of q_i are strictly inside the unit circle except only one at 1. Denoting

$$D_i := \deg(q_i) - 1,$$

the minimal polynomial q_i can be expressed as

$$q_i(\xi) = (\xi - 1) \sum_{0 \leq l \leq D_i} a_l^{(i)} \xi^l, \quad (4.3)$$

where $\mathbf{a}^{(i)} = [a_0^{(i)}, a_1^{(i)}, \dots, a_{D_i}^{(i)}]^T$ satisfies

$$a_{D_i}^{(i)} = 1, \quad \sum_{0 \leq l \leq D_i} a_l^{(i)} \neq 0. \quad (4.4)$$

This decomposition of q_i will be useful in determining the consensus value at each node in finitely many iterations as we briefly describe next; for a full development with all steps given in detail, the reader is referred to [71]. Recall from Definition 4.2.2 that $\mathbf{e}_i^T q_i(W) = \mathbf{0}_n^T$. Thus, for $t \geq 0$,

$$0 = \mathbf{e}_i^T q_i(W) \mathbf{x}(t) \stackrel{(4.3)}{=} \sum_{0 \leq l \leq D_i+1} (a_{l-1}^{(i)} - a_l^{(i)}) \mathbf{e}_i^T W^l \mathbf{x}(t),$$

where $a_{-1}^{(i)} = a_{D_i+1}^{(i)} = 0$ for the convenience of notation. Note that $\mathbf{e}_i^T W^l \mathbf{x}(t) = x_i(t+l)$. Thus, we have

$$\sum_{0 \leq l \leq D_i+1} c_l^{(i)} x_i(t+l) = 0, \quad \forall t \geq 0,$$

where $c_l^{(i)} := a_{l-1}^{(i)} - a_l^{(i)}$. Denote by $X_i(z)$ the z -transform of the signal x_i . Applying the unilateral z -transform to the equation above and invoking the time-shifting property yields

$$q_i(z) X_i(z) = \sum_{1 \leq l \leq D_i+1} c_l^{(i)} \sum_{0 \leq j \leq l-1} x_i(j) z^{l-j}. \quad (4.5)$$

By the Final Value Theorem (see, e.g., [126]), we then have

$$\lim_{t \rightarrow \infty} x_i(t) = \lim_{z \rightarrow 1} (z-1) X_i(z) \stackrel{(4.3)-(4.5)}{=} \frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(l)}{\sum_{l=0}^{D_i} a_l^{(i)}}. \quad (4.6)$$

Note that $\{x_i(0), \dots, x_i(D_i)\}$ are consecutive state values of agent i . Thus (4.6) implies that agent i can find $\lim_{t \rightarrow \infty} x_i(t)$ after D_i iterations of (4.1), provided that $\mathbf{a}^{(i)}$ is known. By (4.2), this limit is the consensus value $x^* = \boldsymbol{\pi}^T \mathbf{x}(0)$.

Remark 4.2.4. The method presented above can be viewed as each agent being an observer with its own information model: $\Theta_i(t+1) = \Theta_i(t) \cup \{x_i(t)\}$ and $\Theta_i(-1) = \emptyset$. We remark, however, that in general, even in a distributed setting, more local information is available to each agent than just its own state, e.g., $\Theta_i(t+1) = \Theta_i(t) \cup \{\mathbf{x}_{\mathcal{N}_i}(t)\}$ (recalling that \mathcal{N}_i denotes the set of direct neighbors of agent i) and $\Theta_i(-1)$ might not be empty.

Remark 4.2.5. Our setting of having just one observer is more general in the sense that the scenario above can be seen as a special case with appropriate choices of observed nodes \mathcal{O} and information model $\Theta(t)$.

Remark 4.2.6. It is obvious from (4.6) that agent i (or the observer that monitors agent i) determine the consensus value x^* as a linear combination of $D_i + 1$ consecutive observations of agent i 's state. This is merely a consequence of the use of the minimal polynomial q_i , which by no means assures the optimality of $D_i + 1$ a priori. Hence, the following the question is also of interest:

Among all possible methods that the observer may use to find x^ , which is associated with the least observation time?*

4.3 Shortest Time Prediction of Consensus and Local Computation of Minimal Polynomials

This section deals with the question posed in the foregoing remark. It turns out that if $\mathcal{O} = \{i\}$, then the number $D_i + 1$ of observations is *optimal* in determining x^* for any initial value of the network and for all possible methods. This optimal value can be achieved by using minimal polynomial q_i as in (4.6). We show this in detail, then present an optimality conjecture for the case of having multiple observed nodes and discuss an idea to achieve this minimum observation time through the computation of minimal polynomials.

We first uncover an intrinsic relation between the consensus value and observation data: if x^* can be computed at some time $r \geq 0$, then x^* is a *linear combination of available observation data* with associated coefficients depending on W .

Theorem 4.3.1. *If $r \in \mathbb{Z}_+$ and $g : \mathbb{R}^{m(r+1)} \rightarrow \mathbb{R}$ are such that for any $\mathbf{x}(0) \in \mathbb{R}^N$*

$$x^* = g(\mathbf{x}_{\mathcal{O}}(r), \mathbf{x}_{\mathcal{O}}(r-1), \dots, \mathbf{x}_{\mathcal{O}}(0)), \quad (4.7)$$

then $\exists \beta_0, \beta_2, \dots, \beta_r \in \mathbb{R}^m$ such that $x^ = \sum_{i=0}^r \beta_i^T \mathbf{x}_{\mathcal{O}}(i)$.*

To prove this result, we make use of the linearity of the dynamic system (4.1) in conjunction with the following lemma, whose proof is an application of the Hahn-Banach theorem.

Lemma 4.3.2. *[127, p. 188] Let f_0, f_1, \dots, f_n be linear functionals on a vector space V and suppose that $f_0(v) = 0$ for every $v \in V$ satisfying $f_i(v) = 0$ for $i = 1, 2, \dots, n$. Then there are constants $\beta_1, \beta_2, \dots, \beta_n$ such that $f_0 = \sum_{i=1}^n \beta_i f_i$.*

Proof of Theorem 4.3.1: Let r and g satisfy (4.7). Define the following functions $f_0, f_{i,t} : \mathbb{R}^N \rightarrow \mathbb{R}$ for any $t \geq 0$ and $i \in \mathcal{O}$ such that for $\forall \mathbf{v} \in \mathbb{R}^N$

$$f_{i,t}(\mathbf{v}) := \mathbf{e}_i^T W^t \mathbf{v}, \quad f_0(\mathbf{v}) := \lim_{t \rightarrow \infty} \mathbf{e}_1^T W^t \mathbf{v}. \quad (4.8)$$

That is, if $\mathbf{x}(0) = \mathbf{v}$, then $f_{i,t}(\mathbf{v}) = x_i(t)$ and $f_0(\mathbf{v}) = \lim_{t \rightarrow \infty} x_1(t) = x^*$ since the network reaches consensus. Clearly, f_0 and $f_{i,t}$ are linear functions on \mathbb{R}^N . Next, define

$$\Omega = \{\mathbf{v} \in \mathbb{R}^N \mid f_{i,t}(\mathbf{v}) = 0, 0 \leq t \leq r, i \in \mathcal{O}\}.$$

It can be verified that Ω is a subspace on which $\mathbf{x}_{\mathcal{O}}(t) = \mathbf{0}$ for $0 \leq t \leq r$. We now consider f_0 on Ω . It follows from (4.7) that for any $\mathbf{v} \in \Omega$ and $\gamma \in \mathbb{R}$

$$f_0(\mathbf{v}) = g(\mathbf{0}, \mathbf{0}, \dots, \mathbf{0}) = f_0(\gamma \mathbf{v}) = \gamma f_0(\mathbf{v}), \quad (4.9)$$

where the second equality holds since $\gamma \mathbf{v} \in \Omega$, and the last equality by linearity of f_0 . As a result, we have $f_0(\mathbf{v}) = 0$ for any $\mathbf{v} \in \Omega$. Therefore, by using Lemma 4.3.2, we have

$$f_0 = \sum_{0 \leq t \leq r, i \in \mathcal{O}} \beta_{i,t} f_{i,t}$$

for some constants $\beta_{i,t}$. This concludes the proof. \square

Next, we will employ Theorem 4.3.1 to assess the optimality of using minimal polynomials in consensus prediction.

4.3.1 Optimality of $(D_i + 1)$

Our main result for the case of single observed node is as follows:

Theorem 4.3.3. *Suppose $\mathcal{O} = \{i\} \subset \mathcal{V}$ and $\Theta(t) = \Theta(t-1) \cup \{x_i(t)\}, \forall t \geq 0$, where $\Theta(-1)$ may contain any information related to W . Then the observation time is always bounded below by $D_i + 1$, regardless of the method used. Furthermore, this bound can be achieved if $q_i \in \Theta(-1)$.*

Proof: Suppose $\mathcal{O} = \{1\}$. We prove by contradiction, i.e., suppose there exist a positive integer $r < D_1$ and a mapping $g : \mathbb{R}^r \rightarrow \mathbb{R}$ such that for any $\mathbf{x}(0) \in \mathbb{R}^N$,

$$x^* = g(x_1(r), x_1(r-1), \dots, x_1(0)). \quad (4.10)$$

Here, g depends on $\Theta(-1)$. By Theorem 4.3.1, we conclude that there exist $\beta_0, \beta_2, \dots, \beta_r$ such that $x^* = \sum_{i=0}^r \beta_i x_1(i)$. Without loss of generality, assume that $\beta_r \neq 0$ (otherwise, we consider β_{r-1} and so on). Define $f_t(\mathbf{v}) := \mathbf{e}_1^T W^t \mathbf{v}$ and $f_0(\mathbf{v}) := \lim_{t \rightarrow \infty} \mathbf{e}_1^T W^t \mathbf{v}$ for any $\mathbf{v} \in \mathbb{R}^N$. Then

$$f_0 = \sum_{0 \leq i \leq r} \beta_i f_i. \quad (4.11)$$

Note that for $\forall t \in \mathbb{Z}_+$ and $\forall \mathbf{v} \in \mathbb{R}^N$, we have

$$f_0(W\mathbf{v}) = f_0(\mathbf{v}), \quad f_t(W\mathbf{v}) = f_{t+1}(\mathbf{v}),$$

which in view of (4.11) then implies that

$$\begin{aligned} \sum_{0 \leq i \leq r} \beta_i f_{i+1} &= \sum_{0 \leq i \leq r} \beta_i f_i \\ \iff \mathbf{e}_1^T W^{r+1} + \sum_{1 \leq i \leq r} \frac{\beta_{i-1} - \beta_i}{\beta_r} \mathbf{e}_1^T W^i - \frac{\beta_0}{\beta_r} \mathbf{e}_1^T &= \mathbf{0}^T. \end{aligned}$$

As a result, the polynomial \tilde{q}_1 given by $\tilde{q}_1(\xi) = \xi^{r+1} + \sum_{i=1}^r \beta_r^{-1}(\beta_{i-1} - \beta_i)\xi^i - \beta_r^{-1}\beta_0$ satisfies

$$\mathbf{e}_1^T \tilde{q}_1(W) = 0, \quad \text{with} \quad \deg(\tilde{q}_1) = r + 1 < D_1 + 1,$$

which, however, contradicts the fact that the minimal polynomial q_1 of node 1 is of degree $D_1 + 1$. This concludes that the observation time is always bounded below by $D_1 + 1$.

It remains to show that this bound is achieved if $q_1 \in \Theta(-1)$. This is obvious in view of (4.6). \square

Remark 4.3.4. As we have shown that the shortest time $D_i + 1$ can be achieved if $q_i \in \Theta(D_i)$, in which case the coefficients β_j in Theorem 4.3.1 can be determined from q_i as $\beta_j = a_j^{(i)} / \sum_{l=0}^{D_i} a_l^{(i)}$. In the case where $\Theta(-1) = \emptyset$, then D_i and $a_j^{(i)}$ for $j = 0, 1, \dots, (D_i - 1)$ become $D_i + 1$ unknowns characterizing q_i , and therefore, the observer would need $D_i + 1$ additional observation data in order to be able to determine these unknowns.

In the following, we consider the case $m \geq 2$ and are interested in quantifying the minimal observation time conditionally on the initial information $\Theta(-1)$ in terms of q_i . Although we have not yet been able to determine the minimum time, we conjecture the following:

Conjecture 4.3.5. *Suppose the observer can monitor the states of a set \mathcal{O} of m nodes and $\Theta(t) = \Theta(t - 1) \cup \{\mathbf{x}_{\mathcal{O}}(t)\}, \forall t \geq 0$. Let T_{inf} denote the least observation time and let $D_{\text{min}} = \min_{i \in \mathcal{O}} D_i$.*

(i) *If $\Theta(-1) = \{q_i, \forall i \in \mathcal{O}\}$, then $T_{\text{inf}} = D_{\text{min}} + 1$.*

(ii) *If $\Theta(-1) = \emptyset$, then $T_{\text{inf}} \geq D_{\text{min}} + 2 + \lceil \frac{D_{\text{min}}}{m} \rceil$.¹*

¹For any $x \in \mathbb{R}$, $\lceil x \rceil$ denotes the least integer greater than or equal to x .

Remark 4.3.6. Case (i) can be reasoned as follows. Without loss of generality, let $\mathcal{O} = \{1, \dots, m\}$ and $D_1 = D_{\min}$. If $T_{\inf} \leq D_1$, i.e., x^* can be found by the time $t = D_1 - 1$. By Theorem 4.3.1, x^* is a linear combination of

$$\{x_i(k), 1 \leq i \leq m, 0 \leq k \leq D_1 - 1\}.$$

However, it follows from (4.6) that, at time $t = D_1 - 1$, we have m linear equations:

$$x^* = \frac{\sum_{k=0}^{D_i} a_k^{(i)} x_i(k)}{\sum_{k=0}^{D_i} a_k^{(i)}}, \quad \forall i \in \mathcal{O}$$

with at least $m + 1$ unknowns including x^* and $\mathbf{x}_{\mathcal{O}}(D_1)$. Thus, in general, x^* is not computable up to time $t = D_1 - 1$.

Remark 4.3.7. Case (ii) of the conjecture is based on our development in the next section where the idea is to demonstrate that the lower bound on T_{\inf} can be achieved if q_k with $k = \arg \min_{i \in \mathcal{O}} D_i$ can be computed from observation data up to that time and if “ideal conditions” (which will be made clear later) are assumed.

Remark 4.3.8. With a different assumption on $\Theta(-1)$, it is possible that $T_{\inf} < D_{\min}$. For example, if $\mathcal{O} = \mathcal{V}$ and $\{\boldsymbol{\pi}\} \subseteq \Theta(-1)$, then $x^* = \boldsymbol{\pi}^T \mathbf{x}_{\mathcal{O}}(0)$, i.e., $T_{\inf} = 1$.

4.3.2 Local Computation of q_i

The minimal polynomial q_i can be computed locally by agent i in many ways.

First, let $\mathbf{c}^{(i)} = [c_0^{(i)}, c_1^{(i)}, \dots, c_{D_i}^{(i)}, 1]^T \in \mathbb{R}^{D_i+2}$ denote the vector of coefficients of q_i . Then it follows from the definition that

$$\mathbf{0} = \mathbf{e}_i^T q_i(W) = \sum_{k=1}^{D_i+1} c_k^{(i)} \mathbf{e}_i^T W^k = (\mathbf{c}^{(i)})^T O_{D_i+2}^{(i)}, \quad (4.12)$$

$$\text{with } O_{D_i+2}^{(i)} = [\mathbf{e}_i \ W^T \mathbf{e}_i \ \dots \ (W^{D_i+1})^T \mathbf{e}_i]^T.$$

Observe that $O_{D_i+2}^{(i)}$ has the form of the observability matrix for the pair (W, \mathbf{e}_i^T) . Therefore, the observer might be able to compute $\mathbf{c}^{(i)}$ by constructing $O_k^{(i)}$ and increasing k until $O_k^{(i)}$ loses rank. This particular value of k is equal to $D_i + 2$, i.e., at time $t = D_i + 1$. Moreover, it should be noted that the construction of $O_k^{(i)}$ need not require the knowledge of the entire network. Specifically, let $\mathcal{N}_k^{(i)}$ denote the set of agents connected to node i through a path of length at most k . Thus $O_k^{(i)}$ can be determined using a submatrix of W with column and row in $\mathcal{N}_k^{(i)}$. This requires appropriate a dynamic information model: $\Theta(t+1) = \Theta(t) \cup \{x_i(t), \mathbf{e}_i^T W^t\}$.

A distributed algorithm for computing q_i was also proposed in [71], where network performs N runs of (4.1) with different initial conditions $\mathbf{x}^{(1)}(0), \mathbf{x}^{(2)}(0), \dots, \mathbf{x}^{(N)}(0)$ assumed to be linearly independent, each for $N + 1$ time steps. During each run, every node stores its own values. After N runs, every node is able to matrix

$$X_{i,t} = \begin{bmatrix} x_i^{(1)}(0) & x_i^{(1)}(1) & \cdots & x_i^{(1)}(t+1) \\ x_i^{(2)}(0) & x_i^{(2)}(1) & \cdots & x_i^{(2)}(t+1) \\ \vdots & \vdots & \ddots & \vdots \\ x_i^{(N)}(0) & x_i^{(N)}(1) & \cdots & x_i^{(N)}(t+1) \end{bmatrix}$$

where $x_i^{(j)}(t)$ is the value of node i at time t in the j -th run. Then D_i is the smallest positive integer for which X_{i,D_i} is not full column rank and the coefficient vector of q_i , denoted by $\mathbf{c}^{(i)}$, can be found from $X_{i,D_i} \mathbf{c}^{(i)} = 0$; see [71] for details.

In [72] the authors presented another algorithm for computing q_i which also uses solely agent i 's state values but requires a fewer number of time steps. In particular, let the network run (4.1) for at most $2N + 1$ time steps, starting from almost arbitrary initial state $\mathbf{x}(0)$ (except for a set of Lebesgue measure zero in \mathbb{R}^N).

Each node i constructs its Hankel matrix $H_{i,k}$ defined through setting $z_i(k+1) = x_i(k+1) - x_i(k)$ and

$$H_{i,k} := \begin{bmatrix} z_i(1) & z_i(2) & \cdots & z_i(k+1) \\ z_i(2) & z_i(3) & \cdots & z_i(k+2) \\ \vdots & \vdots & \ddots & \vdots \\ z_i(k+1) & z_i(k+2) & \cdots & z_i(2k+1) \end{bmatrix} \quad (4.13)$$

and finds the first rank-defective matrix $H_{i,k}$ as k increases, namely H_{i,D_i} . Then $\mathbf{a}^{(i)}$ is computed from $H_{i,D_i} \mathbf{a}^{(i)} = \mathbf{0}$. Although it is hard to characterize the set of initial states $\mathbf{x}(0)$ (of measure zero) for which this computation scheme fails to provide $\mathbf{a}^{(i)}$, practical techniques to alleviate the problem are available; see, e.g., [128]. More importantly, this approach in general provides a minimum time of $2(D_i + 1)$ for consensus prediction in the scenario that $\Theta(t+1) = \Theta(t) \cup \{x_i(t)\}$ with $\Theta(-1) = \emptyset$. See [72] for further details.

We remark that the idea of using Hankel matrix $H_{i,k}$ to compute $\mathbf{a}^{(i)}$ in fact has its roots in the realization theory [73–75]. Here, finding q_i can be regarded as a network identification problem. Aimed with this view, in the next section, we build our algorithms on the previous approach by employing block-Hankel matrices in order to reduce the observation time needed to compute q_i .

4.4 Toward Minimizing Observation Time

In this section, we present partial solutions to the problem described in Section 4.2.1. Clearly, when $m = 1$, it follows from Section 4.3 (see Theorem 4.3.3) that

the solution is given by $\mathcal{O} = \arg \min_{i \in \mathcal{V}} D_i$. Thus in the following we consider $m \geq 2$. Our main idea is to make use of available information to construct block-Hankel matrices instead of (4.13). We will consider two cases: (1) when the minimal polynomials of observed nodes are identical, i.e., $q_i = q_j, \forall i, j \in \mathcal{O}$, and (2) when they are nonidentical, i.e., $\exists i, j \in \mathcal{O}, q_i \neq q_j$. In any case, we do not assume that $q_i \in \Theta(-1), \forall i \in \mathcal{O}$.

4.4.1 Observed Nodes with Identical Minimal Polynomials

Let $q_i =: q$ where $q(\xi) = (\xi - 1) \sum_{k=0}^D a_k \xi^k$ with $a_D = 1$ and $\mathbf{a} := [a_0, a_1, \dots, a_D]^T \in \mathbb{R}^{D+1}$, which is not assumed to be available to the observer at initial time. For any subset $\mathcal{S} \subseteq \mathcal{V}$, define

$$z_{\mathcal{S}}(t) = x_{\mathcal{S}}(t) - x_{\mathcal{S}}(t-1), \quad \forall t \geq 1.$$

For any sequence $\{\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_p\}$, define the following block-Hankel matrix

$$M_{p,D}(\{\mathcal{O}_i\}) = \begin{bmatrix} z_{\mathcal{O}_1}(1) & z_{\mathcal{O}_1}(2) & \cdots & z_{\mathcal{O}_1}(D+1) \\ z_{\mathcal{O}_2}(2) & z_{\mathcal{O}_2}(3) & \cdots & z_{\mathcal{O}_2}(D+2) \\ \vdots & \vdots & \ddots & \vdots \\ z_{\mathcal{O}_p}(p) & z_{\mathcal{O}_p}(p+1) & \cdots & z_{\mathcal{O}_p}(D+p) \end{bmatrix} \quad (4.14)$$

Important properties of this matrix is given next.

Theorem 4.4.1. *There exist $p \in \mathbb{Z}_+$ and a sequence $\{\mathcal{O}_i | i = 1, \dots, p, \mathcal{O}_i \subset \mathcal{O}\}$ such that the following hold:*

$$\text{rank}(M_{p,D}(\{\mathcal{O}_i\})) = \text{rank}(M_{p,D-1}(\{\mathcal{O}_i\})) = D, \quad (4.15)$$

$$M_{p,D}(\{\mathcal{O}_i\}) \mathbf{a} = \mathbf{0}. \quad (4.16)$$

Proof. It follows from Section 4.3.2 that for $\forall i \in \mathcal{O}$,

$$\text{rank}(H_{i,D}) = \text{rank}(H_{i,D-1}) = D, \quad H_{i,D} \mathbf{a} = 0, \quad (4.17)$$

where $H_{i,D} \in \mathbb{R}^{(D+1) \times (D+1)}$ is a Hankel matrix given by (4.13). Let $H_{\mathcal{O},D} = [H_{1,D}^T \ H_{2,D}^T \ \dots \ H_{m,D}^T]^T$. It then follows from (4.17) that $\text{rank}(H_{\mathcal{O},D}) = \text{rank}(H_{\mathcal{O},D-1}) = D$ and $H_{\mathcal{O},D} \mathbf{a} = \mathbf{0}$. Now choose $p = D + 1$, $\mathcal{O}_i = \mathcal{O}, i = 1, \dots, p$, and construct $M_{p,D}(\{\mathcal{O}_i\})$ as in (4.14). It is easy to see that $M_{p,D}(\{\mathcal{O}_i\})$ has the same rows as $H_{\mathcal{O},D}$ but in a different order. Thus the claim follows. \square

As a result, once a sequence $\{\mathcal{O}_i\}_{i=1}^p$ satisfying (4.15) is found, \mathbf{a} can be determined from (4.16) and (4.4). Then the consensus value x^* can be computed as in (4.6), i.e.,

$$x^* = \frac{\sum_{k=0}^D a_k x_i(k)}{\sum_{k=0}^D a_k}, \quad \forall i \in \mathcal{O}. \quad (4.18)$$

It is important to note that the number of time steps needed to construct $M_{p,D}(\{\mathcal{O}_i\})$ in (4.16), denoted by T_c , is given by

$$T_c = D + p + 1. \quad (4.19)$$

Clearly, $T_c \geq D + 2$. Now define

$$p^* := \min\{p : (4.15) \text{ holds}\}, \quad T^* := D + 1 + p^*.$$

Thus T^* is the *minimum time* needed for the observer to compute x^* . Note that p^* depends on the choice of $\{\mathcal{O}_i\}_1^{p^*}$. Our next result provides general bounds on this value.

Theorem 4.4.2. *The following hold: $D + 1 \geq p^* \geq \lceil \frac{D}{m} \rceil$.*

Proof. The first inequality follows from the choices of $p = D + 1$ and the sequence $\{\mathcal{O}_i\}$ to construct a particular $M_{p,D}(\{\mathcal{O}_i\})$ in the proof of Theorem 4.4.1. We now show the lower bound. For any $p \in \mathbb{Z}_+$ and $\{\mathcal{O}_i\}$ satisfying (4.15), since $\text{rank}(M_{p,D}(\{\mathcal{O}_i\})) = D$ and $\mathcal{O}_i \subseteq \mathcal{O}, \forall i = 1, \dots, p$, it follows that

$$D \leq \sum_{1 \leq i \leq p} \text{card}(\mathcal{O}_i) \leq \sum_{1 \leq i \leq p} \text{card}(\mathcal{O}) = pm.$$

Thus, $p \geq \lceil \frac{D}{m} \rceil$. Hence the second inequality follows. \square

This result implies that $D+1+\lceil \frac{D}{m} \rceil \leq T^*$, which agrees with Conjecture 4.3.5; here the lower bound is less than that in case (ii) of the conjecture by 1 since to achieve this we have implicitly assumed the knowledge of D .

We can obtain a sharper bound for any initial state $\mathbf{x}(0)$ except for a set of Lebesgue measure zero as follows.

Proposition 4.4.3. *Suppose that the self-weight $w_{ii} > 0, \forall i \in \mathcal{O}$. Then for any initial state $\mathbf{x}(0)$ except for a set of measure zero, the following hold:*

- (i) *If $m \geq D + 1$, then $p^* = 1$.*
- (ii) *If $m \leq D$, then $D + 1 - m \geq p^*$.*

Proof. [Sketch] First, note that if $w_{ii} > 0, \forall i \in \mathcal{O}$, then it can be seen that $\text{rank}(M_{1,D}(\mathcal{O})) = \text{rank}(M_{1,D-1}(\mathcal{O})) = \min(D, m)$ almost surely. Thus, if $m \geq D+1$, then it follows that $\text{rank}(M_{1,D}(\mathcal{O})) = \text{rank}(M_{1,D-1}(\mathcal{O})) = D$, thus $p^* = 1$.

Now if $m \leq D$, choose $p = D + 1 - m$, $\mathcal{O}_1 = \mathcal{O}$. There must exist $j \in \mathcal{O}$ so that if $\mathcal{O}_2 = \dots = \mathcal{O}_p = \{j\}$ then $\text{rank}(M_{p,D}(\{\mathcal{O}_i\})) = \text{rank}(H_{j,D}) = \text{rank}(H_{j,D-1}) =$

$\text{rank}(M_{p,D-1}(\{\mathcal{O}_i\}))$ for almost any $\mathbf{x}(0)$ except for a set of measure zero. This implies that $p^* \leq p = D + 1 - m$. \square

Next, we note that the set of sequences $\{\mathcal{O}_i\}_1^p$ that satisfy (4.15) and achieve p^* includes a special one, namely $\{\mathcal{O}_i \mid \mathcal{O}_i = \mathcal{O}, \forall i = 1, \dots, p\}$. In fact, by defining

$$M_{p,d} := M_{p,d}(\{\mathcal{O}_i \mid \mathcal{O}_i = \mathcal{O}, \forall i = 1, \dots, p\})$$

for any $p, d \geq 1$, we have the following:

Theorem 4.4.4. *Suppose $\{\mathcal{O}_i\}_{i=1}^p$ is a sequence such that $p \geq p^*$ and $\{\mathcal{O}_i\}_{i=1}^{p^*}$ satisfies (4.15). Then for any $d \geq D$,*

$$\text{rank}(M_{p,d}(\{\mathcal{O}_i\}_1^p)) = \text{rank}(M_{p^*,D-1}) = D. \quad (4.20)$$

Proof. The proof follows from (4.15) and the definition of p^* . \square

Condition (4.20) allows us to construct Algorithm 4.1 below to be implemented by the observer to find \mathbf{a} and x^* , assuming the knowledge of D (in addition to the condition that $q_i = q_j, \forall i, j \in \mathcal{O}$). Starting from $p = 1$, the observer repeatedly increases p and checks if $\text{rank}(M_{p,D}) = D$, i.e, if p^* is found.

Algorithm 4.1: Compute \mathbf{a} and x^* for the case of identical minimum polynomials with knowledge of D

Data: The set \mathcal{O} , $m = \text{card}(\mathcal{O})$ and D

- 1 init: $p \leftarrow 1$
- 2 **while** $\text{rank}(M_{p,D}) < D$ **do**
- 3 $p \leftarrow p + 1$
- 4 Compute \mathbf{a} and x^* using (4.16) and (4.18)

In the case where D is not available in advance, we can find D as the first

value of d such that

$$d = \text{rank}(M_{p,d-1}) = \text{rank}(M_{p,d}) = \text{rank}(M_{p+1,d}). \quad (4.21)$$

Here, we want to find the first column-rank defective matrix $M_{p,d}(\{\mathcal{O}\}^p)$ as p and d increase appropriately. Based on this condition, we propose Algorithm 4.2 to determine \mathbf{a} and x^* without the knowledge of D .

Algorithm 4.2: Compute \mathbf{a} and x^* for the case of identical minimum polynomials without knowing D

Data: The set \mathcal{O} , $m = \text{card}(\mathcal{O})$

- 1 init: $d \leftarrow 1$; $p \leftarrow 1$
 - 2 **while** (4.21) *not met* **do**
 - 3 | increase d and/or p
 - 4 Compute \mathbf{a} and x^* using (4.16) and (4.18)
-

Remark 4.4.5. Algorithm 4.2 requires observation time $T^* = D+2+p^* \geq D+2+\lceil \frac{D}{m} \rceil$ since it uses $M_{p+1,D}$. Moreover, when $m = 1$, the algorithm is the same as that in [72], which was summarized in Subsection 4.3.2 above.

4.4.2 Observed Nodes with Different Minimal Polynomials

For any $\mathcal{S} \subseteq \mathcal{O}$, let $q_{\mathcal{S}}$ denote the least common multiple of $\{q_i, i \in \mathcal{S}\}$, which can be regarded as the *joint minimum polynomial* of the set \mathcal{S} . Define $D_{\mathcal{S}} := \text{deg}(q_{\mathcal{S}}) - 1$. Since 1 is a simple root of each q_i , it is also a simple root of $q_{\mathcal{S}}$. Hence, there exists $\mathbf{a} \in \mathbb{R}^{D_{\mathcal{S}}+1}$ such that

$$q_{\mathcal{S}}(\xi) = (\xi - 1) \sum_{0 \leq k \leq D_{\mathcal{S}}} a_k \xi^k, \quad \sum_{0 \leq k \leq D_{\mathcal{S}}} a_k \neq 0, \quad a_{D_{\mathcal{S}}} = 1.$$

Here \mathbf{a} also depends on \mathcal{S} . Now using Algorithm 4.1 or 4.2 above, the observer can determine \mathbf{a} and thus x^* as if $q_i = q_j = q_{\mathcal{S}}$, which requires a minimum observation

time denoted by $T^*(\mathcal{S})$. Therefore, for a given set \mathcal{O} , the (sub)optimal observation time is

$$T_{\mathcal{O}}^* = \min\{T^*(\mathcal{S}) : \mathcal{S} \subseteq \mathcal{O}\}. \quad (4.22)$$

This is clearly a combinatoric problem, whose solution may be hard to find exactly especially when $\Theta(-1) = \emptyset$. If $q_i \in \Theta(-1)$, then we can resort to a greedy algorithm. In any case, $T_{\mathcal{O}}^*$ is upper bounded by $\min_{i \in \mathcal{O}}\{T^*(\{i\})\}$ and $T^*(\mathcal{O})$, which are easier to compute.

To conclude this section, we remark that in the algorithms in [71, 72], each agent i uses only its opinion history to compute x^* and thus the best observation time is $2D_i + 2$. Our results assert that if each agent i also functions as an observer, then the consensus value could be predicted in a fewer number of iterations.

4.5 Numerical Examples

4.5.1 Example 1: Network with Identical Minimal Polynomials

Consider a ring network of $N = 10$ agents with

$$W = \begin{bmatrix} .8 & .1 & 0 & \cdots & 0 & .1 \\ .1 & .8 & .1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ .1 & 0 & 0 & \cdots & .1 & .8 \end{bmatrix} \in \mathbb{R}^{10 \times 10}$$

and with (randomly generated) initial opinions:

$$\mathbf{x}(0) = [0.9797, 0.2848, 0.5949, 0.9621, 0.1857, \\ 0.1930, 0.3416, 0.9329, 0.3906, 0.2732]^T.$$

It can be seen that $\boldsymbol{\pi} = \mathbf{1}/10$ and thus the consensus value is $x^* = \boldsymbol{\pi}^T \mathbf{x}(0) = 0.5139$. Moreover, $q_i = q_j, \forall i, j \in \mathcal{V}$ due to the symmetry of the network and the weight matrix.

First, consider the scenario where each node $i \in \mathcal{V}$ wishes to x^* from its local information and $\Theta_i(-1) = \emptyset$. Using the algorithm in [72], any agent can find $D = 5$ and compute x^* after $2D + 2 = 12$ time steps, where as, by using Algorithm 4.2, each agent (having has 2 neighbors, hence $m = 3$) can find $D = 5$ and compute x^* after $T_c = 9$ time steps; see also Remark 4.4.5.

Next, consider the case where an observer knows D and can monitor m agents in the network. Results in Table 4.1 holds for any choice of \mathcal{O} . (Note also that $T_c \geq D + 2$; see (4.19)). Here $m = D$ is the smallest number of observed nodes that also gives the minimum observation time.

Table 4.1: Observation times using Algorithm 4.1.

| | | | | | | | | | | |
|-------|----|---|---|---|---|---|---|---|---|----|
| m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| T_c | 11 | 9 | 8 | 8 | 7 | 7 | 7 | 7 | 7 | 7 |

4.5.2 Example 2: Network with Different Minimal Polynomials

Consider the graph given in Figure 4.1 (from [71]). In this example,

$$q_1(\xi) = q_2(\xi) = q_3(\xi) = (\xi - 1)(\xi - \lambda_1)(\xi - \lambda_2)(\xi - \lambda_3),$$

$$q_4(\xi) = (\xi - \lambda_4)q_1(\xi), \quad q_5(\xi) = q_6(\xi) = (\xi - \lambda_5)q_4(\xi).$$

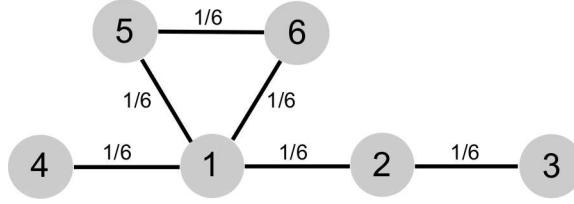


Figure 4.1: Network example 2. Self weights are not shown.

In Table 4.2, we compare the observation times obtained by the algorithm in [72] with those by Algorithm 4.2, where each node monitors its neighbors and naively uses Algorithm 4.2. It is interesting to see that the observation time of node 1 is longer than that of node 2 although it has more neighbors. The reason is that node 1 uses information from nodes 5 and 6, which have the largest observation time among all (or to be precise, the joint minimum polynomial of nodes 1, 5 and 6 is q_6 , which is of highest order).

Table 4.2: Observation time for each node to compute consensus value in Example 2

| Node | T_c by [72] | T_c by Alg. 4.2 | Observed nodes |
|------|---------------|-------------------|----------------|
| 1 | 8 | 8 | {1,2,4,5,6} |
| 2 | 8 | 6 | {1,2,3} |
| 3 | 8 | 7 | {2,3} |
| 4 | 10 | 9 | {1,4} |
| 5 | 12 | 10 | {1,5,6} |
| 6 | 12 | 10 | {1,5,6} |

Finally, suppose that the observer is able to select any m nodes to monitor. Let $\mathcal{O}^* = \arg \min_{\mathcal{O} \subseteq \mathcal{V}, \text{card}(\mathcal{O})=m} T_{\mathcal{O}}^*$, i.e., the set that gives the optimal observation time. The result given in Table 4.3 is obtained by brute-force computations. Thus, the best choice would be $\mathcal{O}^* = \{1, 2\}$ with $T^* = 6$ and only 2 nodes being monitored.

Table 4.3: Optimal time T^* when the observer can choose any m nodes

| m | T^* | \mathcal{O}^* |
|-----|-------|------------------------|
| 1 | 8 | $\{1\}, \{2\}, \{3\}$ |
| 2 | 7 | $\{1, 2\}$ |
| 3 | 6 | $\{1, 2, 3\}$ |
| 4 | 6 | $\{1, 2, 3, 4\}$ |
| 5 | 6 | $\{1, 2, 3, 4, 5\}$ |
| 6 | 6 | $\{1, 2, 3, 4, 5, 6\}$ |

4.6 Toward Selecting Observed Nodes

Recall that the second question in Section 4.2.1 is about optimal selection of observed nodes. Based on the previous section, the (sub)optimal solution to this problem can be given by $\mathcal{O}^* = \arg \min_{\mathcal{O} \subseteq \mathcal{V}, \text{card}(\mathcal{O})=m} T_{\mathcal{O}}^*$. The optimal solution to this problem is not obvious and left for future work. Here, instead, we give *heuristic descriptions* of \mathcal{O}^* , including:

- (A) q_i 's should be similar;
- (B) $\deg(q_i)$'s should be small;
- (C) p^* should be close to $\frac{D_{\mathcal{O}^*}}{m}$.

Note that the degree of q_i and the relationships between q_i and q_j depend not only on the structure of the network, but also on the weight matrix. To have a closer look at the minimal polynomial of a node, let us consider the following. Let $\mathbf{c}^{(i)} = [c_0^{(i)}, c_1^{(i)}, \dots, c_{D_i}^{(i)}, 1]^T \in \mathbb{R}^{D_i+2}$ be vector of coefficients of q_i , i.e., $q_i(\xi) =$

$\xi^{D_i+1} + \sum_{k=0}^{D_i} c_k^{(i)} \xi^k$. From definition,

$$\mathbf{0}^T = \mathbf{e}_i^T q_i(W) = \sum_{k=1}^{D_i+1} c_k^{(i)} \mathbf{e}_i^T W^k = (\mathbf{c}^{(i)})^T \begin{bmatrix} \mathbf{e}_i^T \\ \mathbf{e}_i^T W \\ \vdots \\ \mathbf{e}_i^T W^{D_i+1} \end{bmatrix} \quad (4.23)$$

Thus, it can be shown that

Proposition 4.6.1. $\deg(q_i)$ is the observability index of the pair (\mathbf{e}_i^T, W) .

Let us revisit Example 2 in the previous section. We keep the network structure but consider the following two weight matrices: W_1 corresponds to equal neighbors weights, and W_2 is randomly generated.

$$W_1 = \begin{bmatrix} 1/5 & 1/5 & 0 & 1/5 & 1/5 & 1/5 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 \end{bmatrix} \quad W_2 = \begin{bmatrix} .19 & .24 & 0 & .13 & .17 & .27 \\ .54 & .15 & .31 & 0 & 0 & 0 \\ 0 & .27 & .73 & 0 & 0 & 0 \\ .63 & 0 & 0 & .37 & 0 & 0 \\ .22 & 0 & 0 & 0 & .35 & .45 \\ .32 & 0 & 0 & 0 & .28 & .40 \end{bmatrix}$$

In the case of W_1 , we have

$$q_1 = q_2 = q_3 = q_4, \quad \deg(q_1) = 5, \quad q_5 = q_6 = q_{W_1}, \quad \deg(q_5) = 6.$$

In the case of W_2 , we have

$$q_1 = q_2 = \dots q_6 = q_{W_2}, \quad \deg(q_{W_2}) = 6.$$

Clearly, changing agents' weights can change the agents' minimal polynomials as well as the degrees. However, it is also apparent that certain properties of q_i are pertinent to the network structure and thus are related to the structural observability concept. This direction of investigation is left for future work.

In the following, we restrict ourself to the class of undirected graphs and explore necessary and/or sufficient conditions in terms of graph theory to meet descriptions (A)-(C) above. We also assume that that the weight matrix is given by

$$W = I - \epsilon L \tag{4.24}$$

where $L := D^{in} - A$ is the Laplacian matrix, A is the adjacency matrix, $D^{in} = \text{diag}(A\mathbf{1})$ is the in-degree matrix, and $\epsilon \in (0, \min_i [D^{in}]_{ii}^{-1})$ (which is to ensure that W is a positive weight matrix).

Next we introduce some graph notions [129].

Definition 4.6.2. (*Automorphism*) An *automorphism* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a permutation ψ of \mathcal{V} such that

$$(\psi(i), \psi(j)) \in \mathcal{E} \Leftrightarrow (i, j) \in \mathcal{E} \tag{4.25}$$

Proposition 4.6.3. ([129]) *Let A be the adjacency matrix of the graph \mathcal{G} and ψ a permutation on its node set \mathcal{V} . Associate with this permutation the permutation matrix P . Then ψ is an automorphism of \mathcal{G} if and only if $PA = AP$.*

In the following, a partition of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is denoted by $\mathfrak{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ for some appropriate k , where \mathcal{C}_i 's are called cells.

Definition 4.6.4. (*Almost Equitable Partition-AEP*) Suppose $\mathfrak{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ is a partition of a graph \mathcal{G} .

- \mathfrak{C} is said to be *almost equitable* if each node in \mathcal{C}_i has the same number of neighbors in $\mathcal{C}_j, \forall i, j \in \{1, \dots, k\}, i \neq j$.
- \mathfrak{C} is said to be *almost equitable w.r.t. node v* if \mathfrak{C} is an AEP and $\{v\} \in \mathfrak{C}$.
- The *minimum AEP w.r.t node v* , denoted by \mathfrak{C}_v^* , is an AEP such that $\{v\} \in \mathfrak{C}$ and $\text{card}(\mathfrak{C}_v^*)$ is minimal.

Definition 4.6.5. (*Distance Regular Graph*) An undirected graph \mathcal{G} is said to be *regular* if $\text{deg}(i) = \text{deg}(j), \forall i, j \in \mathcal{V}$. It is called *distance-regular* if it is regular and for any pair of nodes $u, v \in \mathcal{V}$ with $\text{dist}(u, v) = i, 0 < i < \text{diam}(\mathcal{G})$, there exist numbers f_i and g_i such that there are f_i neighbors of v that are of distance $i - 1$ from u and g_i neighbors of v that are of distance $i + 1$ from u .

4.6.1 When $q_i = q_j$?

We have the following result.

Proposition 4.6.6. *If there exists an automorphism ψ of \mathcal{G} such that $\psi(i) = j$, then $q_i = q_j$. The converse is not true.*

Proof. Suppose there exists an automorphism ψ of \mathcal{G} such that $\psi(i) = j$. Let P be the permutation matrix associated with ψ . By Proposition 4.6.3, we have $P^T A = AP^T$ and thus $P^T D^{in} = D^{in} P^T$. Then

$$\begin{aligned} WP^T &= (I - \epsilon(D^{in} - A))P^T = P^T - \epsilon(D^{in} P^T - AP^T) \\ &= P^T - \epsilon(P^T D^{in} - P^T A) = P^T W. \end{aligned}$$

From this, it can be show that $W^k P^T = P^T W^k$ for any integer $k \geq 1$. Note also that $P\mathbf{e}_i = \mathbf{e}_j$. Then, multiplying both sides of (4.23) with P^T yields

$$\mathbf{0}^T = (\mathbf{c}^{(i)})^T \begin{bmatrix} \mathbf{e}_i^T P^T \\ \mathbf{e}_i^T W P^T \\ \dots \\ \mathbf{e}_i^T W^{D_i+1} P^T \end{bmatrix} = (\mathbf{c}^{(i)})^T \begin{bmatrix} \mathbf{e}_i^T P^T \\ \mathbf{e}_i^T P^T W \\ \dots \\ \mathbf{e}_i^T P^T W^{D_i+1} \end{bmatrix} = (\mathbf{c}^{(i)})^T \begin{bmatrix} \mathbf{e}_j^T \\ \mathbf{e}_j^T W \\ \dots \\ \mathbf{e}_j^T W^{D_i+1} \end{bmatrix}$$

Therefore, $\mathbf{e}_j^T q_i(W) = 0$. Since q_j is the minimal polynomial of node j , it follows that $q_j | q_i$. Next, we note that since ψ is an automorphism, ψ^{-1} exists and corresponds to the permutation matrix P^T . That is, $\psi^{-1}(j) = i$, or $P^T \mathbf{e}_j = \mathbf{e}_i$. Now apply the same argument as above, we have $q_i | q_j$. Therefore, $q_i = q_j$.

The converse is not true; see Section 4.5.2 for a counterexample, where $q_1 = q_2$ but there's no permutation between nodes 1 and 2. \square

This proposition only provides sufficient conditions. Consider again Example 4.5.2. Using this Proposition, we can only conclude that $q_5 = q_6$ but not $q_1 = q_2 = q_3$. This result is useful when the graph is highly symmetric. E.g., consider again the example in Subsection 4.5.1, it is easy to see that $q_i = q_j, \forall i, j \in \mathcal{V}$.

We will show in the next subsection that for distance regular graphs, all the agents' minimal polynomials are the same; see Proposition 4.6.9.

4.6.2 Bounds on $\deg(q_i)$

Using the fact that $\deg(q_i)$ is the observability index of the pair (\mathbf{e}_i^T, W) , we can obtain some bounds on the degree of q_i as follows.

Theorem 4.6.7. [72, 130, 131] *If \mathcal{G} is connected and undirected, it holds that*

$$\text{diam}(\mathcal{G}, i) + 1 \leq \deg(q_i) \leq \text{card}(\mathfrak{C}_i^*) \quad (4.26)$$

where $\text{diam}(\mathcal{G}, i) = \max_{v \in \mathcal{V}} \text{dist}(i, v)$ is the longest distance from node i to any others, and \mathfrak{C}_i^* denotes the minimum almost equitable partition w.r.t. node i .

In some special cases, for example in distance regular graphs, the upper bound and lower bound are tight. Examples of distance regular graphs include cycles, hypercubes and complete graphs.

Theorem 4.6.8. [132] *If \mathcal{G} is a distance regular graph, then*

$$\text{diam}(\mathcal{G}) + 1 = \deg(q_i) = \text{card}(\mathfrak{C}_i^*), \quad \forall i \in \mathcal{V} \quad (4.27)$$

Based on this result, we have the following.

Proposition 4.6.9. *If \mathcal{G} is a distance regular graph, then*

$$q_i = q_j, \quad \forall i, j \in \mathcal{V}. \quad (4.28)$$

Proof. It is well known that the adjacency matrix A of a distance regular graph has $\text{diam}(\mathcal{G}) + 1$ distinct eigenvalues [129]. Thus, so does W since $W = I - \epsilon(D^{in} - A) = (I - \epsilon D^{in}) + \epsilon A$ and $D^{in} = f_0 I$ for some f_0 . Moreover, the minimal polynomial of W , q_W is of degree $\text{diam}(\mathcal{G}) + 1$ with $\text{diam}(\mathcal{G}) + 1$ distinct roots. Now by Theorem 4.6.8, $\deg(q_i) = D_i + 1 = \text{diam}(\mathcal{G}) + 1, \forall i \in \mathcal{V}$. Moreover, $q_i | q_W, \forall i \in \mathcal{V}$. Therefore, $q_i = q, \forall i \in \mathcal{V}$ and the Proposition follows. \square

4.7 Limitations and Future Work

The main drawbacks of the method presented in this chapter are as follows. First it can only be applied to networks with fixed topology and linear time invariant dynamics. Second, the computational accuracy of the rank of a (block-) Hankel matrix does not scale well with the size of the matrix. Third, since the prediction value is in the limit as time tend to infinity, the accuracy is sensitive to computational error and may not work well in the presence of observation noises and/or communication delays. Extension and further development to overcome these limitations are thus important directions for future work.

Among possible directions for future work besides resolving the validity of Conjecture 4.3.5, we note a possible application of consensus prediction in network monitoring for misbehavior; see Section 7.2.

Part III: Distributed Optimization

Chapter 5: Local Prediction for Enhanced Convergence of Distributed Optimization Algorithms

Abstract: This chapter studies distributed optimization problems where a network of agents seeks to minimize the sum of their private cost functions. Algorithms are proposed that build on past consensus-based distributed optimization algorithms by incorporating a local predictive step as developed in Chapter 4. The algorithms involve introduction of local optimization variables at the network nodes alongside the original local node states. In the first algorithm, the local optimization variables are updated cyclically through a subgradient step while the opinion variables follow a traditional consensus protocol periodically interrupted by a predictive consensus estimate re-set operation. For convex cost functions with bounded subdifferentials, this algorithm is guaranteed to converge to within some range of the optimal value if using a constant step size or to the optimal value if a diminishing step size is in place. For differentiable cost functions whose sum is convex and has a Lipschitz continuous gradient, convergence to the optimal value can be ensured when using a constant step size, even if some of the individual cost functions are nonconvex. In addition, exponential convergence to the optimal solution is achieved when the private cost functions are further assumed to be strongly convex. In these cases, each

optimization variable behaves like the centralized subgradient method except at a slower time scale. The last two algorithms are specialized for the case of quadratic cost functions and converge in finite time to the optimal solution or a neighborhood of arbitrarily small size. Simulation examples are given to illustrate the algorithms.

5.1 Introduction

We consider a network of N agents, without a central coordinating unit, aiming to cooperatively solve the global optimization problem

$$\min_{x \in X} F(x) := \sum_{i=1}^N f_i(x) \quad (5.1)$$

where $f_i : \mathbb{R} \rightarrow \mathbb{R}$ represents the private cost function of agent i , X is a nonempty constraint set known to all the agents, and it is assumed that each agent is able to communicate with its direct neighbors.

Solving this problem in a distributed fashion calls for strategies of cooperation among all the agents in the network. In this regard, many distributed algorithms have been developed; see e.g., [32,42,76–78,80–85,133] and references therein. Among them, the class of distributed (sub)gradient-based algorithms is well known for its simplicity in implementation and generally mild assumptions imposed on the local cost functions and the network topology. In particular, this class requires each function f_i to be (at least) convex and usually with bounded subgradient or Lipschitz continuous gradient.

Major limitations of algorithms in the category are also well known. First, the convergence of many algorithms depends on the choice of step size sequences. When

a constant step size is used, both Distributed Gradient Descent and Distributed Subgradient methods only yield convergence to a neighborhood of the optimal solution and of the optimal value [78, 86]. This occurs even if the f_i are strongly convex and have Lipschitz continuous gradients, and is thus one of the main differences between these methods and their centralized counterparts. This motivates the use of particular diminishing or adaptive step sizes to achieve asymptotic convergence. However, the convergence rate can be very slow (compared to that of the centralized method), depending on the step size sequence, whose appropriate selection is not trivial. Second, many incremental subgradient methods require all the agents to construct a closed cycle in order to pass an estimate of the solution around the network; see e.g., [32, 88, 89]. Third, even when asymptotic convergence is guaranteed, it is not obvious how each agent can locally decide when to stop the algorithm without affecting other agents' estimates. Put differently, there are no simple criteria for all the agents to stop at the same time while also sharing the same estimate of an optimal solution. This is also true for most (if not all) other distributed optimization methods.

When all the local cost functions are quadratic, many other consensus-based algorithms can outperform those in the subgradient class. For example, the ratio consensus method can be used to solve problem (5.1) without constraints and converges exponentially [34, 90]. Based on this method, [91] proposed a Newton-Raphson-like algorithm which also converges asymptotically for a class of functions having continuous, strictly positive and bounded second derivatives, assuming a sufficiently small discretization step.

Our main contributions are as follows. We propose and study a new distributed optimization technique for solving (5.1) on a fixed and directed network; the new technique involves use of a distributed prediction scheme based on node minimal polynomials introduced in Chapter 4. Specifically, we first present a distributed subgradient-type algorithm for the general setting of problem (5.1) but without constraints (i.e., $X = \mathbb{R}$) which we show has convergence rate similar to that of the centralized (sub)gradient method. In fact, the convergence rate to the optimal value is $O(\ln(t)/\sqrt{t})$ under a diminishing step size for the case where the cost functions f_i are convex and have bounded subgradients, while for the case in which the total cost function F is convex and has Lipschitz continuous gradient, it is $O(1/t)$ under a constant step size. In the former case, an optimal choice of the step size can yield a rate $O(1/\sqrt{t})$, which is the best achievable for both centralized and distributed subgradient methods [117, 134]. In the latter case, if F is further assumed to be strongly convex, then we obtain both exponential convergence to the optimal value as well as exponential convergence to the optimal solution. The performance of our algorithm also resembles that of the centralized subgradient method in that all the agents, in finite time, agree on an identical estimate of a solution and continue to agree thereafter (and possibly approach the global optimal solution), solving problem (5.1) as if they all knew the global function F . Moreover, this algorithm is among very few of gradient type that can deal with the case where some of the local cost functions are nonconvex, as long as the total cost F is convex and has a Lipschitz continuous gradient. Next, with some modifications, we also extend the algorithm to cope with constraints $x \in X$ and non-column stochasticity of the weight

matrix. Finally, in the case where all the functions f_i are quadratic and the problem is unconstrained, we construct two algorithms, one of which is ratio-consensus-like and converges in finite time (which is the fastest convergence achieved to-date in distributed optimization methods), and the other is a gradient-based algorithm that achieves near finite-time convergence. The convergence times of our algorithms scale at most linearly with the network size. In fact, they are linear in the maximum degree of the agents' minimal polynomial, which ranges between the diameter and size of the network.

For comparison, we report here known convergence rates/time of other distributed (sub)gradient methods. First, for convex cost functions with bounded subgradients, [135] shows that the distributed proximal-gradient method under a diminishing step size $O(1/\sqrt{t})$ achieves a rate of $O(\ln(t)/\sqrt{t})$. A similar convergence rate was obtained for the dual averaging method in [81] and for the subgradient-push method in [85]. The recent work [134] presents an algorithm with convergence time linear in the network size, and states that it is the best convergence time so far for problem (5.1) with non-differentiable convex functions. Our first algorithm admits an even better convergence time. Second, for convex cost functions with Lipschitz continuous gradient, the algorithm in [136] uses a second-order update at each iteration, which yields a convergence rate of $O(1/t)$ in terms of best running violation to the first-order optimality condition under a fixed step size. Note that our first algorithm also attains the same rate but in terms of the objective error. Third, for convex cost functions with bounded and Lipschitz continuous gradients, [87] proposed two fast distributed gradient methods; one of which converges

at rate $O(\ln(t)/t)$ under a diminishing step size, while the other achieves $O(1/t^2)$ convergence through the use of an inner consensus loop and Nesterov’s acceleration technique (see, [117, Chap. 2]). Finally, when the global cost function F is strongly convex and all f_i have Lipschitz continuous gradients, the algorithm in [136] also converges at a linear rate.

The rest of the chapter is organized as follows. Section 5.2 contains the problem formulation and some background on subgradient methods and the finite-time consensus prediction introduced in Chapter 4. The main algorithm and convergence results for general cost functions (some possibly nonconvex) are given in Section 5.3. Performance limits of our algorithms are discussed in Section 5.5, which is followed by some simulation examples in Section 5.6 to illustrate the algorithms. Finally, concluding remarks are given in Section 5.7. Most proofs are given in Appendix A.3.

5.2 Problem Statement and Background

5.2.1 Problem Statement

Consider a network consisting of N agents where the underlying communication is characterized by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The objective of all the agents is to solve problem (5.1), repeated here for convenience:

$$\min_{x \in X} F(x) = \sum_{i=1}^N f_i(x), \quad (5.2)$$

where $X \subseteq \mathbb{R}$ is the constraint set. Note that here we assume variable x to be scalar for simplicity of notation. The case of a vector variable can be treated following the

same steps. Let F^* and X^* denote the optimal value and the optimal solution set, respectively, i.e., $X^* = \{x^* \in X, F(x^*) = F^* := \min_{x \in X} F(x)\}$. The following is a blanket assumption:

Assumption 5.2.1. *The set X is convex and X^* is nonempty.*

In our setting, agent i only has access to f_i and local information on its neighbors' opinions, and no central coordinating node is assumed to exist. Thus, the agents need to collaborate in a distributed manner to solve problem (5.1). This involves local iterative computation along with information diffusion. We are interested in the scenario where the communication graph \mathcal{G} connecting the agents is directed and fixed. We make the following additional blanket assumption.

Assumption 5.2.2. (Network Connectivity and Weight Matrix) *The graph \mathcal{G} is fixed and strongly connected. The weight matrix W is fixed, row-stochastic and satisfies $w_{ij} > 0$ for $(i, j) \in \mathcal{E}, i \neq j$, and $w_{ij} = 0$ otherwise. Moreover, W has at least one positive diagonal element.*

5.2.2 Subgradient Methods

Subgradient methods are the simplest numerical algorithms for solving problem (5.1) for the case in which each function f_i is convex. Recall that for the unconstrained problem (i.e., when $X = \mathbb{R}$), the centralized method is based on the iteration (see e.g., [115, Chap. 6], [117, Chap. 3])

$$x(t+1) = x(t) - \gamma(t)g(x(t)), \quad (5.3)$$

where $\gamma(t)$ is the step size at iteration t and $g(x(t))$ is a subgradient of F at $x(t)$, i.e., $g(x(t)) \in \partial F(x(t))$. In the special case where F is differentiable, $g(x(t)) = \nabla F(x(t))$ and (5.3) reduces to the centralized gradient descent method.

In the distributed setting described above, subgradient methods can take many forms (see, e.g., [76, 78]), one of which is as follows. Every agent has its own estimate of an optimal solution, and updates it at each iteration by combining a consensus step with a local optimization step:

$$x_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t) - \gamma(t) g_i(x_i(t)), \quad (5.4)$$

where $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ is the set of in-neighbors of node i (including i), $\gamma(t)$ the step size known to all agents, and $g_i(x_i(t)) \in \partial f_i(x_i(t))$. This algorithm is usually referred to as the Distributed Subgradient Method, or DSM for short. A modified version called the Distributed Projected Subgradient method or DPS (see, e.g., [137]) is used when the problem includes a global constraint $x \in X$. In DPS,

$$x_i(t+1) = P_X \left(\sum_{j \in \mathcal{N}_i} w_{ij} x_j(t) - \gamma(t) g_i(x_i(t)) \right), \quad (5.5)$$

where P_X denotes the projection operator onto the set X (assuming that each agent is able to perform this operation). The convergence of these distributed methods depends on the step size sequence and the weight matrix W . Unlike the centralized version, these algorithms are only guaranteed to converge to an error neighborhood of the optimal solution when ∇f_i 's are Lipschitz continuous and a constant step size is used (see, e.g., [78, 86]).

In the following, we develop new distributed algorithms based on (5.4) that can achieve *faster convergence* and *may not require a diminishing step size*; convergence

can even occur in finite time for quadratic cost functions. Moreover, the convergence is similar to that of the centralized subgradient algorithm in the sense that the agents are able to stop the algorithm at the same time with identical estimates of an optimal solution. The main novelty introduced in this chapter is to take advantage of the finite-time consensus protocol introduced in [71] (described in the next subsection) to result in a modified version of algorithm (5.4) which enjoys accelerated convergence in comparison to (5.4). We find that the new algorithm can even achieve the performance limit of distributed algorithms in many cases (see discussion in Section 5.5). Another benefit of using the new protocol is that it can be implemented in a distributed manner for an arbitrary weight matrix (as long as the associated graph is strongly connected).

5.2.3 Finite-Time Consensus Using Minimal Polynomials

This subsection is a brief summary of 4.2.2. Consider the following update iteration

$$x_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t), \quad \forall i \in \mathcal{V}, \quad (5.6)$$

or in vector form: $\mathbf{x}(t+1) = W\mathbf{x}(t)$. From Subsection 1.5.2 in Chapter 1, we know that under Assumption 5.2.2

$$\exists \lim_{t \rightarrow \infty} W^t = \mathbf{1}\boldsymbol{\pi}^T =: \Phi, \quad (5.7)$$

where $\boldsymbol{\pi} \in \mathbb{R}^N$ is the normalized left Perron eigenvector of W , that is, $\boldsymbol{\pi}^T W = \boldsymbol{\pi}^T$ and $\mathbf{1}^T \boldsymbol{\pi} = 1$. Therefore, the network in (5.6) asymptotically reaches consensus

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \Phi \mathbf{x}(0). \quad (5.8)$$

From Section 4.2.2 in Chapter 4, we also know that each agent i can locally compute the consensus value using its minimal polynomial q_i . In particular,

$$\lim_{t \rightarrow \infty} x_i(t) = \frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(l)}{\sum_{l=0}^{D_i} a_l^{(i)}}, \quad (5.9)$$

where $D_i = \deg(q_i) - 1$ and $\mathbf{a}^{(i)} = [a_0^{(i)}, a_1^{(i)}, \dots, a_{D_i}^{(i)}]^T$ satisfies:

$$q_i(\xi) = (\xi - 1) \sum_{l=0}^{D_i} a_l^{(i)} \xi^l, \quad a_{D_i}^{(i)} = 1, \quad \sum_{l=0}^{D_i} a_l^{(i)} \neq 0. \quad (5.10)$$

Note also that $\mathbf{a}^{(i)}$ can be computed locally and in finite time by agent i . We summarize the analysis above as follows.

Theorem 5.2.3. (Prediction of consensus value by minimal polynomial) *Consider system (5.6) for $t = 0, \dots, \bar{D} - 1$, where $\bar{D} = \max_{i \in \mathcal{V}} D_i$ with $\deg(q_i) = D_i + 1$. Let Assumption 5.2.2 hold. Then*

$$\frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(l)}{\sum_{l=0}^{D_i} a_l^{(i)}} = \mathbf{e}_i^T \Phi \mathbf{x}(0), \quad \forall i \in \mathcal{V}, \quad (5.11)$$

where $\mathbf{a}^{(i)} = [a_0^{(i)}, a_1^{(i)}, \dots, a_{D_i}^{(i)}]^T \in \mathbb{R}^{D_i+1}$ are given in (5.10).

In this regard, the node minimal polynomials q_i can be viewed as a tool providing a shortcut to reaching consensus. This idea will be employed in this chapter to develop new distributed algorithms with desirable features such as behavior similar to that of centralized algorithms, distributed stopping criteria, and finite (or practically finite) time convergence.

5.3 Distributed Subgradient Optimization Using Finite Time Consensus

In this section, we return to optimization problem (5.1) and show how minimal polynomials associated with the agents can be used to improve the convergence speed of the distributed subgradient method (5.4). To this end, we will assume that each agent $i \in \mathcal{V}$ knows its minimal polynomial q_i and a common upper bound κ on $\deg(q_i)$, i.e.,

$$\kappa \geq \deg(q_i), \forall i \in \mathcal{V}. \quad (5.12)$$

Note that the least possible value of κ is always less than or equal to the number of agents N in the network; see Section 5.5.1 for further discussion on this. Therefore, κ can be chosen to be N or a known upper bound on N .

We now consider problem (5.1) and the following possible assumptions, which will not be invoked together below.

Assumption 5.3.1. *For each $i \in \mathcal{V}$, f_i is convex and has bounded subgradients on X , i.e., $\exists L_i \in (0, \infty)$ such that $|g_i(x)| \leq L_i, \forall g_i(x) \in \partial f_i(x), \forall x \in X$.*

Assumption 5.3.2. *For each $i \in \mathcal{V}$, f_i is differentiable on the interior of X . Moreover, the function F is convex on X and the gradient ∇F is $L_{\nabla F}$ -Lipschitz continuous for some $L_{\nabla F} \in (0, \infty)$.*

The former assumption implies that F and the f_i are convex and have bounded subgradients while not necessarily being differentiable; the latter requires convexity of F while not requiring convexity of each f_i .

5.3.1 Main Algorithm

Our main idea is to combine the consensus prediction step offered by (5.9) with the distributed subgradient method (5.4). Specifically, we propose an algorithm, called Finite-time consensus Aided Distributed Optimization (FADO), that performs the following 3 sequential steps in a cyclic manner: (i) κ iterations of usual consensus algorithm (5.6) (used to diffuse information in the network), followed by (ii) a prediction step using minimal polynomials, and then (iii) a (sub)gradient optimization step applied to the predicted consensus value obtained from step (ii). The detailed algorithm is as follows.

Algorithm 5.1. (Finite-time consensus Aided Distributed Optimization - FADO).

Each agent $i \in \mathcal{V}$ initializes a pair of local variables $(s_i(0), x_i(0))$ in X and updates them for $t \geq 1$ according to

$$s_i(t) = \begin{cases} \frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(t - \kappa + l)}{\sum_{l=0}^{D_i} a_l^{(i)}} & \text{if } t = k\kappa \\ s_i(t - 1) & \text{else} \end{cases} \quad (5.13a)$$

$$s_i(t) = \begin{cases} s_i(t) - \gamma_k g_i(s_i(t)) & \text{if } t = k\kappa \\ s_i(t - 1) & \text{else} \end{cases} \quad (5.13b)$$

$$x_i(t) = \begin{cases} s_i(t) - \gamma_k g_i(s_i(t)) & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t - 1) & \text{else} \end{cases} \quad (5.14a)$$

$$x_i(t) = \begin{cases} s_i(t) - \gamma_k g_i(s_i(t)) & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t - 1) & \text{else} \end{cases} \quad (5.14b)$$

where γ_k is the step size at $t = k\kappa$.

Remark 5.3.3. (On implementation of (5.13a)) The foregoing form (5.13a) of the algorithm is quite useful for analysis, but in practice, instead of storing $D_i + 1$ values of x_i 's to compute s_i , each agent can just maintain a single memory register to store a running sum from $k\kappa$ to $(k + 1)\kappa$, denoted by y_i , and update it as a new estimate

$x_i(t)$ becomes available as follows. Agent i sets $y_i(t) = \hat{a}_0^{(i)} x_i(t)$ at time $t = k\kappa$ and then updates this variable as $y_i(k\kappa + \tau + 1) = y_i(k\kappa + \tau) + \hat{a}_\tau^{(i)} x_i(k\kappa + \tau)$ for $\tau = 0, \dots, D_i$, where $\hat{a}_\tau^{(i)} = a_\tau^{(i)} / \sum_{l=0}^{D_i} a_l^{(i)}$. Then $s_i((k+1)\kappa) = y_i(k\kappa + D_i + 1)$. Of course, each agent still needs to store $D_i + 1$ normalized coefficients $\hat{a}_\tau^{(i)}$.

Clearly, information exchanged among the agents at each time involves only $x_i(t)$, but not $s_i(t)$. Moreover, each agent i only needs to update $s_i(t)$ once every κ time steps. Note that whenever $t = k\kappa$, (5.13a) must be carried out prior to (5.14a). The next result asserts that by utilizing minimal polynomials as done in the main algorithm (5.13a) above, we succeed in forcing the states $s_i(t)$ to be identical over the whole network after an initial time period of length κ , i.e., $s_i(t) = s_j(t), \forall t \geq \kappa, \forall i, j \in \mathcal{V}$. In fact, it can be seen that the $s_i(t)$'s will be similar to each other for all time $t \geq 0$ if identically initialized.

Theorem 5.3.4. (Agreement of $s_i, i \in \mathcal{V}$ after κ steps) *Consider (5.13)-(5.14) and let Assumption 5.2.2 hold. If g_i is bounded for any $i \in \mathcal{V}$, then*

$$s_i(t) = s_j(t), \quad \forall i, j \in \mathcal{V}, \forall t \geq \kappa. \quad (5.15)$$

Proof. First, since g_i is bounded $\forall i \in \mathcal{V}$, we have that $x_i(t)$ in (5.14) is well-defined. Also, (5.13) is well-defined as $\sum_{l=0}^{D_i} a_l^{(i)} \neq 0$; cf. (5.10).

Next, for any $k \geq 0$, by (5.14b) we have

$$\mathbf{x}(t) = W\mathbf{x}(t-1), \quad \forall t = k\kappa + 1, \dots, k\kappa + \kappa - 1. \quad (5.16)$$

Then at time $t = (k+1)\kappa$, we have

$$s_i((k+1)\kappa) \stackrel{(5.13a)}{=} \frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(k\kappa + l)}{\sum_{l=0}^{D_i} a_l^{(i)}} = \mathbf{e}_i^T \Phi \mathbf{x}(k\kappa), \quad (5.17)$$

where the last equality follows from Theorem 5.2.3 and (5.12). Here Φ is the consensus matrix defined in (5.7). Therefore,

$$s_i((k+1)\kappa) = \mathbf{e}_i^T \mathbf{1} \boldsymbol{\pi}^T \mathbf{x}(k\kappa) = \boldsymbol{\pi}^T \mathbf{x}(k\kappa), \quad (5.18)$$

which is independent of i . It remains to use (5.13b). \square

In the rest of this subsection, we establish the convergence of the algorithm when problem (5.1) is unconstrained (i.e., $X = \mathbb{R}$) and the weight matrix is doubly stochastic. In the next subsection, we will show how to modify our algorithm to allow relaxing of these assumptions (i.e., the problem may have constraints and W may be only row stochastic.)

Our first convergence result deals with convex cost functions with bounded subgradients.

Theorem 5.3.5. (Local cost functions with bounded subgradients) *Consider problem (5.1) with $X = \mathbb{R}$. Let Assumptions 5.2.1, 5.2.2 and 5.3.1 hold. Assume further that W is doubly stochastic. Let all the agents perform (5.13)-(5.14). Let $\bar{s}(t) := s_i(t), \forall i \in \mathcal{V}, \forall t \geq \kappa$ and $g(\bar{s}(k\kappa)) := \sum_{i=1}^N g_i(\bar{s}(k\kappa)) \in \partial F(\bar{s}(k\kappa))$. Then*

$$\bar{s}((k+1)\kappa) = \bar{s}(k\kappa) - \frac{\gamma_k}{N} g(\bar{s}(k\kappa)), \quad (5.19)$$

Also, let

$$\hat{s}_k := \frac{\sum_{\tau=1}^k \gamma_\tau \bar{s}(\tau\kappa)}{\sum_{\tau=1}^k \gamma_\tau}. \quad (5.20)$$

We then have the following:

(i) If $\gamma_k \equiv \gamma > 0$, then for each $i \in \mathcal{V}$

$$\lim_{k \rightarrow \infty} F(\hat{s}_k) - F^* \leq \frac{\gamma}{2N} L_F^2, \quad L_F := \sum_{j \in \mathcal{V}} L_j \quad (5.21)$$

(ii) For a given number of iterations T , let $K = \lfloor T/\kappa \rfloor$. Let R be any number such that $R \geq \text{dist}(\bar{s}(\kappa), X^*)$. Then with constant step size $\gamma_k \equiv \frac{NR}{L_F \sqrt{K}}$, we have

$$F(\hat{s}_K) - F^* \leq \frac{RL_F}{\sqrt{K}}. \quad (5.22)$$

(iii) If $\gamma_k > 0$, $\lim_{k \rightarrow \infty} \gamma_k = 0$, and $\sum_{k=1}^{\infty} \gamma_k = \infty$, then

$$\lim_{k \rightarrow \infty} F(\hat{s}_k) = F^*, \quad \forall i \in \mathcal{V}. \quad (5.23)$$

In fact, if $\gamma_k = \frac{1}{\sqrt{k}}$, the convergence rate is $O(\frac{\ln k}{\sqrt{k}})$.

Proof. See Appendix A.3.1. □

Remark 5.3.6. (On convergence and limit points) The auxiliary sequence $\{s_i(t)\}_{t=0}^{\infty}$ is generated to yield convergence of $F(\hat{s}_k)$ to F^* (case (iii)) or an optimal value neighborhood (cases (i) and (ii)), rather than $x_i(t)$ directly. In general, $x_i(t)$ does not converge but rather reaches a limit cycle of period κ whenever $s_i(t)$ converges (possibly also to a limit cycle). Note that by definition (5.20) and the fact that $\bar{s}(t) = s_i(t), \forall i \in \mathcal{V}, \forall t \geq \kappa$, \hat{s}_k is a global variable that is available to all the agents. Moreover, each agent i can compute \hat{s}_k using its local variable $s_i(t)$ and an augmented running sum $\Gamma_k \in \mathbb{R}$ in a recursive manner as follows:

$$\Gamma_{k+1} = \Gamma_k + \gamma_k \quad (5.24)$$

$$\hat{s}_{k+1} = (\Gamma_k \hat{s}_k + \gamma_{k+1} s_i((k+1)\kappa)) / \Gamma_{k+1}, \quad (5.25)$$

where $\Gamma_0 = 0, \hat{s}_0 = 0$ (here the subscript k denotes the iteration index in the slow time scale $k = \lfloor t/\kappa \rfloor$). When a constant step size γ is used, the following simplified update is sufficient: $\hat{s}_{k+1} = (k\hat{s}_k + s_i((k+1)\kappa))/(k+1)$.

Remark 5.3.7. (Stopping criteria) By Theorem 5.3.4, all the agents are able to stop at the same time with the same estimate $s_i = \bar{s}$ (or \hat{s}_k) of the optimal solution if they use a common stopping criterion, e.g., running the algorithm for a predetermined number of iterations T as in Theorem 5.3.5(ii), or until one of the following holds (see Appendix A.3.9 for other criteria):

$$|s_i((k+1)\kappa) - s_i(k\kappa)| \leq \epsilon \tag{5.26}$$

$$|F(\hat{s}_{k+1}) - F(\hat{s}_k)| \leq \epsilon. \tag{5.27}$$

This is not the case for many other distributed algorithms, where any consensus can only be achieved in the sense of an asymptotic limit. Note that $F(\hat{s}_k)$ involves evaluation of the global cost function F at \hat{s}_k and algorithm (5.13)-(5.14) does not provide it to each agent. However, we show in Appendix A.3.9 that it is possible for each agent, by using augmented iterations, to locally compute $F(\hat{s}_k)$ at time $t = (k+1)\kappa$ for any $k \geq 1$.

Remark 5.3.8. (Connection with convergence of centralized method) In light of (5.15) and (5.19), our algorithm performs analogously to the centralized subgradient method (5.3) except on a slower time scale, where F is convex with subgradient g bounded in magnitude by $L_F = \sum_{j=1}^N L_j$. As a result, it adopts performance guarantee of the centralized subgradient method, as shown in (5.21)-(5.23). For detailed analysis of this centralized method, see, e.g., [117, Chap. 3] and [138]. It should be noted that

the convergence our algorithm depends on not only that of the virtual centralized subgradient iteration (5.19) but also the process of information diffusing over the network carried out by (5.14b) (hence the importance of strong connectivity).

Remark 5.3.9. (Step size design and objective bound in case (ii)) For a given T , the constant step size $\gamma = \frac{NR}{L_F\sqrt{K}}$ depends on the constant R and the ratio L_F/N . First, it is clear that the smallest value of R , which is $\text{dist}(\bar{s}(\kappa), X^*)$, minimizes the error bound in (5.22) but is rather of theoretical interest only since it requires the knowledge of the solution set X^* (note that $\bar{s}(\kappa) = s_i(\kappa)$). However, in practice, an upper (possibly loose) bound R may be inferred, especially when there is some restriction on the range of global variable in (5.1); see also Subsection 5.3.2 and examples in Section 5.6. Second, the ratio L_F/N is indeed the average of Lipschitz constants of the agents' local cost functions and thus can be computed locally in finite time. For example, using the same algorithm above with $x_i(0) = L_i$, then by (5.18) and double stochasticity of W , we have $s_i(\kappa) = \boldsymbol{\pi}^T \mathbf{x}(0) = \sum_{i \in \mathcal{V}} L_i/N$. Alternatively, the agents can choose $\gamma = \frac{R}{L_{\max}\sqrt{K}}$ instead, where $L_{\max} := \max_{i \in \mathcal{V}} L_i$ which can be computed in a distributed manner and in finite time using a max-consensus protocol [139]. In this case, the corresponding bound in (5.22) becomes

$$F(\hat{s}_K) - F^* \stackrel{\text{(A.22)}}{\leq} \frac{R}{\sqrt{K}} \left(\frac{NL_{\max}}{2} + \frac{L_F^2}{2NL_{\max}} \right) \leq \frac{RNL_{\max}}{\sqrt{K}},$$

where the last inequality follows from $L_F \leq NL_{\max}$.

Remark 5.3.10. (On best convergence rate and scalability) It should be pointed out that the result in case (ii) of Theorem 5.3.5 demonstrates an improvement on best analyses of distributed subgradient algorithms so far. To wit, recall that for any

given number of iterations T , we have $K\kappa \leq T < (K+1)\kappa$ and thus the following:

$$\frac{1}{N}(F(\hat{s}_K) - F^*) \stackrel{(5.22)}{\leq} \frac{RL_F}{N\sqrt{K}} \leq RL_{\max} \sqrt{\frac{K+1}{K}} \sqrt{\frac{\kappa}{T}}. \quad (5.28)$$

That is, the convergence rate of the (averaged) objective error is of order $O(RL_{\max} \sqrt{\frac{\kappa}{T}})$; this also holds for $\gamma = \frac{R}{L_{\max}\sqrt{K}}$ (as shown earlier). In other words, the time it takes for the bound in (5.28) to drop below $\epsilon > 0$ is $O(\kappa/\epsilon^2)$. As N increases, this turns out to be the fastest rate achieved to-date among distributed subgradient-based methods for convex cost functions with bounded subgradients. In this setting, the best known rate so far was demonstrated in [134], where the authors considered the problem of minimizing $\frac{1}{N} \sum_{i=1}^N f_i(x)$ (which is why we need a factor of $\frac{1}{N}$ on the left side of (5.28) for comparison). In this reference, the authors proposed a linear time average consensus protocol and used it to design a new algorithm for solving problem (5.1). The consensus protocol there is a combination of a weighted averaging scheme based on the Metropolis rule [90] and an extrapolation step (which is similar to the idea of adding momentum to speed up iterative methods). Under a fixed step size $\beta = \frac{1}{L_{\max}\sqrt{NT}}$, [134] showed that the aggregated objective error decays as $O(R^2 L_{\max} \sqrt{\frac{N}{T}})$, i.e., it takes $O(N/\epsilon^2)$ time steps to reach an ϵ error. Thus this algorithm scales linearly in the network size; hence the title “linear time.” To implement this algorithm (specifically the Metropolis rule), however, the graph needs to be undirected.

Our algorithm, on the other hand, applies to directed graphs and requires $O(\kappa/\epsilon^2)$ steps to reach an ϵ error. In general, without knowledge of the network topology, we can just take κ to be N or a known upper bound on N , then our

algorithm still possesses the fastest rate (which is similar to the algorithm in [134], where a common upper bound on N is also required). However, for certain graphs such as distance regular graphs, κ can be as small as the graph diameter, which can be small compared to the network size (see Section 5.5.1 for further discussion on this). In this connection, our algorithm convergence time scales at most linearly in the network size.

Note also that the centralized subgradient method converges to an ϵ error of the optimal value in $O(1/\epsilon^2)$ time steps. Hence, our algorithm performance lies between that of the centralized method and the best known rate for distributed ones. This intuitively makes sense since our algorithm is not only distributed but also behaving like the centralized one except the time-scale is slowed down by a factor of κ .

Remark 5.3.11. (On number of subgradient evaluations) Another important aspect that should be noted is the number of subgradient evaluations since the computation of subgradients usually dominates the time it takes to perform the optimization step. Within T time steps, our algorithm requires each agent to evaluate its subgradient T/N times, whereas most, if not all, other distributed subgradient algorithms require T evaluations.

Of course, the advantages of our algorithm are based on the assumption that all the agents are equipped with their own minimal polynomial. Although this seems restrictive at first, we remark that the agents' minimal polynomials can be computed (prior to the main algorithm's implementation) in a centralized or decentralized

manner, or could be done on-the-fly and in finite time as well; see [71, 72, 140] for such algorithms and also Section 5.6 for numerical examples.

Remark 5.3.8 also suggests that known results on the centralized gradient descent method can be used in a straightforward manner to show the convergence of the algorithm when assuming smoothness condition on F (rather than on every f_i). In particular, when F is differentiable, convergence to the optimal value can be ensured with sufficiently small constant step size as shown next.

Theorem 5.3.12. (Convex global cost with Lipschitz gradient) *Consider problem (5.1) with $X = \mathbb{R}$. Let Assumptions 5.2.1, 5.2.2 and 5.3.2 hold. Assume further W is doubly stochastic. Let all the agents synchronously perform (5.13)-(5.14) with $\gamma_k \equiv \gamma \in (0, \frac{2N}{L_{\nabla F}})$. Then for each $i \in \mathcal{V}$*

$$F(s_i(t)) - F^* = O(\kappa/t), \quad \text{as } t \rightarrow \infty. \quad (5.29)$$

Proof. See Appendix A.3.2. □

Remark 5.3.13. (Convergence comparison) Similar to the centralized Gradient Descent method, when F is convex and ∇F Lipschitz continuous, the proposed algorithm converges to the optimal value without the need for a diminishing step size. This is a key difference between our algorithm and the distributed gradient descent method (5.4) and many others, which do not converge to the optimal value under the constant step size rule. Moreover, a running bound on the objective error is available in (A.27), bearing a resemblance to that of the centralized method except for being scaled by a factor of κ .

Faster convergence rates can be obtained when we assume further strong convexity of the global cost function F (but not every individual cost f_i). In particular, the algorithm achieves linear convergence rates to both the optimal value and the optimal solution. (Note that under Assumption 5.2.1 and strong convexity of F , there exists a unique $x^* \in X^*$).

Theorem 5.3.14. (Strongly convex global cost with Lipschitz continuous gradient)

Consider problem (5.1) with $X = \mathbb{R}$. Let Assumptions 5.2.1, 5.2.2 and 5.3.2 hold.

Assume further W is doubly stochastic and F is strongly convex with parameter

$\mu > 0$. If all the agents perform (5.13)-(5.14) with $\gamma_k \equiv \gamma \in (0, \frac{2N}{\mu + L_{\nabla F}}]$, then

$$|s_i(k\kappa) - x^*|^2 \leq \beta^{k-1} |s_i(\kappa) - x^*|^2, \quad (5.30)$$

$$F(s_i(k\kappa)) - F^* \leq \frac{1}{2} L_{\nabla F} \beta^{k-1} |s_i(\kappa) - x^*|^2 \quad (5.31)$$

where $\beta = 1 - \frac{2\gamma\mu L}{N(\mu + L_{\nabla F})} \in (0, 1)$. Thus, $F(s_i(t)) \rightarrow F^*$ and $s_i(t) \rightarrow x^*$ linearly at rates $\beta^{\frac{1}{\kappa}}$ and $\beta^{\frac{1}{2\kappa}}$, respectively.

Proof. See Appendix A.3.3. □

5.3.2 Extensions of the Algorithm 5.1

We now consider possible extensions of the algorithm to deal with two cases: (i) problem (5.1) is subject to a constraint $x \in X$ for some convex set X , and (ii) the weight matrix W is only row stochastic.

Case (i): We assume X is closed and convex. To satisfy this constraint, we resort to a projection operator onto X , denoted P_X . In the special case $X \subset \mathbb{R}$, X is just an interval, thus the projection is simply a cut-off function.

Assuming that all the agents know the set X , we modify our algorithm described above as follows. For any $i \in \mathcal{V}$, initialize $s_i(0) = x_i(0) \in X$, and update for any $t \geq 1$

$$s_i(t) = \begin{cases} P_X \left(\frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(t-\kappa+l)}{\sum_{l=0}^{D_i} a_l^{(i)}} \right) & \text{if } t = k\kappa \\ s_i(t-1) & \text{else} \end{cases} \quad (5.32)$$

$$x_i(t) = \begin{cases} s_i(t) - \gamma_k g_i(s_i(t)) & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t-1) & \text{else} \end{cases} \quad (5.33)$$

The key idea of this extension is that the same P_X is used by all the agents, forcing the modified algorithm to work in the same manner as the original one, as shown next.

Define $\mathbf{g}_s(t) := [g_1(s_1(t)), \dots, g_N(s_N(t))]^T$ and $\bar{s}(t) := \mathbf{1}^T \mathbf{s}(t)/N$, and recall that $\Phi = \mathbf{1}\mathbf{1}^T/N$. We have

$$\begin{aligned} s_i(k\kappa + \kappa) &\stackrel{(5.32)}{=} P_X \left(\frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(k\kappa + l)}{\sum_{l=0}^{D_i} a_l^{(i)}} \right) \stackrel{(\text{Thm.5.2.3})}{=} P_X(\mathbf{e}_i^T \Phi \mathbf{x}(k\kappa)) \\ &\stackrel{(5.33)}{=} P_X \left(\frac{\mathbf{1}^T}{N} (\mathbf{s}(k\kappa) - \gamma_k \mathbf{g}_s(k\kappa)) \right) \\ &= P_X \left(\bar{s}(k\kappa) - \frac{\gamma_k}{N} \sum_{j=1}^N g_j(s_j(k\kappa)) \right). \end{aligned} \quad (5.34)$$

Thus, $s_i(k\kappa) = \bar{s}(k\kappa), \forall i \in \mathcal{V}, k \geq 1$. Hence, (5.15) holds and

$$\bar{s}((k+1)\kappa) = P_X \left(\bar{s}(k\kappa) - \frac{\gamma_k}{N} g(\bar{s}(k\kappa)) \right), \quad (5.35)$$

where $g(\bar{s}(k\kappa)) = \sum g_i(\bar{s}(k\kappa)) \in \partial F(\bar{s}(k\kappa))$. Now (5.35) is the usual (centralized) projected (sub)gradient method, whose convergence results are not much different from those of the (sub)gradient method (see, e.g., [117, 138], [115, Chap. 2]). Therefore, the conclusions of Theorems 5.3.5, 5.3.12 and 5.3.14 still hold.

Case (ii): It should be noted that in distributed settings, a row stochastic weight matrix is much easier to implement than a column (or doubly) stochastic one as each agent can individually decide the weight on the information received from its neighbors. When this is the case, most (if not all) subgradient-based methods do not converge to the optimal value; our proposed algorithm above is no exception. However, it can be modified to overcome this by using the re-weighting technique as in [93, 100] which requires that the value π_i is available to agent i for all $i \in \mathcal{V}$. The modified algorithm is as follows:

$$s_i(t) = \begin{cases} \frac{\sum_{l=0}^{D_i} a_l^{(i)} x_i(t - \kappa + l)}{\sum_{l=0}^{D_i} a_l^{(i)}} & \text{if } t = k\kappa \\ s_i(t - 1) & \text{else} \end{cases} \quad (5.36)$$

$$x_i(t) = \begin{cases} s_i(t) - \gamma_k \frac{g_i(s_i(t))}{N\pi_i} & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t - 1) & \text{else} \end{cases} \quad (5.37)$$

The only difference between this extension and the original algorithm (5.13)-(5.14) is the scaling factor $(N\pi_i)^{-1}$ of the subgradient in (5.37) where $\pi_i > 0, \forall i \in \mathcal{V}$ (see [95, Thm. 8.4.4]). Note that the factor N^{-1} is not crucial to the convergence of the algorithm as its appearance is merely to retain the conclusions in Theorems 5.3.5, 5.3.12 and 5.3.14; see Appendix A.3.4 for a detailed proof.

Here we assume that the value π_i is available to agent i for all $i \in \mathcal{V}$. In fact, each agent can compute its corresponding entry in the vector $\boldsymbol{\pi}$ in finite time (possibly during the process of determining its own minimal polynomial). In particular, consider iteration (5.6) with initial condition $\mathbf{x}(0) = \mathbf{e}_i$ for some $i \in \mathcal{V}$ and suppose

that Assumption 5.2.2 holds. Then,

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \lim_{t \rightarrow \infty} W^t \mathbf{x}(0) = \mathbf{1} \boldsymbol{\pi}^T \mathbf{e}_i = \pi_i \mathbf{1}. \quad (5.38)$$

That is, the network of agents running (5.6) achieves consensus $\bar{x} = \pi_i$. Therefore, by applying one of finite-time consensus algorithms in, e.g., [71, 72] (see also Subsection 5.2.3), agent i can compute π_i in finite time. This is also the main idea employed in [141] for each agent to compute $\boldsymbol{\pi}$.

Remark 5.3.15. When problem (5.1) is subject to both a global constraint $x \in X$ and a row stochastic weight matrix W , we can combine (5.32) and (5.37) together to form a new algorithm. The proof for convergence of the so-obtained algorithm is merely based on the proofs of both cases above, and thus is skipped for brevity.

5.4 Finite-Time Optimization for Quadratic Cost Functions

Now we consider problem (5.1) without any constraint, i.e., $X = \mathbb{R}$, and with quadratic cost functions

$$f_i(x) = b_i(x - c_i)^2, \quad \forall i \in \mathcal{V} \quad (5.39)$$

for some $b_i, c_i \in \mathbb{R}$ with $b_i > 0$. Clearly, the optimal solution is given by $x^* = (\sum_{i=1}^N b_i c_i) / (\sum_{i=1}^N b_i)$. Of course, our algorithm in the previous section is still applicable. Here we aim to achieve (near) finite-time convergence algorithms by capitalizing on the special form of the cost functions.

5.4.1 Ratio-Consensus based Algorithm

Our first algorithm is based on the observation that x^* can be expressed as the ratio of two average quantities, namely, $x^* = (\frac{1}{N} \sum_{i=1}^N b_i c_i) / (\frac{1}{N} \sum_{i=1}^N b_i)$. Thus, inspired by the idea of the ratio-consensus algorithm (see, e.g., [34, 90]), we construct the following finite time algorithm:

Algorithm 5.2. (Finite-time Ratio-consensus) *Let κ satisfy (5.12). Each agent $i \in \mathcal{V}$ initializes a pair of local variables (y_i, z_i) at time $t = 0$ as $y_i(0) = b_i c_i, z_i(0) = b_i$ and updates them according to*

$$y_i(t) = \sum_{j \in \mathcal{N}_i} w_{ij} y_j(t-1), \quad t = 1, \dots, N \quad (5.40)$$

$$z_i(t) = \sum_{j \in \mathcal{N}_i} w_{ij} z_j(t-1), \quad t = 1, \dots, N \quad (5.41)$$

$$x_i^* = \frac{\sum_{l=0}^{D_i} a_l^{(i)} y_i(l)}{\sum_{l=0}^{D_i} a_l^{(i)} z_i(l)}, \quad \forall i \in \mathcal{V}, \quad (5.42)$$

Note that (5.42) is evaluated only once at final time $t = \kappa + 1$. The next result is immediate.

Theorem 5.4.1. (Finite-time optimization for quadratic costs) *Consider problem (5.1) with $X = \mathbb{R}$ and f_i given as in (5.39). Let Assumption 5.2.2 hold, and further let W be doubly stochastic. If the agents perform (5.40)-(5.42), then*

$$x_i^* = x^*, \quad \forall i \in \mathcal{V}. \quad (5.43)$$

Proof. We have $\Phi = \mathbf{1}\mathbf{1}^T/N$ by (5.7) and double stochasticity of W . Application of Theorem 5.2.3 yields $\sum_{l=0}^{D_i} a_l^{(i)} y_i(l) = \frac{1}{N} \mathbf{1}^T \mathbf{y}(0) \sum_{l=0}^{D_i} a_l^{(i)}$ and $\sum_{l=0}^{D_i} a_l^{(i)} z_i(l) = \frac{1}{N} \mathbf{1}^T \mathbf{z}(0) \sum_{l=0}^{D_i} a_l^{(i)}$. The theorem then follows. \square

Although the idea of this algorithm is simple, it, to our knowledge, has not been presented elsewhere in the literature. It also shows an interesting connection with the finite-time behavior of the centralized gradient descent method. In particular, for a quadratic cost function, the centralized method converges in just one iteration by using the Newton step. In a distributed setting, we observe the same finite time converging behavior of our algorithm above, except in κ steps.

Here, at each time $t = 1, \dots, N$, each agent exchanges its pair of variables (y_i, z_i) with its neighbors. Consequently, all the agents reach the optimal solution indirectly through diffusing the coefficients of their quadratic cost functions. In the case that the agents do not want to reveal information about their private cost functions, it is still possible for each agent to use its minimal polynomial in connection with exchanging estimates of the solution with its neighbors to achieve very fast convergence. In the following, we derive such an algorithm.

5.4.2 Gradient-based Algorithm

Our idea is as follows. Consider distributed gradient method (5.4) applied to the problem with the quadratic costs (5.39):

$$x_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t) - \gamma b_i(x_i(t) - c_i), \quad (5.44)$$

where γ is a constant and $x_i(0)$ can be chosen arbitrarily. This iteration does not converge to the optimal solution ¹ but rather an $O(\gamma)$ -neighborhood of x^* , even

¹which can be easily verified by contradiction, i.e., if it does $x^* = \sum_{j \in \mathcal{N}_i} w_{ij} x^* - \gamma b_i(x^* - c_i)$, then $x^* = c_i, \forall i \in \mathcal{V}$.

when W is assumed to be doubly stochastic (see, e.g., [78, 86]):

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = O(\gamma)\boldsymbol{\epsilon} + x^* \mathbf{1}. \quad (5.45)$$

for some $\boldsymbol{\epsilon} \in \mathbb{R}^N$. Therefore, if each agent can predict its final value in finite time (possibly in the same manner as above), then by using a sufficiently small γ all the agents may employ a very few prediction steps in conjunction with a finite number of iterations (5.44) in order to obtain a close estimate to the optimal solution. This idea will be pursued in the following.

To this end, we first show how each agent can compute its final value in (5.44) in finite time and in a distributed manner. This is different from what is reported in subsection 5.2.3 since (5.44) is not a consensus iteration. Eq. (5.44) represents a linear time-invariant system with constant inputs, which can also be expressed in vector form as

$$\mathbf{x}(t+1) = (W - \gamma B)\mathbf{x}(t) + \gamma B\mathbf{c}, \quad (5.46)$$

where $B := \text{diag}([b_1, \dots, b_N])$ and $\mathbf{c} := [c_1, \dots, c_N]^T$. In this connection, it is clear that the convergence of iteration (5.44) depends on the system matrix $(W - \gamma B)$.

We then have the following:

Theorem 5.4.2. (Stability condition) *Let Assumption 5.2.2 hold. Suppose that W has positive diagonal elements. There exists $\gamma_0 > 0$ such that*

$$\rho(W - \gamma B) < 1, \quad (5.47)$$

for any $\gamma \in (0, \gamma_0)$. Moreover, if , we can take $\gamma_0 = \frac{2 \min_{i \in \mathcal{V}} w_{ii}}{\max_{i \in \mathcal{V}} b_i}$.

Proof. Note that $\frac{2\min_{i \in \mathcal{V}} w_{ii}}{\max_{i \in \mathcal{V}} b_i} > 0$. By the Gershgorin circle theorem (see, e.g., [95, p. 344]), all the eigenvalues of $(W - \gamma B)$ are located in the union of N discs

$$\bigcup_{i=1}^N \{z \in \mathbb{C} : |z - w_{ii} + \gamma b_i| \leq 1 - w_{ii}\}. \quad (5.48)$$

As a result, $\rho(W - \gamma B) < 1$ for any $\gamma \in (0, \frac{2\min_{i \in \mathcal{V}} w_{ii}}{\max_{i \in \mathcal{V}} b_i})$. The existence of γ_0 then follows. \square

Condition (5.47) guarantees that the system (5.46) is BIBO stable, thus the states converge to some fixed values, which are not necessarily equal. In fact, it follows from (5.46) that

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \Phi_\gamma \mathbf{c}, \quad \Phi_\gamma := (I - (W - \gamma B))^{-1} \gamma B, \quad (5.49)$$

where $I - (W - \gamma B)$ is invertible because of (5.47).

Let $\mathbf{z}(t)^T \triangleq [\mathbf{x}(t)^T, \mathbf{c}^T]$. The system (5.46) can be described equivalently as

$$\mathbf{z}(t+1) = \tilde{W} \mathbf{z}(t), \quad \tilde{W} \triangleq \begin{bmatrix} W - \gamma B & -\gamma B \\ \mathbf{0}_{N \times N} & I \end{bmatrix}. \quad (5.50)$$

For any $i = 1, \dots, N$, define \tilde{q}_i to be the monic polynomial of minimum degree such that $\mathbf{e}_i^T \tilde{q}_i(\tilde{W}) = \mathbf{0}^T$, where $\mathbf{e}_i \in \mathbb{R}^{2N}$ is the i -th standard unit vector. We will call \tilde{q}_i the minimal polynomial of node i in system (5.50).

Lemma 5.4.3. (Minimal polynomials in system (5.50)) *Let γ satisfy (5.47). For each $i \in \mathcal{V}$, there exists $\tilde{\mathbf{a}}^{(i)} \in \mathbb{R}^{\tilde{D}_i+1}$ such that*

$$\tilde{q}_i(\xi) = (\xi - 1) \sum_{j=0}^{\tilde{D}_i} \tilde{a}_j^{(i)} \xi^j, \quad \tilde{a}_{\tilde{D}_i}^{(i)} = 1, \quad (5.51)$$

where $\deg(\tilde{q}_i) = \tilde{D}_i + 1 \leq 1 + n$. Moreover, all the zeros of \tilde{q}_i are strictly inside the unit circle except for one at 1.

Proof. See Appendix [A.3.6](#). □

Clearly, \tilde{q}_i is of the same form as q_i in (5.10) and also has 1 as the only zero of maximum modulus. Thus, \tilde{q}_i can also be computed locally and in finite time using the schemes presented in Section [4.3.2](#).

Now we assume that all the agents know a common upper bound κ on $\deg(\tilde{q}_i)$. Note that $\deg(\tilde{q}_i) \leq N + 1, \forall i \in \mathcal{V}$. Thus κ can be chosen to be $N + 1$ or an upper bound on $N + 1$. After κ consecutive iterations of (5.44), each agent is able to determine its own final value to which it will converge if all the agents follow (5.44) forever. Using the same arguments as (5.10)-(5.9) in Subsection [5.2.3](#) we have

$$\lim_{t \rightarrow \infty} x_i(t) = \left(\sum_{k=0}^{\bar{D}_i} \tilde{a}_k^{(i)} x_i(k) \right) / \left(\sum_{k=0}^{\bar{D}_i} \tilde{a}_k^{(i)} \right), \quad \forall i \in \mathcal{V}. \quad (5.52)$$

Therefore, in the same spirit of Theorem [5.2.3](#), we can view the right side of (5.52) and κ iterations of (5.44) as a realization of the operator Φ_γ given in (5.49). This realization is carried out in finite time, enabling us to construct the following algorithm:

Algorithm 5.3. (Near Finite-time Gradient-based Optimization) *Each agent $i \in \mathcal{V}$ initializes a pair of local variables (s_i, x_i) at time $t = 0$ as $s_i(0) = x_i(0) = c_i$ and*

updates them for $t \geq 1$ according to

$$s_i(t) = \begin{cases} \frac{\sum_{k=0}^{\bar{D}_i} \tilde{a}_k^{(i)} x_i(t - \kappa + k)}{\sum_{k=0}^{\bar{D}_i} \tilde{a}_k^{(i)}}, & \text{if } t = k\kappa \\ s_i(t - 1), & \text{else} \end{cases} \quad (5.53)$$

$$x_i(t) = \begin{cases} s_i(t), & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t - 1) - \gamma b_i (x_i(t - 1) + s_i(t - 1)), & \text{else} \end{cases} \quad (5.54)$$

where γ satisfies (5.47).

In the following, we show that by choosing γ appropriately, this algorithm achieves exponential convergence to the optimal solution. More importantly, the convergence rate is adjustable using γ .

Theorem 5.4.4. (Convergence of consensus matrix Φ_γ) *Let Assumption 5.2.2 hold and γ satisfy (5.47). Then Φ_γ given by (5.49) is a row stochastic and irreducible matrix, and has $B\boldsymbol{\pi}$ as a left Perron eigenvector. Moreover,*

$$\lim_{k \rightarrow \infty} \Phi_\gamma^k = \mathbf{1}\boldsymbol{\pi}^T B / (\boldsymbol{\pi}^T B \mathbf{1}), \quad (5.55)$$

and the convergence is exponential with rate determined by the second largest eigenvalue $\lambda_2(\Phi_\gamma)$.

Proof. See Appendix A.3.7. □

Theorem 5.4.4 allows us to prove the convergence of (5.53)-(5.54).

Theorem 5.4.5. (Convergence for quadratic costs) *Consider Algorithm (5.53)-(5.54) and let Assumption 5.2.2 hold. Assume further that W is doubly stochastic*

and has positive diagonal elements. Let γ satisfy (5.47). Then

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \lim_{t \rightarrow \infty} \mathbf{s}(t) = x^* \mathbf{1}. \quad (5.56)$$

Moreover, the convergence is linear with rate $|\lambda_2(\Phi_\gamma)|^{\frac{1}{\kappa}}$.

Proof. See Appendix A.3.8. □

Although in this theorem the weight matrix W is assumed to be doubly stochastic, we note that the algorithm can be modified using the re-weighting trick so that W can be taken to be only row stochastic (see Remark 5.4.8 for details).

Theorem 5.4.5 shows the effect of γ on the convergence of Algorithm (5.53)-(5.54) and the rate at which the agents' estimates converge linearly to the optimal solution. Specifically, γ needs to be chosen so as to first ensure the stability of the algorithm (namely, condition (5.47) in Theorem 5.4.2), and then accelerate the convergence speed by reducing the rate $|\lambda_2(\Phi_\gamma)|^{\frac{1}{\kappa}}$. The next remarks successively address these issues.

Remark 5.4.6. (Distributed agreement on step size to ensure stability condition)

In order to guarantee the convergence of the algorithm, it is necessary that all the agents select the same value of step size γ that satisfies (5.47). Such a value can be determined by all the agents in finite time and in a distributed manner as follows. Prior to running Algorithm (5.53)-(5.54), all the agents can implement a max-consensus algorithm (see, e.g., [139]) in order to compute both $b_M := \max_{i \in \mathcal{V}} b_i$ and $a_M := -\max_{i \in \mathcal{V}}(-w_{ii})$. Then set $\gamma = \varepsilon 2a_M/b_M$, where $\varepsilon \in (0, 1)$ is a constant known to all the agents. Clearly, $0 < \gamma < \frac{2a_M}{b_M} = \frac{2 \min_{i \in \mathcal{V}} w_{ii}}{\max_{i \in \mathcal{V}} b_i} \leq \gamma_0$, by Theorem

5.4.2. Moreover, the max-consensus algorithm converges after a finite number of iterations equal to the diameter of the graph. In case this number is unknown to all the agents, any upper bound (e.g., N , the network size) can also be used to terminate this algorithm.

Remark 5.4.7. (Fast convergence by choosing small step size γ) Recall from Theorem 5.4.5 that $\mathbf{x}(t)$ converges linearly to $x^*\mathbf{1}$ with rate $|\lambda_2(\Phi_\gamma)|^{\frac{1}{\kappa}}$, where it can be seen from (5.55) that

$$|\lambda_2(\Phi_\gamma)| = \rho\left(\Phi_\gamma - \frac{\mathbf{1}\boldsymbol{\pi}^T B}{\boldsymbol{\pi}^T B \mathbf{1}}\right) = \rho\left(\Phi_\gamma - \frac{\mathbf{1}\mathbf{b}^T}{\mathbf{b}^T \mathbf{1}}\right). \quad (5.57)$$

Here, $\boldsymbol{\pi} = \mathbf{1}/N$ since W is doubly stochastic. Moreover,

$$\Phi_\gamma \mathbf{c} \stackrel{(5.45)\&(5.49)}{=} O(\gamma)\boldsymbol{\epsilon} + x^*\mathbf{1} = O(\gamma)\boldsymbol{\epsilon} + \mathbf{1}\mathbf{b}^T \mathbf{c} / (\mathbf{b}^T \mathbf{1}),$$

where we have used the fact that $x^* = \mathbf{b}^T \mathbf{c} / \mathbf{b}^T \mathbf{1}$. Rearranging terms yields $\left(\Phi_\gamma - \frac{\mathbf{1}\mathbf{b}^T}{\mathbf{b}^T \mathbf{1}}\right) \mathbf{c} = O(\gamma)\boldsymbol{\epsilon}$, which, in view of (5.57), then implies that

$$\lim_{\gamma \rightarrow 0^+} |\lambda_2(\Phi_\gamma)| = 0. \quad (5.58)$$

Although Φ_γ is not defined at $\gamma = 0$, choosing γ small brings about a fast convergence rate. Moreover, when γ is sufficiently small so that $|\lambda_2(\Phi_\gamma)|^{\frac{1}{\kappa}}$ is close to 0, system (5.53)-(5.54) exhibits a near **dead-beat** response. When this is the case, all the agents may agree to perform the algorithm for $k\kappa$ steps with a small integer k (e.g., 1, 2, 3) so that each agent obtains a close approximation to the optimal solution. In conjunction with Remark 5.4.6, all the agents may choose $\gamma = \min(\theta, \varepsilon 2a_M/b_M)$, where $0 < \theta, \varepsilon \ll 1$ are supposedly known to every agent. A word of warning, however, is that too small a value of γ (say 10^{-16}) could possibly affect the convergence

of the algorithm due to computational round-off errors. This deserves more analysis in future work.

Remark 5.4.8. (Extension of Algorithm (5.53)-(5.54) for row stochastic weight matrix) Assume that π_i is available to agent i . (Indeed, all the agents can cooperate to compute their corresponding π_i in finite time [141].) We modify (5.54) as follows:

$$x_i(t) = \begin{cases} s_i(t), & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} x_j(t-1) - \frac{\gamma b_i (x_i(t-1) + s_i(t-1))}{N\pi_i}, & \text{else} \end{cases}$$

and redefine $\Phi_\gamma = \left(\gamma^{-1} B^{-1} S (I - W) + I \right)^{-1}$, where $S := \text{diag}(N\boldsymbol{\pi})$ and γ is such that $\rho(W - \gamma S^{-1} B) < 1$ in place of stability condition in (5.47). Note also that γ_0 in Theorem 5.4.2 can be chosen to be $\gamma_0 = \frac{2 \min_i n \pi_i w_{ii}}{\max_i b_{ii}}$. Under this condition, it can be verified that Φ_γ is still a valid consensus matrix with $\mathbf{b} = B\mathbf{1}$ being a left Perron eigenvector. Therefore,

$$\lim_{k \rightarrow \infty} \mathbf{s}(k\kappa) = \lim_{k \rightarrow \infty} \Phi_\gamma^k \mathbf{c} = \mathbf{1} \mathbf{b}^T \mathbf{c} / (\mathbf{b}^T \mathbf{1}) = \mathbf{1} x^*,$$

i.e., the optimal solution is achieved by every agent.

5.5 On Minimal Value of κ and Performance Limits of Distributed Subgradient Methods

5.5.1 Minimal Value of κ

It is evident that the convergence speed of each algorithm presented in Sections 5.3 and 5.4 depends on the value κ , which is an upper bound on the degrees of

all the minimal polynomials $(q_i, i \in \mathcal{V})$. Indeed, a smaller κ corresponds to faster convergence. Thus, in the best scenario, $\kappa = \kappa_{\min} := \max_{i \in \mathcal{V}} \deg(q_i)$. We note the following regarding this value.

First, for general directed networks, κ_{\min} satisfies

$$\text{diam}(\mathcal{G}) + 1 \leq \kappa_{\min} \leq \deg(q_W) \leq N, \quad (5.59)$$

where $\text{diam}(\mathcal{G})$ denotes the graph diameter. The lower bound can be shown, e.g., by application of [72, Thm. 3]; an alternative argument is given in the next subsection. The upper bounds follow from definitions of minimal polynomials and the Cayley-Hamilton theorem.

It is interesting to seek classes of graphs for which the lower bound is achieved. Clearly, this is the case for line graphs since then $\text{diam}(\mathcal{G}) + 1 = N$. Next, we show that the lower bound can be achieved with another class of graphs, namely distance regular graphs (see Definition 4.6.5), of which examples include cycles, hypercubes and complete graphs. References [129, 142] and a recent survey [143] provide further information on distance regular graphs. The next result asserts that in the setting of distance regular graphs, we can ensure that $\kappa_{\min} = \text{diam}(\mathcal{G}) + 1$. See Appendix A.3.5 for a proof.

Theorem 5.5.1. *If \mathcal{G} is distance regular and $W = I - \epsilon L(\mathcal{G})$, where $L(\mathcal{G})$ is the Laplacian matrix and $\epsilon > 0$ satisfying $\epsilon(|\mathcal{N}_i| - 1) \leq 1$, then $\text{diam}(\mathcal{G}) + 1 = \deg(q_i), \forall i \in \mathcal{V}$.*

Thus, for distance regular graphs (with $W = I - \epsilon L(\mathcal{G})$), the convergence times of our algorithms are linear in the network diameter rather than the network size.

Next, we discuss the tightness of the upper bounds in (5.59). Note that minimal polynomials q_i , thus κ_{\min} , depend explicitly on the weight matrix W . In fact, zeros of q_i are the eigenvalues of W corresponding to the modes of system (5.6) that are observable from the output x_i (see [141, Sec. V]). This has two direct implications as follows: (i) if the network (5.6) is observable from at least one node, then $\kappa_{\min} = N$ (a line graph falls into this case), (ii) an algorithm for computing q_i locally can be used as a means of verifying system observability or computing graph spectrum in a distributed manner. This also suggests that the observability theory might be useful in the problem of weight design so as to minimize κ_{\min} .

Finally, when all agents know the degree of their own minimal polynomial, they can compute κ_{\min} in a distributed fashion by using a max-consensus algorithm [139] for a finite number of time steps that is upper bounded by $\text{diam}(\mathcal{G})$.

5.5.2 Performance Limit of Distributed Subgradient Methods

We now discuss how fast the convergence of a distributed subgradient method could be in comparison with the corresponding centralized counterpart and with our algorithm and in connection with the network topology.

First, it is clear that for a general problem in the form of (5.1) and a given network topology, $\text{diam}(\mathcal{G})$ is equal to the smallest running time among all possible distributed algorithms since this is the minimum time for information to travel from any node to all others in the network. (This also explains the lower bound in (5.59).) An intuitively simple algorithm that theoretically achieves this fastest convergence

would be based on *communication flooding*; in particular, assuming sufficient communication power and memory capacity of each agent as well as availability of closed form characterization of each local cost function f_i and closed form solution to the global optimization problem, at each time step, every agent broadcasts its function f_i (assumed to have a unique identifier) and all those data received from neighbors at previous time step. As a result, at time $\text{diam}(\mathcal{G}) + 1$, all the agents are able to determine the global cost function F and, hence, can determine the optimum independently. Of course, even when assuming uniqueness of the optimal solution (so as for all the agents also reach consensus), this algorithm is far from being of any practical use. However, a similar behavior exhibits when applying our algorithms (5.40)-(5.42) and (5.53)-(5.54) for quadratic objective functions, where the closed form solution to the global problem is simple (and the centralized Newton method solves it in one iteration). Specifically, (5.40)-(5.42) terminates in κ steps with exact solution while (5.53)-(5.54) can do so with arbitrary small error by using a sufficiently small γ . As noted earlier, $\kappa = \text{diam}(\mathcal{G}) + 1$ in certain graphs.

Second, suppose that the global objective function can be optimized by some centralized subgradient-based method with some convergence time. In the distributed setting, one should expect that the corresponding distributed algorithm, when converging, is slowed down by at least a factor of $\text{diam}(\mathcal{G})$, again, due to the limit of information travel in discrete-time. As noted earlier in Remark 5.3.10, the best analysis on distributed subgradient methods for convex cost functions with bounded subgradients demonstrates $O(N/\epsilon^2)$ convergence time, which is linear in the network size (see [134]), while that of the centralized counterpart is $O(1/\epsilon^2)$.

Our result of $O(\kappa_{\min}/\epsilon^2)$ convergence time is the first to bridge the gap between $O(N/\epsilon^2)$ and $O(\text{diam}(\mathcal{G})/\epsilon^2)$, which we reckon it to be the limit of distributed sub-gradient methods. Of course, the tightness of these bounds depends on the network topology. For example, in a line graph, they are equivalent. Complete graphs are on the other extreme with $\kappa_{\min} = \text{diam}(\mathcal{G}) + 1 = 2$ for any network size. This makes sense since agents in a complete network should be able to act unanimously.

5.6 Simulations

Next we give some simulation results to illustrate the algorithms proposed above. In these examples, each agent does not know its minimal polynomial in advance, but rather computes it using Algorithm 4 in Chapter 4 in connection with M consensus iterations (5.6), provided that M is sufficiently large, e.g., $M \geq 2 \deg(q_i) + 1, \forall i \in \mathcal{V}$.

5.6.1 Example 1: Network of 5 agents with differentiable cost functions having Lipschitz continuous gradient

Consider the network and associated weight matrix shown in Figure 5.1.

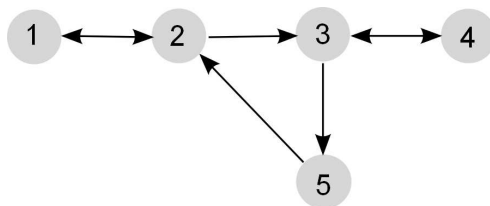


Figure 5.1: Network topology in example 1.

$$W = \begin{bmatrix} .7 & .3 & 0 & 0 & 0 \\ .2 & .6 & 0 & 0 & .2 \\ 0 & .3 & .4 & .3 & 0 \\ 0 & 0 & .5 & .5 & 0 \\ 0 & 0 & .4 & 0 & .6 \end{bmatrix}$$

Let $X = [-2, 2]$ and local cost functions be

$$\begin{aligned} f_1(x) &= (x - 3)^2 + 2x, & f_2(x) &= (x^4 - x^3 + 2x^2)/3, \\ f_3(x) &= (x - 0.1)^4/6, & f_4(x) &= e^x, \\ f_5(x) &= -2x^2 + 2x, & & \text{(nonconvex)}. \end{aligned}$$

Note that $F(x) = \sum_{i=1}^5 f_i(x)$ is convex, but f_5 is not. In fact, F is strongly convex with Lipschitz continuous gradient. Here, $x^* = 0.7427 \in X$ and $F^* = 9.4812$.

Here, we combine both extensions of the main algorithm as proposed in Section 5.3.2 to deal with global constraint X and row stochasticity of the weight matrix; in particular, (5.32) and (5.37) will be employed (see Remark 9). In order to apply these iterations, the agents need to find their corresponding element π_i in the normalized left Perron eigenvector $\boldsymbol{\pi}$ of the weight matrix. This can be done using the idea discussed at the end of Section 5.3.2, which we describe next.

Prior to implementation of the optimization algorithm, let all the agents run the following $2N + 1$ iterations (we assume that the network size N is available to the agents)

$$\mathbf{p}^{(i)}(t + 1) = \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{p}^{(j)}(t), \quad \forall i \in \mathcal{V}, t = 0, \dots, 2N, \quad (5.60)$$

where $\mathbf{p}^{(i)}(0) = \mathbf{e}_i \in \mathbb{R}^N$ (i.e., the i -th unit vector). At time $t = 2N + 1$, each agent i has enough data, namely the sequence $\{[\mathbf{p}^{(i)}]_i(t)\}_{t=0}^{2N}$, to compute $\mathbf{a}^{(i)}$ (or q_i equivalently) as well as π_i as shown in Section 5.2.3. In particular,

$$\begin{aligned}\boldsymbol{\pi} &= [0.20619, 0.30928, 0.20619, 0.12371, 0.15464]^T \\ \mathbf{a}^{(i)} &= [0.0192, -0.261, 1.1, -1.8, 1]^T, \quad \forall i \in \mathcal{V}.\end{aligned}$$

In this case, $\kappa_{\min} = N$. We will take $\kappa = N + 1$.

Next, to ensure convergence of the algorithm, the constant step size γ is chosen to satisfy conditions of Theorem 5.3.12, which requires a Lipschitz constant L of the global gradient ∇F . We now show that all the agents can determine such a constant in finite time and in a distributed manner, and then locally select a suitable step size. To this end, suppose that the agents know a Lipschitz constant \bar{l}_i of their local gradient ∇f_i . If f_i is continuously twice differentiable, such an $\bar{l}_i := \max_{x \in X} |\nabla^2 f_i(x)|$ can be found easily especially when X is compact (which holds in this example). Clearly, ∇F is also Lipschitz continuous on X with constants \bar{L} and \hat{L} given by

$$\bar{L} := \sum_{i \in \mathcal{V}} \bar{l}_i \leq N \max_{i \in \mathcal{V}} \bar{l}_i =: \hat{L}.$$

On the one hand, by using a max-consensus protocol [139], all the agents can determine $\max_{i \in \mathcal{V}} \bar{l}_i$ as well as \hat{L} (since it is assumed that N is known to every agent). However, \hat{L} could be a very loose bound, leading to a very small step size, thus reducing convergence speed. On the other hand, \bar{L} is usually a tighter bound, which can also be computed locally by using minimal polynomials obtained earlier as follows. After (5.60), let all the agents also perform the following update (where t

denotes the iteration index, not physical time)

$$l_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} l_j(t), \quad \forall i \in \mathcal{V}, \quad t = 0, \dots, N,$$

where $l_i(0) = \pi_i^{-1} \bar{l}_i$. Upon termination, each agent i can find

$$\left(\sum_{\tau=0}^{D_i} a_\tau^{(i)} l_i(\tau) \right) / \left(\sum_{\tau=0}^{D_i} a_\tau^{(i)} \right) \stackrel{(\text{Thm. 5.2.3})}{=} \sum_{i \in \mathcal{V}} \pi_i l_i(0) = \sum_{i \in \mathcal{V}} \bar{l}_i = \bar{L}.$$

Now the agents can locally compute the step size $\gamma = \varepsilon 2N / \bar{L}$, where a common $\varepsilon \in (0, 1)$ is known to the agents beforehand. Since \bar{L} is usually not the least Lipschitz constant, ε can be set to 1. (A small ε leads to a small step size, which can reduce the convergence speed.) In this example, $\{\bar{l}_i\} = \{2, 20, 8.9, 7.4, 4\}$ and we take $\gamma = 2N / \bar{L} = \frac{10}{42.3}$. Moreover, we let every agent i compute $F_k^{(i)} = \frac{1}{N} F(s_i(k\kappa))$ and use a relative tolerance $\epsilon = 10^{-6}$ (see Appendix A.3.9 and criterion (A.42)) to locally terminate the main algorithm.

The simulation results of the our algorithm are given in Fig. 5.2, with (randomly generated) initial conditions:

$$\mathbf{x}(0) = [0.6238, 1.4262, -0.9162, 1.5838, -1.1648]^T.$$

As expected, s_i 's (depicted by solid lines) become identical after the first κ steps, and then converge to the optimal solution x^* (shown by dash-line) while x_i 's reach limit cycles of period κ . Every agent locally decides to stop at $t = 186$ (i.e., $k = 31$) since each finds that $|F_{30}^{(i)} - F_{29}^{(i)}| = 7.5253 \times 10^{-7} |F_{30}^{(i)}|$ (noting that $F_{30}^{(i)}$ is computed at time $t = 31N$). Upon termination, $s_i(186) = 0.7402$ and $F(s_i(186)) = 9.4813$.

We also carry out the centralized subgradient method (5.3) in the form of (5.19) using the same step size $\gamma_k = \gamma$ and with the starting point $\bar{s}(\kappa) = \boldsymbol{\pi}^T \mathbf{x}(0)$.

The simulation result of $\bar{s}(k\kappa)$ is marked with \circ in the top-left sub-figure, which agrees with $s_i(k\kappa)$. We further compare the objective error of our algorithm with that obtained from the DPS method (5.5) with diminishing and non-summable step sizes of the form $\gamma(t) = \frac{a}{t^b}$ where $a = [0.01 : 0.05 : 0.5]$ and $b = [0.5 : 0.1 : 1]$. For this method, we denote $\bar{s}(t) = \frac{1}{N} \sum x_i(t)$. Results for a few samples of (a, b) are given in the right subplot of Fig. 5.2. Here we also scale gradients in DPS by a factor of $(N\pi_i)^{-1}$ just like the re-weighting technique in [93, 100] and also (5.37). Note that generally the DPS method is not guaranteed to converge when some f_i are nonconvex. Moreover, the DSM (5.4) fails to converge if $\gamma(t)$ is not selected carefully, e.g., $a > 0.5$ and $b = 0.5$. Clearly, our algorithm outperforms DPS in terms of the convergence rate and number of gradient evaluations.

With the same network and weight matrix as before, we now consider quadratic cost functions $f_i(x) = b_i(x - c_i)^2$, where $\mathbf{c} = [0, 4, 3, 1, 1]^T$ and $\mathbf{b} = [3, 1, 3, 3, 4]^T$. Fig. 5.3 shows the performance of Algorithm (5.53)-(5.54) with different values of step size γ . Clearly, sufficiently small γ 's yield near dead-beat responses.

5.6.2 Example 2: Network of 200 agents with ℓ_1 cost functions

Now we consider a set of $N = 200$ agents communicating over a ring graph, where $\mathcal{N}_i = \{i, i \pm 1, i \pm 10\}$, for $\forall i \in \mathcal{V}$ (if $i + 10 > 200$, by $i + 10$ we mean $i + 10 - 200$). Here, the graph is distance regular with diameter $\text{diam}(\mathcal{G}) = 10$, and each agent has 5 neighbors (including itself). Assume that $W = [w_{ij}]$ is such that $w_{ij} = \frac{1}{|\mathcal{N}_i|}$ if $j \in \mathcal{N}_i$ and $w_{ij} = 0$ otherwise. In this case, by Theorem 5.5.1 (with $\epsilon = 0.2$), we

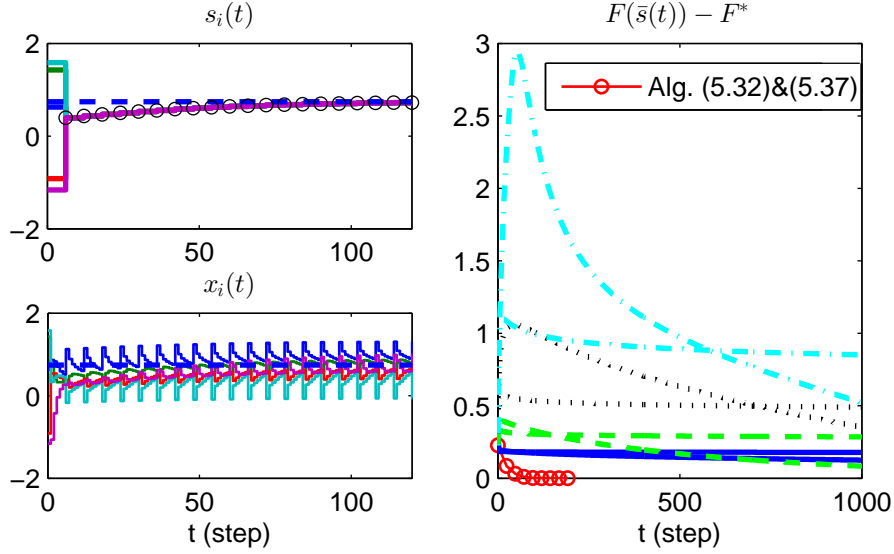


Figure 5.2: Network responses for example 1 with convex cost functions having Lipschitz continuous gradient using Algorithm (5.32) and (5.37). *Left:* For any $i \in \mathcal{V}$, $s_i(t)$ (solid lines) converges to optimal solution (dash-line) and $x_i(t)$ reaches a limit cycle of period κ . In the top-left figure, \circ represents $\bar{s}(k\kappa)$ of the centralized subgradient method implemented as (5.19). *Right:* Objective error comparisons with DPS using step size $\gamma(t) = \frac{a}{t^b}$, where (blue) solid lines correspond to $a = 0.01$, (green) dashed lines $a = 0.05$, (black) dotted lines $a = 0.1$, and (cyan) dash-dotted ones $a = 0.2$. For each a , we plot the results for $b = 0.5$ and 1 . The results from our algorithm are shown in red circles \circ . The algorithm terminates locally for all the agents at $t = 186$ with relative error of the global cost function guaranteed to be less than $\epsilon = 10^{-6}$.

have $\deg(q_i) = \text{diam}(\mathcal{G}) + 1 = 11, \forall i \in \mathcal{V}$.

The local cost functions are $f_i(x) = |x - c_i|$ with $x \in X = [0, 100]$ and $c_i = i$ if $i \leq 100$ and $c_i = 0.2(i - 100)$ otherwise. So, the goal of all the agents is to find the median value of $\{c_i\}_1^N$. Thus, in this case, $x^* = 16.9$. The choice of c_i is motivated by the desire to have x^* far from the mean μ_c of the elements in $\{c_i\}$ (which is $\mu_c = 30.3$ and can be found by any averaging protocol).

Here, each local cost function is convex but non-differentiable and has subgradients bounded by $L_0 = 1$. Therefore, we will use Algorithm (5.32)-(5.33), where each agent initializes $x_i(0) = c_i$. We also want to compare the performance of our

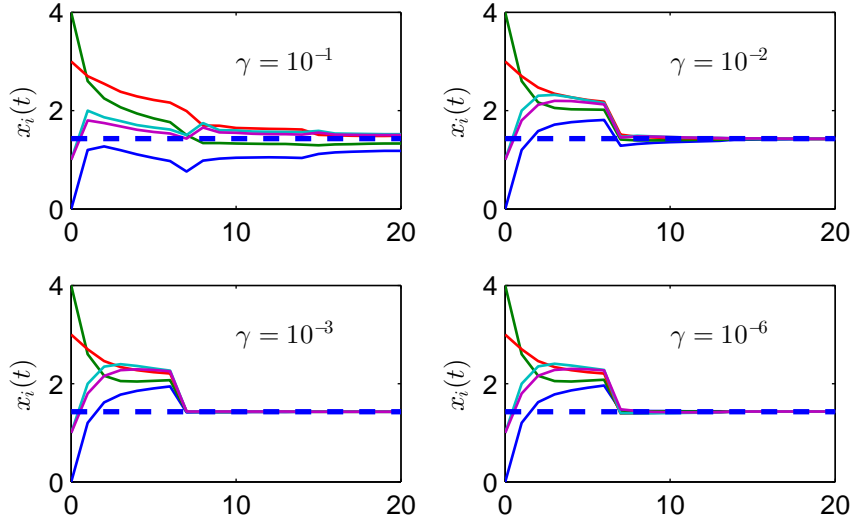


Figure 5.3: Network responses for example 1 with quadratic cost functions when using Algorithm 5.3 with $\kappa = 7$, $\mathbf{x}(0) = \mathbf{c}$, and with 4 values of γ .

algorithm with the one in [134]. Thus, we choose the step size by Theorem 5.3.5-(ii), i.e., $\gamma = \frac{R}{L_0} \sqrt{\frac{\kappa}{T}}$. In this example, we take the number of iterations $T = 4N$ (as suggested by [134]), $R = 100$ (which is the size of X),² and $L_i = L_0 = 1$. Since $\deg(q_i) = 11, \forall i \in \mathcal{V}$, κ needs to satisfy the condition $\kappa \geq 11$. We suppose that each agent sets $\kappa = 50$. Hence, $\gamma = 25$.

In this example, we let each agent i reevaluate $\mathbf{a}^{(i)}$ (equivalently its minimal polynomial) after every κ steps (note that $\kappa > 2 \deg(q_i)$ in this case). (The reason is that we observe from simulations that computation of minimal polynomials in large graphs is prone to error.) The results are given in Fig. 5.4. Algorithm (5.32)-(5.33) and the one in [134] do not converge asymptotically to the optimal solution but

²A better choice for R can be found as follows: at time κ , $\bar{s}(\kappa) = s_i(\kappa) = \sum_{i=1}^N x_i(0)/N = \mu_c$. Thus each agent can take $R = \max_{u \in X} |s_i(\kappa) - u| = 100 - s_i(\kappa) = 69.7$, which then improves the objective error bound in (5.22) by 30.3%.

rather approach a solution neighborhood since both use constant step sizes. Ours does so faster and the size of the neighborhood is much smaller (thanks to the rate $O(RL_{\max}\sqrt{\frac{\kappa}{T}})$ as compared to $O(R^2L_{\max}\sqrt{\frac{N}{T}})$ in [134]; see Remark 5.3.10). There are some small numerical errors in the simulation of $\mathbf{s}(t)$, but that does not cause instability to our algorithm. The DSM (5.4) with $\gamma(t) = \frac{1}{\sqrt{t}}$ admits asymptotic convergence but with very slow rate. Furthermore, it is not trivial to choose a “good” step size sequence or a stopping criterion with some performance guarantee (such as consensus and objective error bound); it is especially difficult doing so in a distributed fashion. In contrast, our algorithm allows efficient stopping criteria such as (5.26)-(5.27) (and (A.42) in Appendix A.3.9), or with a predetermined number of iterations. Moreover, consensus (of the optimization variables s_i) is always guaranteed upon termination.

5.7 Concluding Remarks

We have presented three fast algorithms for the distributed optimization problem (5.1) on a fixed and directed graph with convergence time being linear in the maximum degree of the agents’ minimal polynomials rather than the network size. From a broader view, our algorithms can be seen as a way of distributing the centralized subgradient method without sacrificing its convergence behavior, at the cost of the algorithm being slowed down due to a larger time-scale needed for diffusion of information through the network.

Among possible directions for future work, we mention the problem of design-

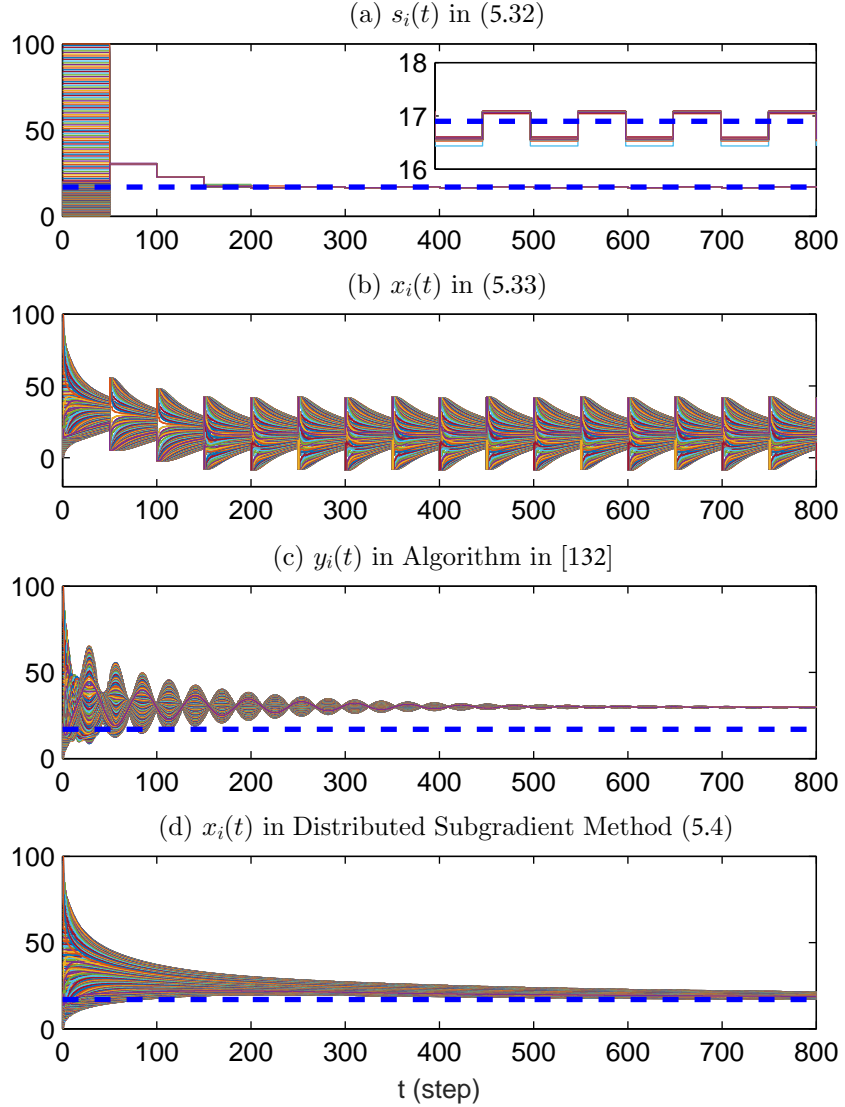


Figure 5.4: Responses of the network in example 2. Dashed line: optimal solution. (a)-(b): Algorithm (5.32)-(5.33), where sub-figure within (a) is a zoom-in of period $[400, 800]$; (c): Algorithm by Olshevsky (2016) with a constant step size $\beta = \frac{1}{L_0\sqrt{NT}}$; and (d): Distributed Subgradient Method (5.4) with $\gamma(t) = \frac{1}{\sqrt{t}}$.

ing the weight matrix W for a given network topology so as to achieve the smallest possible κ_{\min} . For the case of large networks, κ_{\min} could be large, and hence determining the exact minimal polynomial could be a challenging task for each agent

in terms of memory storage and computational capability. In this scenario, the use of approximations of minimal polynomials or other finite-time consensus protocol may be a problem deserving investigation. Another appealing direction is to adapt the algorithms to time-varying networks, as well as networks influenced by noises and/or delays.

Chapter 6: Distributed Optimization over Directed Graphs with Row Stochasticity and Constraint Regularity

Abstract: This chapter deals with an optimization problem over a network of agents, where the cost function is the sum of the individual (possibly nonsmooth) objectives of the agents and the global constraint set is the intersection of local constraints; this problem is more general than that in the previous chapter. The main goals of this chapter are: (i) to remove the need for column stochasticity; (ii) to relax the compactness assumption, and (iii) to provide a unified convergence analysis. Specifically, assuming the communication graph to be fixed and directed and the weight matrix to (only) be row stochastic, a distributed projected subgradient algorithm and a variation of the algorithm are presented to solve the problem for cost functions that are convex and Lipschitz continuous. The key component of the algorithms is to adjust the subgradient of each agent by an *estimate* of its corresponding entry of the normalized left Perron eigenvector of the weight matrix. These estimates are obtained locally from an augmented consensus iteration using the same row stochastic weight matrix and requiring very limited global information about the network. Moreover, based on a regularity assumption on the local constraint sets, a unified analysis is given that can be applied to both unconstrained

problems and constrained ones without assuming compactness of the constraint sets. Finally, the convergence rate of the algorithms is studied in terms of the distance from each agent’s available estimate to the global constraint set and an objective error defined on this set.

6.1 Introduction

As in Chapter 5, we consider a network of agents without a central coordination unit that is tasked with solving a global optimization problem in which the objective function is the sum of local costs of the agents, that is, $F(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x})$ where $f_i : \mathbb{R}^m \rightarrow \mathbb{R}$ represents the private objective of agent i and N is the number of agents in the network. In addition, each agent may be associated with a private constraint set. Many distributed optimization methods have been developed to address this problem; see e.g., [32, 39, 42, 43, 76–81, 83, 94, 134, 144, 145] and references therein.

Although much research has been carried out on this problem area, most of the existing literature invokes the assumption that communication among agents is bidirectional. That is, for any pair of neighboring agents, each agent receives information from the other. This assumption further allows many distributed algorithms to employ *doubly stochastic* weight matrices, allowing straightforward mechanisms for the agents to reach an optimal consensus. However, the double stochasticity assumption is not always practical in real world applications. This is the case, for example, where agents have different communication ranges due to environmental effects or individual broadcast power limits.

In this work, we consider a more general case in which the communication among agents is not necessarily bidirectional, and thus is naturally represented by a directed graph. This scenario has recently been considered in [82, 93, 145]. A common idea in these chapters is the combination of a (sub)gradient distributed optimization algorithm and the Push-Sum protocol [146]. One essential requirement of this protocol is that each agent knows its out-degree exactly and/or controls its outgoing weights so that they sum up to one, leading to a *column stochastic* weight matrix. The same requirement is also imposed in [147], where the authors develop a distributed subgradient algorithm by employing a weight balancing technique. Such requirement, however, could be impractical in many other situations, especially when agents use a broadcast-based communication scheme and thus they neither know their out-neighbors nor are able to adjust their outgoing weights (i.e., the weights that others put on its information). In wireless sensor networks, for instance, directed communications can arise as a consequence of geometric network layout or nonuniform power limits, each node can only send information to those lying within its coverage area without receiving acknowledgment signals from them. A similar scenario may also happen during operation of a network initially designed to implement a column stochastic weight matrix; for example, a node encounters an unreliable or broken in-coming channel and has no “cheap” and local means to inform the sender of it not receiving packages (i.e., logically not being an out-neighbor of the sender). As a result, the performance of the network as a whole is not guaranteed due to this malfunctioning communication link, even if the network is still strongly connected. Thus, networks relying on a column stochastic weight

matrix may not be robust to link failure.

In comparison with a column stochastic weight matrix, one that is *row stochastic* is much easier to achieve in a distributed setting. Here, each agent can individually (and to some extent arbitrarily) decide the weights placed on information it receives from its neighbors. Thus, if the weight matrix is required to be only for a row stochastic, there is no need for nodes to send acknowledgment signals. As an immediate but important consequence, a network requiring only a row stochastic weight matrix is more robust to link losses/jamming, and even changes in the network structure. This makes row stochastic matrices suitable for reaching consensus in broadcast-based communication environments, for example *ad hoc* wireless networks. However, when a row stochastic matrix is used for distributed optimization, most (if not all) (sub)gradient based algorithms fail to achieve an optimal solution due to the nonuniform stationary distribution of the weight matrix (also known as the normalized left Perron eigenvector). In [93], the authors suggest a re-weighting technique that makes it possible to use a row stochastic matrix in distributed optimization. The same technique is also employed in [94]. However, the implementation of the algorithms in [93, 94] assume knowledge of the graph, that is the stationary distribution of the weight matrix and the number of agents in the network. Indeed, a fully distributed algorithm employing only row stochastic weight matrices has not been available in the field of distributed optimization thus far.

In this work, we achieve such algorithms under mild requirements on available global network information and under the assumption that the network is strongly connected. More precisely, we present a distributed algorithm and a variation on the

algorithm that use a row stochastic weight matrix and assume that each agent knows only an upper bound on the number of agents in the network. Our idea is as follows. We let all the agents perform an augmented consensus protocol in order to estimate the stationary distribution of the weight matrix while updating their states using an iteration akin to that in the Distributed Projected Subgradient (DPS) method (see, e.g., [137, 148, 149]), except that subgradient values are now scaled appropriately and locally by the agents. Here, the estimation step is implemented concurrently with the optimization step, and thus no network communication overhead is added. Moreover, although the algorithm is based on the projected subgradient method, we believe that its principle (i.e., the use of a particular augmented consensus) can be generalized to a class of distributed algorithms that use consensus and subgradient steps.

Another important contribution is our unified convergence analysis (as well as the convergence rate) that applies to both unconstrained and constrained problems with identical or nonidentical private constraint sets. Most existing works on subgradient based methods usually assume the problem to be either unconstrained [78, 134, 145], or constrained with identical (often compact) constraint sets [81, 93, 137, 148, 150]. Nonidentical constraints are considered in [94, 137, 151], where the local constraint sets are assumed to be compact and their intersection has a nonempty interior. In our work, we assume regularity of the constraint sets, which is weaker than requiring boundedness and allows the global constraint set to have an empty interior. We establish convergence of our algorithms to the optimal solution, and demonstrate how the rate of convergence depends on the step size sequence,

exhibiting similarity to that of the centralized subgradient approach. To the best of our knowledge, convergence rates of distributed subgradient methods have not been studied before for the case of nonidentical unbounded constraint sets (possibly with an empty-interior intersection).

Preliminary work along the line of this chapter appeared in [150], where only one algorithm was presented and several proofs were omitted. In addition, it is assumed in [150] that all the local constraint sets are identical and compact, while in this chapter we consider nonidentical constraint sets and relax the compactness requirement, allowing for a broader class of applications. The current chapter further introduces a variation on the algorithm presented in [150], and presents a new convergence analysis that holds for both algorithms under these relaxations. Here the proof technique relies on the regularity assumption on the local constraint set, thus significantly different from that in [150]. Finally, the rate of convergence, which was not shown in [150], is studied here for both algorithms.

The rest of the chapter proceeds as follows. The problem formulation and proposed algorithms are given in Section 6.2. The convergence and the convergence rate of the algorithms are studied in Sections 6.3 and 6.4, respectively. Section 6.5 includes a numerical example to illustrate our findings. Concluding remarks are given in Section 6.6.

Additional Notation and Terminology:

The projection of a vector \mathbf{x} on a nonempty closed convex set $X \subseteq \mathbb{R}^m$ is denoted by $P_X(\mathbf{x})$, i.e., $P_X(\mathbf{x}) = \arg \min_{\mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|$, where as usual, $\|\cdot\|$ denotes the 2-norm. We also denote by $\text{dist}(\mathbf{x}, X)$ the (Euclidean) distance from \mathbf{x} to X , i.e.,

$\text{dist}(\mathbf{x}, X) = \|\mathbf{x} - P_X(\mathbf{x})\|$. The following inequality is called the *nonexpansiveness property*:

$$\|P_X(\mathbf{x}) - P_X(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^m \quad (6.1)$$

We will employ the notion of regularity of the constraint sets, which plays an important role in the study of projection algorithms. This notion involves upper estimating the distance of a point to the intersection of a collection of closed convex sets in terms of the distance to each set (see [152,153]). Recalled next is the definition needed here, stated for a finite dimensional setting.

Definition 6.1.1. A collection of closed convex sets $\{X_i, i \in \mathcal{V}\}$ (with a nonempty intersection) is *regular* with respect to a nonempty set $B \subseteq \mathbb{R}^m$ if there exists a constant $r_B \geq 1$ such that

$$\text{dist}(\mathbf{x}, \cap_{i \in \mathcal{V}} X_i) \leq r_B \max_{i \in \mathcal{V}} \text{dist}(\mathbf{x}, X_i), \quad \forall \mathbf{x} \in B. \quad (6.2)$$

It is said to be regular if $B = \mathbb{R}^m$.

For example, if the sets X_i are identical, then they are regular. The following two sets $X_1 = \{(x_1, x_2) : x_2 \geq x_1^2\}$ and $X_2 = \{(x_1, x_2) : x_2 \leq 0\}$ with $X_1 \cap X_2 = \{\mathbf{0}\}$ are not regular with respect to any ball centered at the origin.

6.2 Problem Formulation and Proposed Algorithms

Consider a network consisting of N agents where the underlying communication is characterized by a fixed directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. All agents share the objective

of solving

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^m} \quad & F(\mathbf{x}) := \sum_{i \in \mathcal{V}} f_i(\mathbf{x}), \\ \text{s.t.} \quad & \mathbf{x} \in \bigcap_{i \in \mathcal{V}} X_i =: X \end{aligned} \tag{6.3}$$

where each $f_i : \mathbb{R}^m \rightarrow \mathbb{R}$ is a convex function representing the private objective of agent i , and each X_i is a convex constraint set only available to agent i . Obviously, F is also convex. Let F^* and X^* denote the optimal value and the optimal solution set of the problem (i.e., $X^* = \{\mathbf{x} \in X, F(\mathbf{x}) = F^*\}$). Let

$$U := \text{conv}\left(\bigcup_{i \in \mathcal{V}} X_i\right), \tag{6.4}$$

i.e., the convex hull of $\bigcup_{i \in \mathcal{V}} X_i$. The following assumptions are adopted in the sequel.

Assumption 6.2.1. (Basic Problem Assumptions) *Problem (6.3) satisfies the following:*

(a) (Constraint sets) *The sets $X_i \subseteq \mathbb{R}^m$ are closed and convex, and $X \neq \emptyset$.*

Moreover, $\{X_i, i \in \mathcal{V}\}$ is regular with respect to U .

(b) (Bounded subdifferential) *For any $i \in \mathcal{V}$, $f_i : \mathbb{R}^m \rightarrow \mathbb{R}$ is convex with subdifferential bounded on U , i.e.,*

$$\exists L_f \in (0, \infty), \|\mathbf{g}_i\| \leq L_f, \forall \mathbf{g}_i \in \partial f_i(\mathbf{x}), \forall \mathbf{x} \in U \tag{6.5}$$

(c) *The solution set X^* is nonempty.*

Here, the regularity assumption on the collection of constraint sets is weaker than requiring boundedness, which allows us to consider a broader class of optimization problems. This assumption holds trivially with $r_U = 1$ when the constraint sets

are identical. An unconstrained problem is a special case with $X_i = \mathbb{R}^m, \forall i \in \mathcal{V}$. The regularity assumption is also satisfied if the sets X_i are compact and X has a nonempty interior; such assumptions are used in [94, 137]. In fact, by [152, Cor. 2], one can deduce that $r_U = \sum_{i=1}^N (D_U/\delta)^i$ is a regularity constant, where D_U denotes the diameter of U and δ is the radius of a ball lying in U . Other important cases include when the sets X_i are hyperplanes or half-spaces (see [153]).

Note also that since each f_i is convex on \mathbb{R}^m , Assumption 6.2.1(b) implies that each individual cost function f_i is also L_f -Lipschitz continuous on U . This will be the case if all X_i are compact, since then U is also compact. Assumption 6.2.1(c) can be satisfied when, e.g., the sets X_i are closed and at least one them is compact, since then X is compact. In general, however, we do not require compactness of the constraint sets.

In our setting, agent i only has access to f_i and local information on its neighbors' opinions, and no central coordinating node is assumed to exist. Thus, the agents need to collaborate in a distributed manner to solve problem (6.3). This involves local iterative computation along with information diffusion. We are interested in the scenario where the communication graph \mathcal{G} connecting the agents is directed and fixed. We make the following additional blanket assumptions.

Assumption 6.2.2. (Connectivity) *The network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is strongly connected.*

Assumption 6.2.3. (Unique ID) *The agents are labeled $1, 2, \dots, N$ and their messages carry a unique identifier of the sender. Moreover, all the agents know the value N (or an upper bound $N' \geq N$).*

Assumption 6.2.3 is only technical, implying that each agent can distinguish messages from its neighbors. This will be the case if media access control (MAC) addresses are used.

Here, at any time slot, each agent exchanges its current state with its neighbors (in accordance with the directed network structure). Upon receiving the information from its neighbors (including itself), agent i incorporates knowledge of these states using a weighted average scheme. Thus, each edge $(i, j) \in \mathcal{E}$ is associated with a weight $w_{ij} \geq 0$ (locally chosen by agent i). Let the weight matrix $W = [w_{ij}]$ satisfy the following condition.

Assumption 6.2.4. (Weight Rule) *The matrix W satisfies $w_i > 0$ for $i \in \mathcal{V}$, $w_{ij} > 0$ for $(i, j) \in \mathcal{E}$ and $w_{ij} = 0$ otherwise. Moreover, W is row stochastic.*

This assumption means that the zero-nonzero structure of the weight matrix W reflects the network structure. Note also that W has positive diagonal elements reflecting that each agent has access to its own state. Further, W is irreducible under Assumption 6.2.2.

Again we stress that unlike the case with existing algorithms in the literature, the weight matrix W is assumed to be only row stochastic, and not either doubly stochastic or column stochastic. As a result, each agent i controls the i -th row of W , independently with others. This also gives each agent the freedom in deciding the weights that it places on its neighbors' information. This explains why row stochastic matrices are more suitable for *ad hoc* wireless networks.

We now propose the following distributed algorithm to solve problem (6.3)

under all the assumptions above.

Algorithm 6.1. *At time $t = 0$, agent i initializes an estimate $\mathbf{x}_i(0) \in X_i$ and a variable $\mathbf{z}_i(0) = \mathbf{e}_i \in \mathbb{R}^N$ (or $\in \mathbb{R}^{N'}$ if only a bound $N' \geq N$ is available). For each time $t \geq 0$, all agents update their states as follows:*

$$\mathbf{x}_i(t+1) = P_{X_i} \left(\sum_{j \in \mathcal{V}} w_{ij} \mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} \right) \quad (6.6)$$

$$\mathbf{z}_i(t+1) = \sum_{j \in \mathcal{V}} w_{ij} \mathbf{z}_j(t). \quad (6.7)$$

Here, \mathcal{N}_i is the set of node i 's in-neighbors (including itself), $\gamma(t)$ is a nonnegative step size (which will be specified later), $\mathbf{g}_i(t) \in \partial f_i(\sum_{j \in \mathcal{V}} w_{ij} \mathbf{x}_j(t))$ is a subgradient of f_i , and $\mathbf{z}_i(t) = [z_{i1}, z_{i2}, \dots, z_{iN}]^T$ for each $i \in \mathcal{V}$.

Note that because of $w_{ii} > 0, \forall i \in \mathcal{V}$ (cf. Assumption 6.2.4), (6.7), and $z_{ii}(0) = 1$, it can be shown (later in Lemma 6.3.3) that $z_{ii}(t) > 0, \forall t \geq 0, \forall i \in \mathcal{V}$. Thus (6.6) is well defined.

In essence, the update (6.6) can be viewed as a modified version of the distributed projected subgradient (DPS) method [137] where each private cost function's subgradient is scaled by $z_{ii}(t)$ obtained from (6.7). Here, the update (6.7) is, in fact, a consensus iteration aiming to provide each agent $i \in \mathcal{V}$ with an estimate of $\boldsymbol{\pi} = [\pi_1, \dots, \pi_N]^T$ - the left normalized Perron eigenvector of W , i.e., the left eigenvector $\boldsymbol{\pi}$ satisfying $\mathbf{1}^T \boldsymbol{\pi} = 1$. This iteration resembles those used in [154, 155]. Of course, if each agent $i \in \mathcal{V}$ knows the π_i in advance, then iteration (6.7) is redundant as all the agents can simply use $z_{ii}(t) = \pi_i, \forall t \geq 0$. (In fact, if initialized with $\mathbf{z}_i(0) = \boldsymbol{\pi}$, then it follows from (6.7) that $\mathbf{z}_i(t) = \boldsymbol{\pi}$ for all $t \geq 0$.) In this case, our

rescaling subgradient technique reduces to the reweighting scheme used in [93, 94].

We also remark that the DPS method in [137] can be applied to time-varying networks but requires the weight matrix to be doubly stochastic at each time t . Further, for nonidentical constraint sets, [137] only considers complete graphs and assumes that the intersection set X has nonempty interior. Later, [94] extended the method to directed time-varying graphs possibly with (fixed and uniform) communication delays but still requires doubly stochastic weight matrices and compact constraint sets with nonempty interior. Thus, the results in these works are not readily applicable to cases where the X_i are unbounded (e.g., $X = \mathbb{R}^m$) and/or X has an empty interior (e.g., an X_i includes linear equality constraints) and the weight matrix is only row stochastic. Another extension in [145] dealing with the unconstrained case employs column stochastic matrices. Algorithm 6.1 can be seen as an extension of DPS under the fixed network setting where only row stochastic weight matrices are used. Note also that here we assume the network is fixed during one run of the algorithm. Between any two consecutive runs, the network structure is allowed to change, and our algorithm need not be adjusted except each agent i may need to reselect new weights w_{ij} for its new neighbor set (which is a trivial task). Moreover, our development technique does not employ the compactness of the constraint sets as well as nonempty interior of their intersection.

The following variation on Algorithm 6.1 will also be considered, where each agent takes the subgradient step first, followed by the consensus step:

Algorithm 6.2. *With the same initializations as in Algorithm 6.1, all agents update*

their states according to

$$\mathbf{x}_i(t+1) = P_{X_i} \left(\sum_{j \in \mathcal{V}} w_{ij} \left(\mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right) \right) \quad (6.8)$$

$$\mathbf{z}_i(t+1) = \sum_{j \in \mathcal{V}} w_{ij} \mathbf{z}_j(t), \quad (6.9)$$

where $\mathbf{g}_j(t) \in \partial f_j(\mathbf{x}_j(t))$, i.e., a subgradient of f_j at $\mathbf{x}_j(t)$ (which differs from the subgradient used in (6.6) of Algorithm 6.1).

It has been shown in [76] that the order of the optimization step and the consensus step in the original DPS method can be interchanged, which, if a constant step size is used, often gives a better convergence speed to a solution neighborhood [156]. Comparison between Algorithms 6.1 and 6.2, however, is out of the scope of this chapter.

In this work, the following type of diminishing step size sequence will be used to ensure convergence of our algorithms to the optimal solution. For the convergence rate analysis, a less restrictive assumption will be employed.

Assumption 6.2.5. (Step Size Rule) *The step size sequence $\{\gamma(t)\}$ is positive non-increasing and satisfies $\sum_{t=0}^{\infty} \gamma(t) = \infty$ and $\sum_{t=0}^{\infty} \gamma^2(t) < \infty$.*

There are many ways to choose the step size sequence $\gamma(t)$ satisfying this assumption, e.g., $\gamma(t) = \frac{c}{t^\theta}$, $\forall t \geq 1$, for some constants $c > 0$ and $\theta \in (0.5, 1]$.

6.3 Basic Relations and Convergence Result

In this section, we simultaneously prove the convergence of both algorithms (6.6)-(6.7) and (6.8)-(6.9). We begin with a few basic results that will be used later.

First, besides the nonexpansiveness property (6.1), other properties of a projection operator are given in the following lemma.

Lemma 6.3.1. (*[137]*) *Let $Y \subseteq \mathbb{R}^m$ be a nonempty closed convex set. Then for any $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in Y$,*

$$(a) \quad (P_Y(\mathbf{x}) - \mathbf{x})^T(\mathbf{x} - \mathbf{y}) \leq -\|P_Y(\mathbf{x}) - \mathbf{x}\|^2.$$

$$(b) \quad \|P_Y(\mathbf{x}) - \mathbf{y}\|^2 \leq \|\mathbf{x} - \mathbf{y}\|^2 - \|P_Y(\mathbf{x}) - \mathbf{x}\|^2.$$

Second, the following lemma is a consequence of the convexity of the function $\|\cdot\|^2$.

Lemma 6.3.2. *For any $a_1, \dots, a_N \geq 0$ such that $\sum_{i=1}^N a_i = 1$, we have $\|\sum_{i=1}^N a_i \mathbf{x}_i\|^2 \leq \sum_{i=1}^N a_i \|\mathbf{x}_i\|^2$ for $\forall \mathbf{x}_i \in \mathbb{R}^m, i = 1, \dots, N$.*

Next, we characterize the convergence of the power iteration of the weight matrix in the following lemma, which is a consequence of the Perron-Frobenius theorem (see, e.g., [95]).

Lemma 6.3.3. (Convergence of power of weight matrix) *Let Assumptions 6.2.2 (Connectivity) and 6.2.4 (Weight Rule) hold. Then $\lim_{t \rightarrow \infty} W^t = \mathbf{1}\boldsymbol{\pi}^T$, where $\boldsymbol{\pi} > 0$ is the normalized left Perron eigenvector of W . Moreover, the convergence is geometric with rate $\lambda \in (|\lambda_2(W)|, 1)$, where $\lambda_2(W)$ is the second largest eigenvalue of W .*

Proof. Under Assumptions 6.2.2 and 6.2.4, W is an irreducible row stochastic matrix with positive diagonal entries, and thus primitive (i.e., W is irreducible and has only

one eigenvalue of maximum modulus; see, e.g., [95, Thm. 8.5.2 and Lem. 8.5.5]).

The result now follows from [95, Thm. 8.5.1]. \square

The next proposition, describing the convergence of the estimation step in (6.7), follows directly from the foregoing lemma and will be used in the sequel.

Proposition 6.3.4. (Convergence of z_{ii}) *Consider iteration (6.7). Let Assumptions (Connectivity) and 6.2.4 (Weight Rule) hold. Then for each $\lambda \in (|\lambda_2(W)|, 1)$, there exists $C=C(\lambda, W) > 0$ such that the following hold for $\forall i, j \in \mathcal{V}$ and $\forall t \geq 0$:*

$$|[W^t]_{ji} - \pi_i| \leq C\lambda^t, \quad |z_{ii}(t) - \pi_i| \leq C\lambda^t. \quad (6.10)$$

Moreover, there exists $\eta > 0$ such that

$$\eta^{-1} \leq z_{ii}(t) \leq 1, \quad \forall t \geq 0, \forall i \in \mathcal{V}. \quad (6.11)$$

Proof. Let $Z(t) = [\mathbf{z}_1(t), \mathbf{z}_2(t), \dots, \mathbf{z}_N(t)]^T$. It follows from Algorithm 6.1 that for any $t \geq 0$,

$$Z(t+1) = WZ(t), \quad Z(0) = I.$$

Thus, $Z(t) = W^t, \forall t \geq 0$. Hence, (6.10) follows by Lemma 6.3.3 for some $C > 0$ and $\lambda \in (|\lambda_2(W)|, 1)$.

Next, for each $i \in \mathcal{V}$, by (6.7), we have $z_{ii}(t+1) = \sum_{j \in \mathcal{V}} w_{ij} z_{ji}(t)$, where $z_{ii}(0) = 1, z_{ji}(0) = 0, \forall j \neq i$. Clearly, $1 \geq z_{ij}(t) \geq 0, \forall i, j \in \mathcal{V}, \forall t \geq 0$. Since $\lim_{t \rightarrow \infty} z_{ii}(t) = \pi_i > 0$, there exists $t_0 \geq 0$ such that $z_{ii}(t) \geq \pi_i/2, \forall i \in \mathcal{V}, \forall t > t_0$. Moreover, we have that $z_{ii}(t_0) \geq w_{ii} z_{ii}(t_0 - 1) \geq \dots \geq w_{ii}^{t_0} z_{ii}(0) > 0$ since $w_{ii} > 0$ (cf. Assumption 6.2.4). Therefore, $z_{ii}(t) > 0$ for any $t \in [0, t_0]$. By taking

$$\eta^{-1} = \min\{z_{ii}(t), \pi_i/2, \forall i \in \mathcal{V}, \forall t \in [0, t_0]\},$$

then (6.11) follows as desired. \square

Remark 6.3.5. In the rest of the chapter, the parameters C , λ and η refer to the constants in Proposition 6.3.4.

We now turn to iterations (6.6) and (6.8). Our next result describes a general relation on the overall evolution of the states of the agents in terms of their distances from any point $\mathbf{v} \in X$ as well as the weighted averaged state vector $\bar{\mathbf{x}}(t)$, defined as

$$\bar{\mathbf{x}}(t) := \sum_{j \in \mathcal{V}} \pi_j \mathbf{x}_j(t), \quad \forall t \geq 0. \quad (6.12)$$

This relation also involves the step size sequence $\gamma(t)$ and an error term ($F(\bar{\mathbf{x}}(t)) - F(\mathbf{v})$), which in general is not the global objective error since $\bar{\mathbf{x}}(t)$ may not be in X ; it is so if the constraint sets $\{X_i, i \in \mathcal{V}\}$ are identical.

Theorem 6.3.6. (Bound on evolution of \mathbf{x}_i) *Let Assumptions 6.2.1 (Basic Problem Assumptions), 6.2.2 (Connectivity), 6.2.3 (Unique ID) and 6.2.4 (Weight Rule) be satisfied. Then for both Algorithms 6.1 and 6.2, the following holds for any $\mathbf{v} \in X$ and $t \geq 0$:*

$$\begin{aligned} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq (1 + D_1 \lambda^{2t}) \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 \\ &\quad - 2\gamma(t) (F(\bar{\mathbf{x}}(t)) - F(\mathbf{v})) - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2 \\ &\quad + D_2 \gamma(t) \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| + D_3 \gamma^2(t), \end{aligned} \quad (6.13)$$

where $D_1 = NCL_f\eta$, $D_2 = 2L_f\eta$, $D_3 = L_f^2\eta^2 + NL_fC\eta$, and

$$\phi_i(t) := P_{X_i} \left(\sum_{j \in \mathcal{V}} w_{ij} \mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} \right) - \left(\sum_{j \in \mathcal{V}} w_{ij} \mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} \right) \quad (6.14)$$

for Algorithm 6.1 whereas for Algorithm 6.2, $\phi_i(t)$ is defined as

$$\phi_i(t) := P_{X_i} \left(\sum_{j \in \mathcal{V}} w_{ij} \left(\mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right) \right) - \sum_{j \in \mathcal{V}} w_{ij} \left(\mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right) \quad (6.15)$$

Proof. We provide here a proof for the case of Algorithm 6.1. The proof for Algorithm 6.2 is given in Appendix A.4.1. Let $\mathbf{y}_i(t) := \sum_{j \in \mathcal{V}} w_{ij} \mathbf{x}_j(t)$. By using (6.6) and the definition of $\phi_i(t)$ (cf. (6.14)), we have for any $\mathbf{v} \in X \subseteq X_i$

$$\begin{aligned} \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &= \left\| \mathbf{y}_i(t) - \mathbf{v} - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} + \phi_i(t) \right\|^2 \\ &= \left\| \mathbf{y}_i(t) - \mathbf{v} - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} \right\|^2 + \|\phi_i(t)\|^2 + 2\phi_i(t)^T \left(\mathbf{y}_i(t) - \mathbf{v} - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} \right) \\ &\leq \left\| \mathbf{y}_i(t) - \mathbf{v} - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} \right\|^2 - \|\phi_i(t)\|^2, \end{aligned} \quad (6.16)$$

where the last inequality follows from the fact that (cf. Lemma 6.3.1(a))

$$\phi_i(t)^T \left(\mathbf{y}_i(t) - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} - \mathbf{v} \right) \leq -\|\phi_i(t)\|^2.$$

The first term on the right side of (6.16) equals

$$\|\mathbf{y}_i(t) - \mathbf{v}\|^2 + \frac{2\gamma(t)}{z_{ii}(t)} \mathbf{g}_i(t)^T (\mathbf{v} - \mathbf{y}_i(t)) + \frac{\gamma^2(t)}{z_{ii}^2(t)} \|\mathbf{g}_i(t)\|^2. \quad (6.17)$$

We now derive an upper bound for each term in this sum. Rewriting $\mathbf{y}_i(t) - \mathbf{v} = \sum_{j \in \mathcal{V}} w_{ij} (\mathbf{x}_j(t) - \mathbf{v})$ then using Lemma 6.3.2 yields

$$\|\mathbf{y}_i(t) - \mathbf{v}\|^2 \leq \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \mathbf{v}\|^2. \quad (6.18)$$

Next, ignoring the positive factor $\frac{2\gamma(t)}{z_{ii}(t)}$, the second term in (6.17) can be bounded as follows:

$$\begin{aligned} \mathbf{g}_i(t)^T (\mathbf{v} - \mathbf{y}_i(t)) &\leq f_i(\mathbf{v}) - f_i(\mathbf{y}_i(t)) \leq f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t)) + |f_i(\mathbf{y}_i(t)) - f_i(\bar{\mathbf{x}}(t))| \\ &\leq f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t)) + L_f \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \bar{\mathbf{x}}(t)\|. \end{aligned} \quad (6.19)$$

where the first inequality holds since $\mathbf{g}_i(t) \in \partial f_i(\mathbf{y}_i(t))$, the second follows from the triangle inequality, and the last one from L_f -Lipschitz continuity of f_i over $\text{conv}(\bigcup_{i \in \mathcal{V}} X_i)$ (cf. Assumption 6.2.1(b)) and the triangle inequality. By continuing (6.16) and using (6.17), (6.18), (6.19) and the conditions that $\|\mathbf{g}_i(t)\| \leq L_f$ and $z_{ii}^{-1}(t) \leq \eta, \forall i \in \mathcal{V}, \forall t \geq 0$, we have

$$\begin{aligned} \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \mathbf{v}\|^2 + \frac{2\gamma(t)}{z_{ii}(t)} (f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t))) - \|\phi_i(t)\|^2 \\ &\quad + 2L_f \frac{\gamma(t)}{z_{ii}(t)} \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \bar{\mathbf{x}}(t)\| + \gamma^2(t) L_f^2 \eta^2. \end{aligned} \quad (6.20)$$

Multiplying both sides by π_i then summing over $i \in \mathcal{V}$ yields

$$\begin{aligned} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq \sum_{i \in \mathcal{V}} \pi_i \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \mathbf{v}\|^2 \\ &\quad + 2 \sum_{i \in \mathcal{V}} \frac{\pi_i \gamma(t)}{z_{ii}(t)} (f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t))) - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2 \\ &\quad + 2L_f \sum_{i \in \mathcal{V}} \frac{\pi_i \gamma(t)}{z_{ii}(t)} \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \bar{\mathbf{x}}(t)\| + \gamma^2(t) L_f^2 \eta^2. \end{aligned} \quad (6.21)$$

Now consider each term on the right side of (6.21). First,

$$\sum_{i \in \mathcal{V}} \pi_i \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \mathbf{v}\|^2 = \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2, \quad (6.22)$$

where we have used the fact that $\boldsymbol{\pi}^T W = \boldsymbol{\pi}^T$. Second,

$$\begin{aligned} \sum_{i \in \mathcal{V}} \frac{\pi_i}{z_{ii}(t)} (f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t))) &= \sum_{i \in \mathcal{V}} f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t)) + \sum_{i \in \mathcal{V}} \left(\frac{\pi_i}{z_{ii}(t)} - 1 \right) (f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t))) \\ &\leq F(\mathbf{v}) - F(\bar{\mathbf{x}}(t)) + \sum_{i \in \mathcal{V}} \frac{|z_{ii}(t) - \pi_i|}{z_{ii}(t)} |f_i(\bar{\mathbf{x}}(t)) - f_i(\mathbf{v})| \\ &\leq F(\mathbf{v}) - F(\bar{\mathbf{x}}(t)) + NCL_f \eta \lambda^t \|\bar{\mathbf{x}}(t) - \mathbf{v}\|, \end{aligned} \quad (6.23)$$

where $C > 0$ and $\lambda \in (0, 1)$ satisfy (6.10), and η satisfies (6.11). Third, we also have

$$\begin{aligned} \sum_{i \in \mathcal{V}} \frac{\pi_i}{z_{ii}(t)} \sum_{j \in \mathcal{V}} w_{ij} \|\mathbf{x}_j(t) - \bar{\mathbf{x}}(t)\| &\leq \eta \sum_{i, j \in \mathcal{V}} \pi_i w_{ij} \|\mathbf{x}_j(t) - \bar{\mathbf{x}}(t)\| \\ &= \eta \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| \end{aligned} \quad (6.24)$$

Now, combining (6.21)-(6.24) yields

$$\begin{aligned} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 - 2\gamma(t) (F(\bar{\mathbf{x}}(t)) - F^*) \\ &\quad - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2 + 2\gamma(t) NCL_f \eta \lambda^t \|\bar{\mathbf{x}}(t) - \mathbf{v}\| \\ &\quad + 2\gamma(t) L_f \eta \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| + \gamma^2(t) L_f^2 \eta^2. \end{aligned} \quad (6.25)$$

Finally, by writing $\bar{\mathbf{x}}(t) - \mathbf{v} = \sum_{i \in \mathcal{V}} \pi_i (\mathbf{x}_i(t) - \mathbf{v})$ and then using the Cauchy-Schwarz inequality and Lemma 6.3.2, we have

$$\begin{aligned} 2\gamma(t) \lambda^t \|\bar{\mathbf{x}}(t) - \mathbf{v}\| &\leq \gamma^2(t) + \lambda^{2t} \left\| \sum_{i \in \mathcal{V}} \pi_i (\mathbf{x}_i(t) - \mathbf{v}) \right\|^2 \\ &\leq \gamma^2(t) + \lambda^{2t} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 \end{aligned}$$

Using this bound for (6.25) and then rearranging terms yields (6.13) as desired. \square

Before proceeding further, it is worth highlighting the differences between this result, in particular (6.13), with that obtained from the usual DPS method [137] in the context of Algorithm 6.1. First, since the (normalized) left Perron eigenvector $\boldsymbol{\pi}$ is nonuniform, we opt for employing the weighted average vectors $\bar{\mathbf{x}}(t)$ (as well as $\sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2$) instead of the exact average one. Of course, when $\boldsymbol{\pi} = \mathbf{1}/N$, i.e., W is doubly stochastic, the former vector reduces to the latter. Second, the term $D_1 \lambda^{2t} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2$ (or more precisely the term $2\gamma(t) NCL_f \eta \lambda^t \|\bar{\mathbf{x}}(t) - \mathbf{v}\|$ in

(6.25)) arises as a consequence of each agent i using an estimate $z_{ii}(t)$ of π_i generated from the estimation step (6.7). Finally, since we do not require the constraint sets to be bounded or identical (or have a nonempty interior), the projection error ϕ_i is not guaranteed to be bounded a priori and the term $(F(\bar{\mathbf{x}}(t)) - F(\mathbf{v}))$ does not reflect the global objective error (as $\bar{\mathbf{x}}(t)$ need not be in X). Therefore, quantifying the behaviors of these terms and errors will be the main challenging task in analyzing the convergence as well as the convergence rates of our algorithms; this calls for new results that are more accessible than (6.13) as we develop in the sequel.

We now provide some bounds on the terms $\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|$ and $\|\phi_i(t)\|$ appearing in (6.13) in terms of the step size sequence $\gamma(t)$ and the total projection error $\beta(t)$, defined as

$$\beta(t) := \sum_{i \in \mathcal{V}} \|\phi_i(t)\|, \quad \forall t \geq 0. \quad (6.26)$$

Theorem 6.3.7. *Let Assumptions 6.2.1 (Basic Problem Assumptions), 6.2.2 (Connectivity), 6.2.3 (Unique ID), and 6.2.4 (Weight Rule) hold. The following hold for both Algorithms 6.1 and 6.2:*

(a) Let $D_4 := C \sum_{j \in \mathcal{V}} \|\mathbf{x}_j(0)\|$. For any $i \in \mathcal{V}$,

$$\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| \leq D_4 \lambda^t + D_1 \sum_{0 \leq s \leq t-1} \lambda^{t-s} \gamma(s) + C \sum_{0 \leq s \leq t-1} \lambda^{t-s} \beta(s). \quad (6.27)$$

(b) Define

$$\theta(t) := \gamma(t) \sum_{0 \leq s \leq t-1} \lambda^{t-s} \beta(s), \quad \theta(0) = 0. \quad (6.28)$$

If $\{\gamma(t)\}$ is nonincreasing, then

$$\theta(t+1) \leq \lambda \theta(t) + \lambda \gamma(t) \beta(t). \quad (6.29)$$

Proof. (a) First, we express (6.6) and (6.8) in the form

$$\mathbf{x}_i(t+1) = \sum_{j \in \mathcal{V}} w_{ij} \mathbf{x}_j(t) + \boldsymbol{\epsilon}_i(t), \quad (6.30)$$

where $\boldsymbol{\epsilon}_i(t) \in \mathbb{R}^m$ is an error term. Then, we have

$$\mathbf{x}_i(t) = \sum_{j \in \mathcal{V}} [W^t]_{ij} \mathbf{x}_j(0) + \sum_{0 \leq s \leq t-1} \sum_{j \in \mathcal{V}} [W^{t-s}]_{ij} \boldsymbol{\epsilon}_j(s).$$

Since $\bar{\mathbf{x}}(t) = \sum_{j \in \mathcal{V}} \pi_j \mathbf{x}_j(t)$ and $\boldsymbol{\pi}^T W = \boldsymbol{\pi}^T$, it follows that

$$\bar{\mathbf{x}}(t) = \sum_{j \in \mathcal{V}} \pi_j \mathbf{x}_j(0) + \sum_{0 \leq s \leq t-1} \sum_{j \in \mathcal{V}} \pi_j \boldsymbol{\epsilon}_j(s).$$

Thus, the term $\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|$ can be expressed as

$$\begin{aligned} & \left\| \sum_{j \in \mathcal{V}} ([W^t]_{ij} - \pi_j) \mathbf{x}_j(0) + \sum_{s=0}^{t-1} \sum_{j \in \mathcal{V}} ([W^{t-s}]_{ij} - \pi_j) \boldsymbol{\epsilon}_j(s) \right\| \\ & \leq \sum_{j \in \mathcal{V}} |[W^t]_{ij} - \pi_j| \|\mathbf{x}_j(0)\| + \sum_{s=0}^{t-1} \sum_{j \in \mathcal{V}} |[W^{t-s}]_{ij} - \pi_j| \|\boldsymbol{\epsilon}_j(s)\|. \end{aligned}$$

Hence, by using the bound in (6.10), we then have

$$\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| \leq D_4 \lambda^t + C \sum_{0 \leq s \leq t-1} \lambda^{t-s} \sum_{j \in \mathcal{V}} \|\boldsymbol{\epsilon}_j(s)\|. \quad (6.31)$$

Now consider Algorithm 6.1, where it follows from (6.6) and (6.14) that $\boldsymbol{\epsilon}_i(t) = \phi_i(t) - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)}$. By using the triangle inequality and the facts that $\|\mathbf{g}_i(t)\| \leq L_f$ (cf. Assumption 6.2.1(b)) and that $z_{ii}^{-1} \leq \eta$ (see (6.11)), we obtain

$$\|\boldsymbol{\epsilon}_i(t)\| \leq \|\phi_i(t)\| + \gamma(t) L_f \eta, \quad \forall i \in \mathcal{V}. \quad (6.32)$$

Next, we show that this bound also holds for Algorithm 6.2. From (6.8) and (6.15)

we have $\boldsymbol{\epsilon}_i(t) = \phi_i(t) - \gamma(t) \sum_{j \in \mathcal{V}} w_{ij} \frac{\mathbf{g}_j(t)}{z_{jj}(t)}$. As a result, for $\forall i \in \mathcal{V}$

$$\|\boldsymbol{\epsilon}_i(t)\| \leq \|\phi_i(t)\| + \gamma(t) \sum_{j \in \mathcal{V}} w_{ij} \frac{\|\mathbf{g}_j(t)\|}{|z_{jj}(t)|} \leq \|\phi_i(t)\| + \gamma(t) L_f \eta.$$

By combining (6.32) and (6.31) and rearranging terms, we have

$$\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| \leq D_4 \lambda^t + C \sum_{0 \leq s \leq t-1} \lambda^{t-s} (N\gamma(s)L_f\eta + \beta(s)).$$

(b) By using the definition of $\theta(t)$ and the monotonicity of $\{\gamma(t)\}$, we have

$$\theta(t+1) \leq \gamma(t) \sum_{0 \leq s \leq t} \lambda^{t+1-s} \beta(s) = \lambda\theta(t) + \lambda\gamma(t)\beta(t),$$

which concludes the proof. \square

We note the following. First, it is clear from (6.27) that the effect of initial conditions on the differences between agents' states vanishes exponentially. Second, one can view the last two terms on the right side of (6.27) as the convolutions of $\gamma(t)$ and $\beta(t)$ with λ^t . Thus, for the convergence of the algorithms, we expect these terms to decay to zero under a suitable choice of $\gamma(t)$. For example, when $\lim_{t \rightarrow \infty} \gamma(t) = 0$, we show next that $\lim_{t \rightarrow \infty} \sum_{s=0}^{t-1} \lambda^{t-s} \gamma(s) = 0$. However, whether this also implies $\lim_{t \rightarrow \infty} \sum_{s=0}^{t-1} \lambda^{t-s} \beta(s) = 0$ is inconclusive since $\beta(t)$ depends on the agents' states and the sets X_i . Finally, we introduced $\theta(t)$ in order to study the behavior of the term $\gamma(t) \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|$ in (6.13).

Corollary 6.3.8. *In Theorem 6.3.7, if $\lim_{t \rightarrow \infty} \beta(t) = 0$, then $\lim_{t \rightarrow \infty} \theta(t) = 0$.*

Additionally, if $\lim_{t \rightarrow \infty} \gamma(t) = 0$, then $\lim_{t \rightarrow \infty} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| = 0$.

Proof. Clearly, it suffices to prove that for any $\lambda \in (0, 1)$ and any nonnegative sequence $\{\beta(t)\}_{t \geq 0}$ satisfying $\lim_{t \rightarrow \infty} \beta(t) = 0$, $\lim_{t \rightarrow \infty} \sum_{s=0}^t \lambda^{t-s} \beta(s) = 0$. This claim is stated in [137, Lem. 7]. \square

Our next result is basically a consequence of Theorems 6.3.6 and 6.3.7 under the regularity assumption on the constraint sets. Specifically, we will apply

the bounds obtained in (6.27) and (6.29) to (6.13), and then select suitable associated coefficients to generate a more accessible relation, which is key to proving the convergence as well as convergence rate of the algorithms.

Theorem 6.3.9. *Let Assumptions 6.2.1 (Basic Problem Assumptions), 6.2.2 (Connectivity), 6.2.3 (Unique ID), and 6.2.4 (Weight Rule) be satisfied. The following holds for both Algorithms 6.1 and 6.2 and for any nonincreasing step size sequence $\{\gamma(t)\}$:*

$$\begin{aligned}
& \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 + ab\theta(t+1) \\
& \leq (1 + D_1\lambda^{2t}) \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 + ab\theta(t) \\
& \quad - 2\gamma(t)(F(\mathbf{s}(t)) - F(\mathbf{v})) - D_6 \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2 \\
& \quad + D_{24}\gamma(t)\lambda^t + D_{21}\gamma(t) \sum_{0 \leq s \leq t-1} \lambda^{t-s}\gamma(s) + D'_3\gamma^2(t), \tag{6.33}
\end{aligned}$$

where $\mathbf{s}(t) = P_X(\bar{\mathbf{x}}(t))$, $\pi_{\min} = \min_{i \in \mathcal{V}} \pi_i$, $b = \sqrt{\frac{\pi_{\min}}{N\lambda}}$, $a = \frac{D'_2 C}{(1-\lambda)b}$, $D'_2 = D_2 + \frac{2LR}{\pi_{\min}}$, R is a regularity constant of $\{X_i, i \in \mathcal{V}\}$, $D_6 = \frac{\pi_{\min}}{2}$, $D_{24} = D'_2 D_4$, $D_{21} = D'_2 D_1$ and $D'_3 = \frac{2D_3 + \lambda a^2}{2}$.

Proof. By adding and subtracting $F(\mathbf{s}(t))$ and using the Lipschitz continuity of F we have

$$F(\mathbf{v}) - F(\bar{\mathbf{x}}(t)) \leq F(\mathbf{v}) - F(\mathbf{s}(t)) + L_f \|\mathbf{s}(t) - \bar{\mathbf{x}}(t)\|.$$

Now we find an upper bound on the term $\|\mathbf{s}(t) - \bar{\mathbf{x}}(t)\|$. By the regularity assumption of $\{X_i, i \in \mathcal{V}\}$, there exists R such that $\text{dist}(\mathbf{x}, X) \leq R \max_{i \in \mathcal{V}} \text{dist}(\mathbf{x}, X_i)$, $\forall \mathbf{x} \in$

$\text{conv}(\cup_{i \in \mathcal{V}} X_i)$. As a result, we have

$$\begin{aligned} \|\mathbf{s}(t) - \bar{\mathbf{x}}(t)\| &= \text{dist}(\bar{\mathbf{x}}(t), X) \leq R \max_{i \in \mathcal{V}} \text{dist}(\bar{\mathbf{x}}(t), X_i) \\ &\leq R \sum_{i \in \mathcal{V}} \frac{\pi_i}{\pi_{\min}} \text{dist}(\bar{\mathbf{x}}(t), X_i) \leq R \sum_{i \in \mathcal{V}} \frac{\pi_i}{\pi_{\min}} \|\mathbf{x}_i - \bar{\mathbf{x}}(t)\|, \end{aligned} \quad (6.34)$$

where the last inequality holds since $\text{dist}(\bar{\mathbf{x}}(t), X_i) \leq \|\mathbf{x}_i - \bar{\mathbf{x}}(t)\|$ (cf. Lem. 6.3.1(b)).

Hence,

$$F(\mathbf{v}) - F(\bar{\mathbf{x}}(t)) \leq F(\mathbf{v}) - F(\mathbf{s}(t)) + \frac{L_f R}{\pi_{\min}} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|.$$

Using this bound for (6.13), we then have

$$\begin{aligned} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq (1 + D_1 \lambda^{2t}) \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 \\ &\quad - 2\gamma(t) (F(\mathbf{s}(t)) - F(\mathbf{v})) - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2 \\ &\quad + (D_2 + \frac{2LR}{\pi_{\min}}) \gamma(t) \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| + D_3 \gamma^2(t). \end{aligned}$$

Next, by adding $ab\theta(t+1)$ to both sides of this relation and using the bounds (6.27)

and (6.29), we further have

$$\begin{aligned} &\sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 + ab\theta(t+1) \\ &\leq (1 + D_1 \lambda^{2t}) \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 + ab\theta(t) \\ &\quad + ab(\lambda - 1)\theta(t) + ab\lambda\gamma(t)\beta(t) \\ &\quad - 2\gamma(t) (F(\mathbf{s}(t)) - F(\mathbf{v})) - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2 + D_3 \gamma^2(t) \\ &\quad + D_{24} \gamma(t) \lambda^t + D_{21} \gamma(t) \sum_{0 \leq s \leq t-1} \lambda^{t-s} \gamma(s) + D_2 C \theta(t). \end{aligned} \quad (6.35)$$

Now with the choice of $a = \frac{D'_2 C}{(1-\lambda)b}$, the terms $ab(\lambda - 1)\theta(t)$ and $D'_2 C\theta(t)$ cancel out.

Further, by the Cauchy-Schwarz inequality, we have

$$\begin{aligned} ab\gamma(t)\beta(t) &\leq \frac{a^2\gamma^2(t) + b^2\beta^2(t)}{2} \leq \frac{a^2\gamma^2(t)}{2} + \frac{b^2n}{2} \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2 \\ &= \frac{a^2}{2}\gamma^2(t) + \frac{\pi_{\min}}{2\lambda} \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2. \end{aligned}$$

The last equality holds since $b^2 = \frac{\pi_{\min}}{N\lambda}$. As a result, we have

$$ab\lambda\gamma(t)\beta(t) - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2 \leq \frac{\lambda a^2}{2}\gamma^2(t) - \frac{\pi_{\min}}{2} \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2.$$

It remains to apply the relations above to (6.35) and then rearrange terms to obtain (6.33). \square

It should be noted that (6.33) holds uniformly on X since the constants D_i are independent of the choice of $\mathbf{v} \in X$. When restricted to X^* , we immediately have a relation between the (weighted average) distance (squared) from the optimal solution, i.e., $\sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}^*\|^2$, and the global objective error $F(\mathbf{s}(t)) - F^*$ (as $\mathbf{s}(t) \in X$), both of which are desired to converge under a suitable choice of step size sequence.

We are now ready to give a convergence result that applies to both Algorithms (6.6)-(6.7) and (6.8)-(6.9), whose proof is based on the Theorem 6.3.9 and the following lemma.

Lemma 6.3.10. ([157]) *Let $\{v_t\}_{t=0}^\infty$, $\{u_t\}_{t=0}^\infty$, $\{b_t\}_{t=0}^\infty$ and $\{c_t\}_{t=0}^\infty$ be nonnegative sequences such that $\sum_{t=0}^\infty b_t < \infty$, $\sum_{t=0}^\infty c_t < \infty$ and for $\forall t \geq 0$*

$$v_{t+1} \leq (1 + b_t)v_t - u_t + c_t. \tag{6.36}$$

Then $\{v_t\}$ converges and $\sum_{t=0}^{\infty} u_t < \infty$.

Theorem 6.3.11. (Convergence to optimal solution) *Let Assumptions 6.2.1-6.2.5 be satisfied. Then both Algorithms 6.1 and 6.2 yield convergence to the optimal solution, i.e.,*

$$\exists \mathbf{x}^* \in X^* : \lim_{t \rightarrow \infty} \mathbf{x}_i(t) = \mathbf{x}^*, \quad \forall i \in \mathcal{V}. \quad (6.37)$$

Proof. The proof proceeds in two steps: (i) apply Lemma 6.3.10 to (6.33), and then (ii) prove convergence to the optimal solution.

Step (i): Let \mathbf{x}^\dagger be arbitrary in X^* and define the nonnegative sequences $\{v_t\}$, $\{u_t\}$, $\{b_t\}$ and $\{c_t\}$ as follows:

$$\begin{aligned} v_t &:= \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{x}^\dagger\|^2 + ab\theta(t), & b_t &:= D_1 \lambda^{2t}, \\ u_t &:= 2\gamma(t)(F(\mathbf{s}(t)) - F^*) + D_6 \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2, \\ c_t &:= D_{24}\gamma(t)\lambda^t + D_{21}\gamma(t) \sum_{0 \leq s \leq t-1} \lambda^{t-s}\gamma(s) + D'_3\gamma^2(t). \end{aligned}$$

By adding the nonnegative term $D_1\lambda^{2t}ab\theta(t)$ to the right hand side of (6.33), we obtain

$$v_{t+1} \leq (1 + b_t)v_t - u_t + c_t, \quad \forall t \geq 0.$$

We now show that other conditions of Lemma 6.3.10 also hold, namely $\sum_{t=0}^{\infty} b_t < \infty$ and $\sum_{t=0}^{\infty} c_t < \infty$. The former condition is obvious since $\lambda \in (0, 1)$ implies that $\sum_{t=0}^{\infty} b_t = (1 - \lambda^2)^{-1}$. To prove the latter, consider each term in c_t . First, $\sum_{t \geq 0} \gamma^2(t) < \infty$ by Assumption 6.2.5. Second, by the Cauchy-Schwarz inequality,

we have $\gamma(t)\lambda^t \leq (\gamma^2(t) + \lambda^{2t})/2$. Thus

$$\sum_{t \geq 0} \gamma(t)\lambda^t \leq \frac{1}{2} \sum_{t \geq 0} \gamma^2(t) + \frac{1}{2} \sum_{t \geq 0} \lambda^{2t} < \infty. \quad (6.38)$$

Third, by monotonicity of sequence $\{\gamma(t)\}$ (cf. Assumption 6.2.5) the second term in c_t can be bounded as follows: $\gamma(t) \sum_{s=0}^{t-1} \lambda^{t-s} \gamma(s) \leq \sum_{s=0}^{t-1} \lambda^{t-s} \gamma^2(s) \leq \sum_{s=0}^t \lambda^{t-s} \gamma^2(s)$.

Thus, for any $N \geq 1$ we have

$$\begin{aligned} \sum_{t=1}^N \gamma(t) \sum_{s=0}^{t-1} \lambda^{t-s} \gamma(s) &\leq \sum_{0 \leq s \leq t \leq N} \lambda^{t-s} \gamma^2(s) \leq \sum_{s=0}^N \gamma^2(s) \sum_{t=s}^{\infty} \lambda^{t-s} \\ &= \sum_{0 \leq s \leq N} \frac{\gamma^2(s)}{1-\lambda} \leq \frac{\sum_{s \geq 0} \gamma^2(s)}{1-\lambda} < \infty. \end{aligned}$$

This concludes that $\{c_t\}$ is summable as desired. Therefore, in view of Lemma 6.3.10, the following hold:

$$\exists \lim_{t \rightarrow \infty} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{x}^\dagger\|^2 + ab\theta(t) =: \delta \geq 0 \quad (6.39)$$

$$\sum_{t \geq 0} \gamma(t) (F(\mathbf{s}(t)) - F^*) + \frac{D_6}{2} \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2 < \infty. \quad (6.40)$$

Step (ii): First, by (6.40), we have $\lim_{t \rightarrow \infty} \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2 = 0$. Thus, $\lim_{t \rightarrow \infty} \beta(t) = 0$, which by Corollary 6.3.8 yields $\lim_{t \rightarrow \infty} \theta(t) = 0$. It then follows from (6.39) that

$$\lim_{t \rightarrow \infty} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{x}^\dagger\|^2 = \delta. \quad (6.41)$$

As a result, for each $i \in \mathcal{V}$, $\{\mathbf{x}_i(t)\}_{t \geq 0}$ is a bounded sequence. Thus so are $\{\bar{\mathbf{x}}(t)\}_{t \geq 0}$ and $\{\mathbf{s}(t)\}_{t \geq 0}$.

Next, since $\sum_{t \geq 0} \gamma(t) = \infty$, it then follows from (6.40) that $\liminf_{t \rightarrow \infty} F(\mathbf{s}(t)) = F^*$. Thus, there exists a subsequence $\{\mathbf{s}(t_k)\} \subseteq \{\mathbf{s}(t)\}$ such that

$$\lim_{k \rightarrow \infty} F(\mathbf{s}(t_k)) = F^*. \quad (6.42)$$

Now since $\{\mathbf{s}(t_k)\}$ is also a bounded sequence, there exists a convergent subsequence $\{\mathbf{s}(t_l)\} \subseteq \{\mathbf{s}(t_k)\}$. Denote $\lim_{l \rightarrow \infty} \mathbf{s}(t_l) = \mathbf{x}^*$ for some $\mathbf{x}^* \in X$ (since X is closed).

We next show that $\mathbf{x}^* \in X^*$. By the continuity of F on \mathbb{R}^m

$$\lim_{l \rightarrow \infty} F(\mathbf{s}(t_l)) = F(\mathbf{x}^*). \quad (6.43)$$

which in view of (6.42) implies that $F(\mathbf{x}^*) = F^*$. By convexity of F , we conclude that $\mathbf{x}^* \in X^*$. Since $\mathbf{x}^\dagger \in X^*$ was chosen arbitrarily, we can let $\mathbf{x}^\dagger = \mathbf{x}^*$.

Now it remains to show that $\delta = 0$, which by (6.41) will then complete the proof. By the triangle and Cauchy-Schwarz inequalities

$$\begin{aligned} \|\mathbf{x}_i(t) - \mathbf{x}^*\|^2 &\leq (\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| + \|\bar{\mathbf{x}}(t) - \mathbf{s}(t)\| + \|\mathbf{s}(t) - \mathbf{x}^*\|)^2 \\ &\leq 3(\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2 + \|\bar{\mathbf{x}}(t) - \mathbf{s}(t)\|^2 + \|\mathbf{s}(t) - \mathbf{x}^*\|^2). \end{aligned}$$

Next, since $\|\bar{\mathbf{x}}(t) - \mathbf{s}(t)\| \leq \frac{R}{\pi_{\min}} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i - \bar{\mathbf{x}}(t)\|$ (cf. (6.34)), we have $\|\mathbf{s}(t) - \bar{\mathbf{x}}(t)\|^2 \leq \frac{R^2}{\pi_{\min}^2} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i - \bar{\mathbf{x}}(t)\|^2$ by Lemma 6.3.2. As a result,

$$\frac{1}{3} \|\mathbf{x}_i(t) - \mathbf{x}^*\|^2 \leq \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2 + \frac{R^2}{\pi_{\min}^2} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i - \bar{\mathbf{x}}(t)\|^2 + \|\mathbf{s}(t) - \mathbf{x}^*\|^2.$$

Multiplying both sides by π_i and summing over $i \in \mathcal{V}$ yields

$$\sum_{i \in \mathcal{V}} \frac{\pi_i}{3} \|\mathbf{x}_i(t) - \mathbf{x}^*\|^2 \leq R' \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2 + \|\mathbf{s}(t) - \mathbf{x}^*\|^2,$$

where $R' = 1 + \frac{R^2}{\pi_{\min}^2}$. Taking \liminf as $t \rightarrow \infty$ both sides of this inequality and using (6.41) yield:

$$\begin{aligned} \frac{\delta}{3} &\leq \liminf_{t \rightarrow \infty} \left(R' \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2 + \|\mathbf{s}(t) - \mathbf{x}^*\|^2 \right) \\ &= \liminf_{t \rightarrow \infty} \|\mathbf{s}(t) - \mathbf{x}^*\|^2. \end{aligned} \quad (6.44)$$

Here we have used the superadditivity property of the limit inferior and the fact that $\lim_{t \rightarrow \infty} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|^2 = 0$ since $\lim_{t \rightarrow \infty} \beta(t) = 0$ (see Corollary 6.3.8). Since the subsequence $\{\mathbf{s}(t_l)\}$ converges to \mathbf{x}^* , we have $\liminf_{t \rightarrow \infty} \|\mathbf{s}(t) - \mathbf{x}^*\| = 0$, which in view of (6.44) implies that $\delta = 0$. \square

6.4 Rate of Convergence

We now discuss the convergence rate of our algorithms, which evidently depends on the choice of $\gamma(t)$. Since the estimation step (6.7) converges exponentially, one should expect that the convergence rate of the objective error is equivalent to that of usual distributed subgradient methods in the case when the constraint sets are identical and/or compact. We emphasize, however, that such assumptions are relaxed in our work, i.e., the sets X_i can be nonidentical and unbounded. Moreover, the global constraint set X is also allowed to have an empty interior. Thus, for all $i \in \mathcal{V}$, the agents' estimates $\mathbf{x}_i(t)$ as well as their weighted average $\bar{\mathbf{x}}(t)$ need not be in the set X at any time t . As a result, local analysis around the optimal solution does not readily apply.

In this work, to quantify the distance from the optimum, we propose to use a combined error term which involves (i) the distance from a local estimate $\tilde{\mathbf{x}}_i(t)$ of each agent to some point $\tilde{\mathbf{s}}(t) \in X$ and (ii) the global objective error evaluated at $\tilde{\mathbf{s}}(t)$, i.e., $F(\tilde{\mathbf{s}}(t)) - F^*$. Specifically, we define

$$\tilde{\mathbf{x}}_i(t) := \frac{\sum_{k=0}^t \gamma(k) \mathbf{x}_i(k)}{\sum_{k=0}^t \gamma(k)}, \quad \tilde{\mathbf{s}}(t) := \frac{\sum_{k=0}^t \gamma(k) \mathbf{s}(k)}{\sum_{k=0}^t \gamma(k)}. \quad (6.45)$$

Here, for each $t \geq 1$, $\tilde{\mathbf{x}}_i(t)$ is a convex combination of $\mathbf{x}_i(0), \mathbf{x}_i(1), \dots, \mathbf{x}_i(t)$, which

can be computed locally by agent i but might not be in X . In contrast, $\tilde{\mathbf{s}}(t)$ always belongs to X but is not directly available to each agent. The following theorem asserts that both errors $\|\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{s}}(t)\|$ and $F(\tilde{\mathbf{s}}(t)) - F^*$ decay as $O(\frac{\sum_{k=0}^t \gamma^2(k)}{\sum_{k=0}^t \gamma(k)})$.

Theorem 6.4.1. (Convergence rate) *Let Assumptions 6.2.1 (Basic Problem Assumptions), 6.2.2 (Connectivity), 6.2.3 (Unique ID) and 6.2.4 (Weight Rule) hold. Let $\{\gamma(t)\}$ be a nonnegative and nonincreasing sequence. Then for both Algorithms 6.1 and 6.2, the following holds for $\forall t \geq 0$:*

$$C_0 \|\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{s}}(t)\| + F(\tilde{\mathbf{s}}(t)) - F^* \leq \frac{C_1 + C_2 \sum_{k=0}^t \gamma^2(k)}{\sum_{k=0}^t \gamma(k)}, \quad (6.46)$$

where $C_0 = \frac{D_6(1-\lambda)}{2N(N+1)C\lambda}$, some $C_1 > 0$ and $C_2 = O(\frac{N^4 C^2}{(1-\lambda)^2} e^{\frac{D_1}{1-\lambda^2}})$ as $N \rightarrow \infty$ and $\lambda \rightarrow 1$ (recalling that $D_1 = NCL_f\eta$ and $D_6 = \pi_{\min}/2$). Moreover, if $\{X_i, i \in \mathcal{V}\}$ are compact, the constant C_2 is $O(\frac{N^4 C^2}{(1-\lambda)^2})$.

The proof of Theorem 6.4.1 is structured in the following steps:

- (i) Use the bound (6.33) in Theorem 6.3.9 to upper estimate the sum

$$\sum_{k=0}^t 2\gamma(k)(F(\mathbf{s}(k)) - F^*) + D_6 \sum_{i \in \mathcal{V}} \|\phi_i(k)\|^2$$

in terms of $\sum_{0 \leq k \leq t} \gamma(k)$ and $\sum_{k=0}^t \gamma^2(k)$.

- (ii) Relate the left side of (6.46) to this sum by using the convexity of F and the bounds given in Theorem 6.3.7.
- (iii) Analyze the constants C_i .

The following technical lemma will be used in Step (i) for the general case where $\{X_i, \forall i \in \mathcal{V}\}$ are not necessarily bounded.

Lemma 6.4.2. For any $D > 0$ and $\lambda \in (0, 1)$, it holds that

$$1 + \frac{D}{1 - \lambda} \leq \prod_{t \geq 0} (1 + D\lambda^t) \leq e^{\frac{D}{1 - \lambda}}. \quad (6.47)$$

Proof of Lemma 6.4.2. Note that for any $T \geq 1$ we have

$$1 + D \sum_{0 \leq t \leq T} \lambda^t \leq \prod_{0 \leq t \leq T} (1 + D\lambda^t) \leq e^{D \sum_{t=0}^T \lambda^t}$$

where the second inequality follows from the basic relation that $1 + x \leq e^x$ for any $x \geq 0$. Taking the limit as $T \rightarrow \infty$ yields the desired result. \square

Proof of Theorem 6.4.1. We proceed through the 3 steps described above.

Step (i): Let the nonnegative sequences $\{v_t\}$, $\{u_t\}$, $\{b_t\}$ and $\{c_t\}$ be defined as in Step (i) of the proof of Theorem 6.3.11, i.e.,

$$\begin{aligned} v_t &:= \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{x}^*\|^2 + ab\theta(t), & b_t &:= D_1 \lambda^{2t}, \\ u_t &:= 2\gamma(t)(F(\mathbf{s}(t)) - F^*) + D_6 \Phi_t, & \Phi_t &:= \sum_{i \in \mathcal{V}} \|\phi_i(t)\|^2 \\ c_t &:= D_{24}\gamma(t)\lambda^t + D_{21}\gamma(t) \sum_{0 \leq s \leq t-1} \lambda^{t-s}\gamma(s) + D_3'\gamma^2(t). \end{aligned}$$

By using Theorem 6.3.9 and adding the nonnegative term $b_t ab\theta(t)$ to the right hand side of (6.33), we have

$$v_{t+1} \leq (1 + b_t)v_t - u_t + c_t, \quad \forall t \geq 0,$$

which then implies that

$$v_{t+1} \leq \prod_{0 \leq k \leq t} (1 + b_k)v_0 + \sum_{0 \leq k \leq t} (c_k - u_k) \prod_{k+1 \leq s \leq t} (1 + b_s). \quad (6.48)$$

By Lemma 6.4.2, the following holds for any $t, k \geq 0$

$$1 < \prod_{k \leq s \leq t} (1 + b_s) < e^{\frac{D_1}{1-\lambda^2}} =: D_e.$$

As a result, (6.48) implies that

$$v_{t+1} \leq D_e v_0 + \sum_{0 \leq k \leq t} D_e c_k - \sum_{0 \leq k \leq t} u_k, \quad (6.49)$$

from which by rearranging terms and using the fact that $v_{t+1} \geq 0$, we have (recalling the definition of u_t)

$$\sum_{0 \leq k \leq t} \gamma(k)(F(\mathbf{s}(k)) - F^*) + \frac{D_6}{2} \Phi_k \leq R_1 + R_2 \sum_{0 \leq k \leq t} c_k, \quad (6.50)$$

where $R_1 = D_e v_0/2$ and $R_2 = D_e/2$. Next, we will derive an upper bound on the term $\sum_{k=0}^t c_k$ based on the following estimates:

$$\sum_{0 \leq k \leq t} \gamma(k) \lambda^k \leq \sum_{0 \leq k \leq t} \gamma(0) \lambda^k \leq \frac{\gamma(0)}{1-\lambda}, \quad (6.51)$$

and

$$\begin{aligned} \sum_{0 \leq k \leq t} \gamma(k) \sum_{0 \leq s \leq k} \lambda^{k-s} \gamma(s) &\leq \sum_{0 \leq k \leq t} \sum_{0 \leq s \leq k} \gamma^2(s) \lambda^{k-s} = \sum_{0 \leq s \leq t} \gamma^2(s) \sum_{s \leq k \leq t} \lambda^{k-s} \\ &\leq \frac{\sum_{s=0}^t \gamma^2(s)}{1-\lambda}. \end{aligned} \quad (6.52)$$

Hence,

$$\sum_{0 \leq k \leq t} c_k \leq \frac{D_{24} \gamma(0)}{1-\lambda} + \left(\frac{D_{21}}{1-\lambda} + D'_3 \right) \sum_{0 \leq k \leq t} \gamma^2(k).$$

Therefore,

$$\sum_{0 \leq k \leq t} \gamma(k)(F(\mathbf{s}(k)) - F^*) + \frac{D_6}{2} \Phi_k \leq M_1 + M_2 \sum_{0 \leq k \leq t} \gamma^2(k), \quad (6.53)$$

where

$$M_1 = R_1 + \frac{R_2 D_{24} \gamma(0)}{1 - \lambda}, \quad M_2 = R_2 \left(\frac{D_{21}}{1 - \lambda} + D'_3 \right).$$

Step (ii): Now we derive lower bounds on the left hand side of (6.53). Recall that $\tilde{\mathbf{s}}(t) = \sum_{k=0}^t \gamma(k) \mathbf{s}(k) / \sum_{k=0}^t \gamma(k)$. By convexity of F , we then have

$$F(\tilde{\mathbf{s}}(t)) - F^* \leq \frac{\sum_{k=0}^t \gamma(k) (F(\mathbf{s}(k)) - F^*)}{\sum_{k=0}^t \gamma(k)}. \quad (6.54)$$

Next, we will relate the term $\|\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{s}}(t)\|$ with $\sum_{k=0}^t \Phi_k$. By the triangle inequality, it can be shown that

$$\|\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{s}}(t)\| \leq \frac{\sum_{k=0}^t \gamma(k) \|\mathbf{x}_i(k) - \mathbf{s}(k)\|}{\sum_{k=0}^t \gamma(k)}. \quad (6.55)$$

We now quantify the numerator of the right hand side of (6.55). First, note that (cf. (6.27))

$$\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| \leq D_4 \lambda^t + D_1 \sum_{0 \leq s \leq t-1} \lambda^{t-s} \gamma(s) + C \sum_{0 \leq s \leq t-1} \lambda^{t-s} \beta(s).$$

Second, let R be a regularity constant of $\{X_i, i \in \mathcal{V}\}$. Then

$$\begin{aligned} \|\mathbf{s}(t) - \bar{\mathbf{x}}(t)\| &= \text{dist}(\bar{\mathbf{x}}(t), X) \leq R \max_{i \in \mathcal{V}} \text{dist}(\bar{\mathbf{x}}(t), X_i) \\ &\leq R \sum_{i \in \mathcal{V}} \text{dist}(\bar{\mathbf{x}}(t), X_i) \leq R \sum_{i \in \mathcal{V}} \|\mathbf{x}_i - \bar{\mathbf{x}}(t)\|. \end{aligned}$$

Thus, by the triangle inequality and the two previous relations,

$$\frac{\|\mathbf{x}_i(k) - \mathbf{s}(k)\|}{(N+1)C} \leq \frac{D_4}{C} \lambda^t + \frac{D_1}{C} \sum_{0 \leq s \leq t-1} \lambda^{t-s} \gamma(s) + \sum_{0 \leq s \leq t-1} \lambda^{t-s} \beta(s)$$

which implies that (see the definition of $\theta(t)$ in Theorem 6.3.7(b))

$$\begin{aligned}
& \sum_{0 \leq k \leq t} \gamma(k) \frac{\|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\|}{(N+1)C} \\
& \leq \frac{D_4}{C} \sum_{0 \leq k \leq t} \gamma(k) \lambda^k + \frac{D_1}{C} \sum_{0 \leq k \leq t} \gamma(k) \sum_{0 \leq s \leq k-1} \lambda^{k-s} \gamma(s) + \sum_{0 \leq k \leq t} \theta(k). \\
& \stackrel{(6.51)-(6.52)}{\leq} \frac{D_4 \gamma(0)}{(1-\lambda)C} + \frac{D_1}{(1-\lambda)C} \sum_{0 \leq s \leq t} \gamma^2(s) + \sum_{0 \leq k \leq t} \theta(k). \tag{6.56}
\end{aligned}$$

The last term can be bounded as follows. By (6.29) and noting that $\theta(0) = 0$, we have

$$\begin{aligned}
\sum_{0 \leq k \leq t} \theta(k) & \leq \lambda \sum_{0 \leq k \leq t-1} \theta(k) + \lambda \sum_{0 \leq k \leq t-1} \gamma(k) \beta(k) \\
& \leq \lambda \sum_{0 \leq k \leq t} \theta(k) + \lambda \sum_{0 \leq k \leq t} \frac{\gamma^2(k)}{4} + \beta^2(k),
\end{aligned}$$

where we have used the fact that $\gamma\beta \leq \frac{\gamma^2}{4} + \beta^2, \forall \gamma, \beta \in \mathbb{R}$. Rearranging terms yields

$$\sum_{0 \leq k \leq t} \theta(k) \leq \frac{\lambda}{1-\lambda} \sum_{0 \leq k \leq t-1} \frac{\gamma^2(k)}{4} + \beta^2(k). \tag{6.57}$$

Moreover, by the Cauchy-Schwarz inequality, we have

$$\sum_{0 \leq k \leq t} \beta^2(k) = \sum_{0 \leq k \leq t} \left(\sum_{i \in \mathcal{V}} \|\phi_i(k)\| \right)^2 \leq \sum_{0 \leq k \leq t} N \sum_{i \in \mathcal{V}} \|\phi_i(k)\|^2 \leq N \sum_{0 \leq k \leq t} \Phi_k. \tag{6.58}$$

Using this bound and (6.57) for (6.56), we obtain

$$C_0 \sum_{0 \leq k \leq t} \gamma(k) \|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\| \leq M_3 + M_4 \sum_{0 \leq k \leq t} \gamma^2(k) + \frac{D_6}{2} \sum_{0 \leq k \leq t} \Phi_k,$$

$$\text{with } C_0 = \frac{D_6(1-\lambda)}{2N(N+1)C\lambda}, \quad M_3 = \frac{D_6 D_4 \gamma(0)}{2N\lambda C}, \quad M_4 = \frac{D_6}{2N} \left(\frac{D_1}{\lambda C} + \frac{1}{4} \right).$$

Combining the inequality above with (6.53) yields

$$\begin{aligned}
& C_0 \sum_{0 \leq k \leq t} \gamma(k) \|\mathbf{x}_i(k) - \bar{\mathbf{x}}(k)\| + \sum_{0 \leq k \leq t} \gamma(k) (F(\mathbf{s}(k)) - F^*) \\
& \leq (M_1 + M_3) + (M_2 + M_4) \sum_{0 \leq k \leq t} \gamma^2(k).
\end{aligned}$$

Let $C_1 := M_1 + M_3, C_2 := M_2 + M_4$. Dividing both sides by $\sum_{k=0}^t \gamma(k)$ and then using (6.54) and (6.55) yields (6.46) as desired, i.e.,

$$C_0 \|\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{s}}(t)\| + F(\tilde{\mathbf{s}}(t)) - F^* \leq \frac{C_1 + C_2 \sum_{k=0}^t \gamma^2(k)}{\sum_{k=0}^t \gamma(k)}.$$

Step (iii): We now discuss the constant associated with the convergence rate in terms of the network size and the spectral gap $1 - \lambda$. To this end, we assume, for simplicity that $\pi_{\min}^{-1} = O(N)$ (in fact $\pi_{\min} \leq \frac{1}{N}$) and that $\eta = O(N)$. Then it can be verified that the dominant term is M_2 in C_2 , which is

$$O\left(\frac{N^4 C^2 R_2}{(1 - \lambda)^2}\right) = O\left(\frac{N^4 C^2}{(1 - \lambda)^2} e^{\frac{D_1}{1 - \lambda^2}}\right).$$

A better estimate can be obtained if we assume further that $\{X_i, i \in \mathcal{V}\}$ are compact. In this case, there exists $D_X > 0$ such that $\|\mathbf{x}_i(t) - \mathbf{x}^*\|^2 \leq D_X, \forall i \in \mathcal{V}, \forall t \geq 0$. Thus, by using Theorem 6.3.9, we have for any $t \geq 0$

$$\begin{aligned} v_{t+1} &\leq v_t + b_t \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{x}^\dagger\|^2 - u_t + c_t \\ &\leq v_t + D_X b_t - u_t + c_t \\ &\leq v_0 + \sum_{0 \leq k \leq t} D_X b_k - u_k + c_k \\ &\leq v_0 + \frac{D_X D_1}{1 - \lambda^2} + \sum_{0 \leq k \leq t} c_k - u_k. \end{aligned}$$

Here we have used the facts that $\sum_{i \in \mathcal{V}} \pi_i = 1$ and

$$\sum_{k=0}^t b_k = D_1 \sum_{k=1}^t \lambda^{2k} \leq \frac{D_1}{1 - \lambda^2}.$$

As a result,

$$\sum_{0 \leq k \leq t} u_k \leq v_0 + \frac{D_1 D_X}{1 - \lambda^2} + \sum_{0 \leq k \leq t} c_k, \quad \forall t \geq 0. \quad (6.59)$$

Thus, we have that (6.50) still holds but with $R_1 = v_0 + \frac{D_1 D_X}{1-\lambda^2}$ and $R_2 = 1$ (compared to $R_2 = D_\epsilon/2$ as before). Hence, the constant C_2 reduces to $O(\frac{N^4 C^2}{(1-\lambda)^2})$. \square

We remark that the explicit formulas for C_1 and C_2 obtained in the proof are rather involved. Thus to simplify the estimate orders of C_2 , we have assumed that $\pi_{\min}^{-1} = O(N)$ (in fact $\pi_{\min} \leq \frac{1}{N}$) and that $\eta = O(N)$. Note also that the spectral gap, defined as $1 - |\lambda_2(W)|$, also affects the constant bounds since $|\lambda_2(W)| < \lambda < 1$, signifying the importance of the strong connectivity assumption.

This result demonstrates how the convergence property of the step size sequence implies that of our algorithms; as a side note Assumption 6.2.5 is not needed for (6.46) to hold. In particular, convergence rate analysis now boils down to studying the behavior of the right side of (6.46); exactly the same task has been carried out thoroughly in the literature for centralized (projected) subgradient methods (see, e.g., [115, 117, 138]). Thus, we proceed no further than providing a few notable results and proving another convergence bound on the objective error in the case of identical constraint sets.

Corollary 6.4.3. *Let the assumptions of Theorem 6.4.1 be satisfied. Let $E(t) = \frac{C_1 + C_2 \sum_{k=0}^t \gamma^2(k)}{\sum_{k=0}^t \gamma(k)}$. The following hold.*

(a) *If $\gamma(t) \equiv \gamma$, then $E(t) = C_2 \gamma + \frac{C_1}{\gamma t}$. If $\lim_{t \rightarrow \infty} \gamma(t) = 0$ and $\sum_{t \geq 0} \gamma(t) = \infty$,*

then $\lim_{t \rightarrow \infty} E(t) = 0$.

(b) *If $\{X_i, i \in \mathcal{V}\}$ are identical, then there exist $\tilde{C}_1 > 0, \tilde{C}_2 > 0$ such that*

$$F(\tilde{\mathbf{x}}_i(t)) - F^* \leq \frac{\tilde{C}_1 + \tilde{C}_2 \sum_{k=0}^t \gamma^2(k)}{\sum_{k=0}^t \gamma(k)}. \quad (6.60)$$

Further, if $\gamma(t) = O(\frac{1}{\sqrt{t}})$ then $F(\tilde{\mathbf{x}}_i(t)) - F^* = O(\frac{\ln t}{\sqrt{t}})$.

Proof. We only prove (6.60) in part (b). Note that $\mathbf{x}_i(t) \in X$ for $\forall t \geq 0$ and $\forall i \in \mathcal{V}$. By Lipschitz continuity of F , we have $F(\tilde{\mathbf{x}}_i(t)) - F^* = F(\tilde{\mathbf{x}}_i(t)) - F(\tilde{\mathbf{s}}(t)) + F(\tilde{\mathbf{s}}(t)) - F^* \leq NL_f \|\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{s}}(t)\| + F(\tilde{\mathbf{s}}(t)) - F^*$. It remains to use (6.46). \square

Note that for unconstrained problems, the convergence rate of $O(\frac{\ln t}{\sqrt{t}})$ is also achieved by recent distributed subgradient based methods such as Dual Averaging [81] or Subgradient-Push [145].

6.5 Numerical Example

Consider a machine learning problem via the l_1 -norm regularized logistic loss functions

$$\min_{\mathbf{x} \in X} F(\mathbf{x}) = \sum_{1 \leq i \leq r} \log \left(1 + \exp \left(-l_i(\mathbf{p}_i^T \mathbf{u} + v) \right) \right) + \mu \|\mathbf{u}\|_1$$

with variable $\mathbf{x} = [\mathbf{u}^T, v]^T$, $\mathbf{u} \in \mathbb{R}^m$, $v \in \mathbb{R}$. Here, $\mu > 0$ is a regularization parameter.

The training set consists of r pairs (\mathbf{p}_i, l_i) where $\mathbf{p}_i \in \mathbb{R}^m$ is a feature vector and $l_i \in \{-1, 1\}$ is the corresponding label. Suppose that \mathbf{x} satisfies a linear equality constraint: $X = \{\mathbf{x} \in \mathbb{R}^{m+1}, A_{eq}\mathbf{x} = \mathbf{b}_{eq}\}$, where $A_{eq} \in \mathbb{R}^{q \times (m+1)}$ and $\mathbf{b}_{eq} \in \mathbb{R}^q$. In general, when the problem data is distributed or too large to store and/or process on a single machine, employing a network of machines provides a solution. This arises in many applications such as online social network data, wireless sensor networks, and cloud computing.

In our example, this problem is to be solved by a network of $N = 9$ nodes with the communication graph described in Fig. 6.1. We assume $r = 500$, $m = 50$

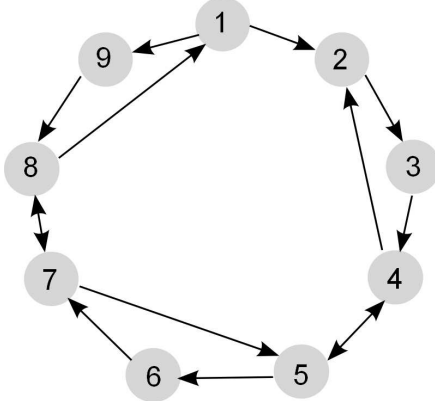


Figure 6.1: Directed communication graph of the network example.

and $q = 36$, and select (\mathbf{p}_i, l_i) , A_{eq} and \mathbf{b}_{eq} based on normally distributed random numbers. We choose $\mu = 50$. Suppose the problem data are distributed among the N nodes as follows: each node i stores a partition \mathcal{P}_i of roughly $\frac{r}{N}$ training data and a set of $\frac{q}{N}$ equality constraints, referred to by $(A_{eq}^{(i)}, b_{eq}^{(i)})$. Thus, for each agent $i \in \mathcal{V}$, the local cost function and constraint set are given by

$$f_i(\mathbf{x}) = \sum_{j \in \mathcal{P}_i} \log \left(1 + \exp \left(-l_j (\mathbf{p}_j^T \mathbf{u} + v) \right) \right) + \frac{\mu}{N} \|\mathbf{u}\|_1$$

$$X_i = \{\mathbf{x} \in \mathbb{R}^{m+1} : A_{eq}^{(i)} \mathbf{x} = \mathbf{b}_{eq}^{(i)}\}.$$

We assume that the weight matrix $W = [w_{ij}]$ is such that $w_{ij} = \frac{1}{|\mathcal{N}_i|}$ if $j \in \mathcal{N}_i$ and $w_{ij} = 0$ otherwise. We carry out simulations with Algorithms 6.1 and 6.2 using step size $\gamma(t) = \frac{1}{N^2(t+1)}$ and the usual DPS method (denoted by DPS-(a)), and its variation DPS-(b) (i.e., the order of the subgradient and consensus steps is reversed) using step size $\gamma'(t) = \frac{1}{N(t+1)}$. Here $\gamma(t)$ and $\gamma'(t)$ are different by a factor N for the sake of comparison since subgradients in our algorithms are scaled by π_i^{-1} (which equals N if W is doubly stochastic). The initial state vectors $\mathbf{x}_i(0) = \mathbf{0}$ for

$\forall i \in \mathcal{V}$. The simulation results in terms of relative errors in the objective function and the optimal solution are shown in Fig. 6.2, where F^* and \mathbf{x}^* are obtained by solving the global problem using a centralized method. Clearly, both Algorithms 6.1 and 6.2 converge to the optimal solution and have similar performances, which are comparable to the DSP methods combined with the reweighting technique [93, 94], where the knowledge of $\boldsymbol{\pi}$ is assumed in advance, (or equivalently (6.6) and (6.8) with $z_{ii}(t) = \pi_i, \forall i \in \mathcal{V}, \forall t \geq 0$). The usual DPS methods fail to converge to the optimal solution. We also consider the case where link $1 \rightarrow 2$ is lost. The reweighting technique requires the whole network to be reprogrammed with a new Perron eigenvector, which may not be available immediately. In contrast, our algorithms are unchanged except for node 2 adjusting its incoming link weights. Clearly convergence is still achieved (since the network is still strongly connected) but slower since the spectral gap decreases.

6.6 Conclusions and Extensions

In this chapter, we have proposed two modified versions of the DPS method that use require only a row stochastic weight matrix and studied their convergence and convergence rates. Moreover, our analysis does not invoke the compactness requirement usually imposed on the local constraint sets and is able to deal with various scenarios, including constrained/unconstrained problems, the sets X_i being bounded/unbounded or identical/nonidentical. It is important to note the following:

First, it is possible to employ other eigenvector estimation schemes in place of

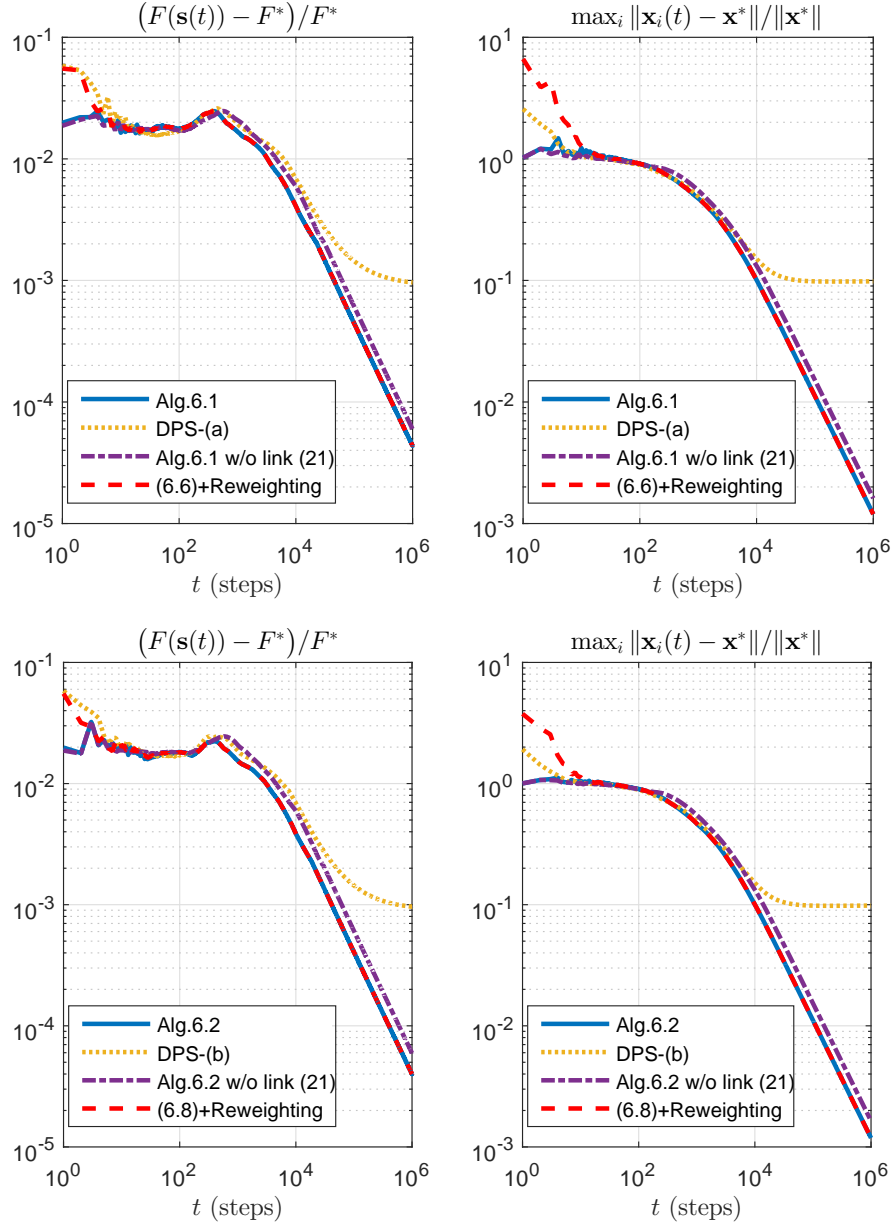


Figure 6.2: Performances of Algorithms 6.1, 6.2, and DSP methods with and without reweighting technique. Reweighting means for each $i \in \mathcal{V}$, π_i is known to agent i in advance and $z_{ii}(t) = \pi_i, \forall t \geq 0$. Here, $\mathbf{s}(t) = P_X(\bar{\mathbf{x}}(t))$.

(6.7) as long as $z_{ii}(t) \rightarrow \pi_i$ sufficiently fast (e.g., satisfying (6.10)). These include any finite-time computation algorithm, e.g., [141]. Moreover, as we have seen from Section 6.4 and also the numerical example, the convergence rate of our algorithms is much slower than the estimation step offered by (6.7). Therefore, it is also possible

to have (6.7) run asynchronously with (6.6), for example, at a slower time scale to save communication bandwidth for exchanging \mathbf{x}_i variables and/or to reliably communicate \mathbf{z}_i with no errors as it is important for the scaling step used in (6.6).

Second, the convergence analysis developed in this chapter can be adapted to either relax the compactness requirement in others projected subgradient based methods (e.g., [94, 137]) or impose regular constraints to other subgradient based algorithms (e.g., [134, 145]); this holds even when the network is time-varying and possibly with fixed communication delays.

Third, the idea of using the augmented iteration (6.7) to adjust (sub)gradient magnitudes as in (6.6) is not only applicable to the distributed projected subgradient methods, but also can be employed to alleviate the condition of the weight matrix being doubly stochastic for some other existing distributed algorithms (using consensus and (sub)gradient steps). For example, we have observed through simulations that the gradient-based method proposed in [136, 144] can be modified in the same spirit and still retains fast convergence speed under a suitable constant step size.

Based on this idea, we have recently proposed a new algorithm [158] that converges linearly under the strong convexity assumption on the cost functions. We now briefly introduce this algorithm.

Algorithm 6.3. *For any $t \geq 0$, each agent i maintains 3 vectors $\mathbf{x}_i(t), \mathbf{y}_i(t) \in$*

$\mathbb{R}^m, \mathbf{z}_i(t) \in \mathbb{R}^N$ and update them as follows:

$$\mathbf{x}_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{x}_j(t) - \gamma \mathbf{y}_i(t) \quad (6.61)$$

$$\mathbf{z}_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{z}_j(t) \quad (6.62)$$

$$\mathbf{y}_i(t+1) = \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{y}_j(t) + \frac{\mathbf{g}_i(t+1)}{z_{ii}(t+1)} - \frac{\mathbf{g}_i(t)}{z_{ii}(t)}, \quad (6.63)$$

where initial estimate $\mathbf{x}_i(0) \in X_i$, $\mathbf{y}_i(0) = \nabla f_i(\mathbf{x}_i(0))$, $\mathbf{z}_i(0) = \mathbf{e}_i \in \mathbb{R}^N$, γ is a positive constant step size, and $\mathbf{g}_i(t) = \nabla f_i(\mathbf{x}_i(t))$.

Assumption 6.6.1. (Lipschitz continuous gradients and strong convexity) *The functions f_i are differentiable and strongly convex. Moreover, the gradients ∇f_i are Lipschitz continuous.*

Theorem 6.6.2. ([158]) *Suppose $X_i = \mathbb{R}^m, \forall i \in \mathcal{V}$ and let the agents implement Algorithm 6.3. Under Assumptions 6.2.2, 6.2.3, 6.2.4, 6.6.1, there exist $\bar{\gamma} > 0$ and $\mu \in (0, 1)$ such that if $\gamma \in (0, \bar{\gamma})$ then $\|\mathbf{x}_i(t) - \mathbf{x}^*\| = O(\mu^t)$.*

Estimations of $\bar{\gamma}$ and μ are rather involved and conservative; see [158] for details. Note also that although selection of appropriate step size γ requires global information about the network and cost functions (thus centralized initialization), the implementation of the algorithm is distributed and exponential convergence is achieved.

Chapter 7: Conclusions

7.1 Summary of Results

This dissertation developed theory and algorithms that advance the state-of-the-art in analysis and applications of distributed consensus in multi-agent networks where communications are broadcast-based and directed; hence the notion of network asymmetry.

Networks with Leaders: In the first part of the thesis, we considered a DeGroot model with the presence of an external media node, representing a leader, or truth, or a source of news having a constant opinion value.

First, when consensus is the main goal of the leader, we introduced the notion of a persistent leader and developed new sufficient conditions for guaranteeing convergence for both fixed and switching topologies and in the presence of other competing but nonpersistent leaders. We also demonstrated that the results can be readily extended to the case where the persistent leader's opinion is time-varying opinion and the case of communications with time-varying but bounded delays.

Second, we studied the problem of a leader that aims to maximize its influence the opinions of agents in a directed network subject to the constraint that the

number of direct followers selected is not more than K . When there is only one leader and consensus is guaranteed a priori, we characterized the influence of that leader through the transient error of the network while when there is a stubborn agent or a second leader with a competing opinion and consensus is not possible, we measured the leader influence in terms of the steady state error of the network. We described the optimal solution for special cases, namely $K = 1, 2$, in which we introduce a few notions of centrality that can be useful for practical applications. Then for a general K , we studied a general combinatorial problem encompassing many other existing problems in the literature. We proved the supermodularity property of the objective function and the convexity of its continuous relaxation for general directed networks, and then developed practical approaches for suboptimal solutions. We demonstrate through numerical examples that the two approaches can be combined to provide effective tools and better analysis for optimal design of influence spreading in diffusive networks.

Our analysis has been shown to be useful for various applications other than those considered here. In particular, the convexity analysis offers (i) an affirmative answer to a conjecture recently proposed in [105] on optimization of on-chip thermoelectric cooling systems and (ii) a convexity result for the state trajectory of a class of bilinear discrete-time systems. The supermodularity analysis can also be used for sensor selection problems.

Consensus Prediction: In this part, we introduced and studied the problem of consensus prediction in a network whose dynamics is described by a DeGroot model.

By an application of the Hahn-Banach theorem, we established a fundamental relation between the consensus value and network data, that is, if the consensus value can be computed at a particular time for any initial opinions, then it can be expressed as a linear combination of available observation data. This allowed us to prove a tight lower bound on the monitoring time for the case of a single observed node regardless of the method used by the observer. We also demonstrated that this bound can be achieved if the minimal polynomial associated with the observed node is available to the observer. For the case of multiple observed nodes, we proposed a conjecture on lower bounds of the monitoring time and developed algorithms toward achieving the those bounds through local observations and computations. Our results in this direction can also be regarded as a data-driven method for network identification.

Distributed Optimization: First we demonstrated that consensus prediction can be employed for enhancing convergence of the distributed gradient method for solving a distributed convex optimization problem on a strongly connected directed network. The convergence rates of our algorithms are similar to those of the centralized gradient method, including finite time convergence for the case of quadratic objective functions, except being slower by a constant factor depending on the network structure and the weight matrix. The convergence times of these algorithms scale linearly with the network diameter for certain structures (e.g., distance regular graphs) and at most linearly with the network size in general.

Second, we proposed a rescaling technique that enables distributed subgradi-

ent algorithms to work with directed networks and row stochastic matrices instead of column or doubly stochastic ones. Based on a regularity assumption, we then developed unified analyses for convergence and convergence rate of a distributed projected subgradient method that can be applied to both unconstrained problems or constrained ones with nonidentical (and possibly unbounded) local constraint sets. We also introduced another algorithm that also uses the rescaling technique above but converges linearly under a stronger assumption on the local objective cost functions.

7.2 Directions for Future Work

In this dissertation, we only considered discrete-time models for consensus, prediction and distributed optimization. However, there are a wide range of applications where continuous-time models are appropriate. Thus, development of similar results for the continuous-time case will be useful.

Networks with Leaders: Use of stability conditions developed for system (2.31) to study consensus conditions in the case of a leaderless network is worth exploring to reduce the gap between necessary and sufficient conditions for consensus. Note that in the latter, we may regard any agent, e.g., agent N , as a “leader” with $u(t) = \sum_{j \in \mathcal{N}_N} w_{Nj}(t) (x_j(t) - x_N(t))$.

It would also be interesting to investigate and design consensus protocols for the case where there are multiple persistent leaders with time-varying states and/or malicious agents. The intuition is that if the convex hull of all the leaders’ states

shrinks overtime to a point and the effects of malicious agents are non-persistent, then it is still possible to reach consensus asymptotically.

Extension of the consensus results developed in Chapter 2 to coordination and synchronization of multi-agent systems is also of interest.

The convexity and supermodularity results established in Chapter 3 find applications in various important problems: including sensor/actuator placement for observability/controllability in consensus networks. Moreover, in the case of 2 competing leaders, it is also important to study the game played by the 2 leaders on the network, assuming that each has a limited budget (e.g., number of direct followers).

Consensus Prediction: Besides resolving the validity of the main conjecture for the case of multiple observed nodes, which requires further rigorous analysis other than the argument presented in Remark 4.3.6, there are numerous problems worth exploring in this topic, including the following:

Coping with Noise and Delays: First, if the communication delays are fixed, then our results can be applied in a fairly straightforward manner as the network is still a linear time-invariant system. The case of time-varying delays remains difficult. Second, in the presence of observation or communication noises, exact consensus prediction in finite time is impossible. Since communication noise can derail consensus, observation noise is more relevant to the current topic, in which case we need to estimate the joint minimal polynomial and predict trend of the network states and/or range of the consensus value. In this connection, the (partial) realization theory [73–75, 159] and system identification techniques [160] might be

brought into bear [161, 162].

Network Monitoring for Misbehavior: As we have seen, for a network of agents whose dynamics follow the time-invariant DeGroot model, it is possible to predict the future behavior of the observed nodes as well as consensus value by using the minimal polynomials of these nodes, which in turn can be computed from observation data. This allows the observer to detect certain changes in the dynamics of a set of nodes, dubbed *misbehavior*, which may be caused by faults or attacks. In the case where only approximations of these minimal polynomials are available, possibly due to corrupted or noisy observations, one can still expect to capture the trend of the network response using these approximate polynomials. Consequently, certain types of faults in the network agents' dynamics may still be detected by the observer. Thus, characterizing misbehavior detectable from local observation is an interesting and important direction to pursue. This has a close connection to the topic of distributed fault detection and identification in the literature [163–166].

Distributed Optimization: First, since our distributed algorithm FADO developed in Chapter 5 behaves in a similar manner as the centralized gradient method, we can apply acceleration techniques by Nesterov [117] and others to FADO in order to achieve better convergence rates. The problem of designing the weight matrix W for a given network topology so as to achieve the smallest possible κ_{\min} is also of interest as another way of speeding up algorithms we presented in this chapter. Second, extending our algorithms in Chapter 6 to the case of switching communication graphs is also worth exploring. Finally, in all these algorithms, communication

noise and delays were not considered. Therefore, these issues deserve more research as well as attention for applications in practice.

Appendix A: Omitted Proofs

A.1 Known Matrix Results

Theorem A.1.1. ([95, Thm. 8.3.1]) *If $A \in \mathbb{R}_+^{N \times N}$, then $\rho(A)$ is an eigenvalue of A and there exists $\mathbf{x} \in \mathbb{R}_+^N \setminus \{\mathbf{0}\}$ such that $A\mathbf{x} = \rho(A)\mathbf{x}$.*

Theorem A.1.2. ([95, Thm. 8.1.18]) *Let $A, B \in \mathbb{R}^{N \times N}$. If $|A| \leq B$, then $\rho(A) \leq \rho(|A|) \leq \rho(B)$.*

Lemma A.1.3. ([167]) *Let $P \in \mathbb{R}^{N \times N}$ be the inverse of a nonsingular M -matrix. Then $P \geq 0$ and*

$$P_{jk} \geq P_{ji}P_{ii}^{-1}P_{ik}, \quad \forall i, j, k = 1, \dots, N.$$

Theorem A.1.4. (Woodbury Matrix Identity [168, p. 258]) *Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, $C \in \mathbb{R}^{r \times r}$, $D \in \mathbb{R}^{r \times n}$. Then the following holds whenever any involved inverse exists:*

$$(A - BC^{-1}D)^{-1} = A^{-1} + A^{-1}B(C - DA^{-1}B)^{-1}DA^{-1} \quad (\text{A.1})$$

Lemma A.1.5. ([169]) *Let $L \in \mathbb{R}^{N \times N}$ be a Laplacian matrix. Suppose that 0 be a simple eigenvalue of L . Let \mathbf{z} denote the left eigenvector associated with this*

eigenvalue and let L^\dagger be the pseudo-inverse of L . Then

$$\mathbf{1}^T L^\dagger = \mathbf{0}, \quad L^\dagger \mathbf{z} = \mathbf{0}, \quad L^\dagger L = I - \frac{1}{N} \mathbf{1} \mathbf{1}^T, \quad LL^\dagger = I - \frac{1}{\|\mathbf{z}\|^2} \mathbf{z} \mathbf{z}^T.$$

Lemma A.1.6. ([169, 170]) *Let $\mathbf{d}, \mathbf{e} \in \mathbb{R}^N$. The Moore-Penrose pseudoinverse of the rank-1 update of a matrix $F \in \mathbb{R}^{N \times N}$ is given by*

$$(F + \mathbf{e} \mathbf{d}^T)^\dagger = F^\dagger + G$$

where

$$G = -\frac{1}{\|\mathbf{w}\|^2} \mathbf{v} \mathbf{w}^T - \frac{1}{\|\mathbf{m}\|^2} \mathbf{m} \mathbf{h}^T + \frac{1 + \mathbf{d}^T F^\dagger \mathbf{e}}{\|\mathbf{m}\|^2 \|\mathbf{w}\|^2} \mathbf{m} \mathbf{w}^T$$

and $\mathbf{v} = F^\dagger \mathbf{e}$, $\mathbf{h} = (F^\dagger)^T \mathbf{d}$, $\mathbf{w} = (I - FF^\dagger) \mathbf{e}$ and $\mathbf{m} = (I - F^\dagger F) \mathbf{d}$.

A.2 Omitted Proofs in Chapter 3

A.2.1 Proof of Theorem 3.3.1

Suppose $\mathcal{K} = \{k\} \subset \mathcal{V}$, then we have

$$J_{\{k\}}^{(1)} = \mathbf{b}^T (L + \alpha_k \mathbf{e}_k \mathbf{e}_k^T)^{-1} |\boldsymbol{\xi}_0|$$

Now applying Lemma A.1.6 (cf. Appendix A.1) with $F = L$, $\mathbf{e} = \mathbf{e}_k$ and $\mathbf{d} = \alpha_k \mathbf{e}_k$ yields

$$(L + \alpha_k \mathbf{e}_k \mathbf{e}_k^T)^{-1} = L^\dagger + G$$

where

$$\begin{aligned}
G &= -\frac{1}{\|\mathbf{w}\|^2} \mathbf{v} \mathbf{w}^T - \frac{1}{\|\mathbf{m}\|^2} \mathbf{m} \mathbf{h}^T + \frac{1 + \mathbf{d}^T L^\dagger \mathbf{e}}{\|\mathbf{m}\|^2 \|\mathbf{w}\|^2} \mathbf{m} \mathbf{w}^T \\
\mathbf{w} &= (I - LL^\dagger) \mathbf{e}_k = \frac{1}{\|\boldsymbol{\pi}\|^2} \boldsymbol{\pi} \boldsymbol{\pi}^T \mathbf{e}_k = \frac{\pi_k}{\|\boldsymbol{\pi}\|^2} \boldsymbol{\pi} \\
\mathbf{m} &= (I - L^\dagger L) \mathbf{d} = \frac{\alpha_k}{N} \mathbf{1} \mathbf{1}^T \mathbf{e}_k = \frac{\alpha_k}{N} \mathbf{1} \\
\mathbf{v} &= L^\dagger \mathbf{e}_k \\
\mathbf{h} &= (L^\dagger)^T \alpha_k \mathbf{e}_k
\end{aligned}$$

Thus,

$$\begin{aligned}
G &= -\frac{1}{\frac{\pi_k^2}{\|\boldsymbol{\pi}\|^2}} L^\dagger \mathbf{e}_k \frac{\pi_k}{\|\boldsymbol{\pi}\|^2} \boldsymbol{\pi}^T - \frac{1}{\frac{\alpha_k^2}{N}} \alpha_k \mathbf{1} \alpha_k \mathbf{e}_k^T L^\dagger + \frac{1 + \alpha_k \mathbf{e}_k^T L^\dagger \mathbf{e}_k}{\frac{\alpha_k^2}{N} \frac{\pi_k^2}{\|\boldsymbol{\pi}\|^2}} \frac{\alpha_k}{N} \mathbf{1} \frac{\pi_k}{\|\boldsymbol{\pi}\|^2} \boldsymbol{\pi}^T \\
&= -\frac{1}{\pi_k} L^\dagger \mathbf{e}_k \boldsymbol{\pi}^T - \mathbf{1} \mathbf{e}_k^T L^\dagger + \frac{1 + \alpha_k L_{kk}^\dagger}{\alpha_k \pi_k} \mathbf{1} \boldsymbol{\pi}^T \tag{A.2}
\end{aligned}$$

Note also that $\mathbf{b}^T \mathbf{1} = 1$. Then we have

$$J_{\{k\}}^{(1)} = (\mathbf{b}^T L^\dagger - L^{\dagger(k)}) |\boldsymbol{\xi}_0| + (\alpha_k^{-1} + L_{kk}^\dagger - \mathbf{b}^T L_k^\dagger) \frac{\boldsymbol{\pi}^T |\boldsymbol{\xi}_0|}{\pi_k}.$$

Moreover, if $\mathbf{b} = \mathbf{1}/N$, then by Lemma A.1.5 (cf. Appendix A.1) we have $\mathbf{b}^T L^\dagger = \mathbf{0}^T$.

Hence, (3.19) follows immediately. \square

A.2.2 Proof of Theorem 3.3.2

By using Woodbury identity (A.1) (cf. Appendix A.1-Theorem A.1.4) and recalling that $P = L_\beta^{-1}$, we have

$$J_{(k)}^{(2)} = \mathbf{b}^T (L_\beta + \alpha_k \mathbf{e}_k \mathbf{e}_k^T)^{-1} \boldsymbol{\beta} = \mathbf{b}^T \left(P - \frac{P \mathbf{e}_k \mathbf{e}_k^T P}{\alpha_k^{-1} + \mathbf{e}_k^T P \mathbf{e}_k} \right) \boldsymbol{\beta}. \tag{A.3}$$

Now since $L\mathbf{1} = \mathbf{0}$, we have

$$L_\beta \mathbf{1} = L\mathbf{1} + \text{diag}(\boldsymbol{\beta})\mathbf{1} = \boldsymbol{\beta}.$$

Left-multiplying both sides with P yields $P\boldsymbol{\beta} = \mathbf{1}$. It remains to use this relation to simplify [A.3](#). □

A.2.3 Proof of Theorem [3.3.4](#)

Denote $P = (L + \alpha_i \mathbf{e}_i \mathbf{e}_i^T)^{-1}$. By Woodbury matrix identity ([A.1](#)) we have

$$\begin{aligned} \mathbf{p}_{ij}^T &= \frac{\mathbf{1}^T}{N} (L + \alpha_i \mathbf{e}_i \mathbf{e}_i^T + \alpha_j \mathbf{e}_j \mathbf{e}_j^T)^{-1} = \frac{\mathbf{1}^T}{N} \left(P - \frac{P \mathbf{e}_j \mathbf{e}_j^T P}{\alpha_j^{-1} + \mathbf{e}_j^T P \mathbf{e}_j} \right) \\ &= \mathbf{p}_i^T \left(I - \frac{\mathbf{e}_j \mathbf{e}_j^T P}{\alpha_j^{-1} + \mathbf{e}_j^T P \mathbf{e}_j} \right) \end{aligned} \quad (\text{A.4})$$

where we have used $\mathbf{p}_i^T = \frac{1}{N} \mathbf{1}^T P$; see Theorem [3.3.1](#). Next, by ([A.2](#)),

$$P = (L + \alpha_i \mathbf{e}_i \mathbf{e}_i^T)^{-1} = L^\dagger - \frac{1}{\pi_i} L^\dagger \mathbf{e}_i \boldsymbol{\pi}^T - \mathbf{1} \mathbf{e}_i^T L^\dagger + \frac{1 + \alpha_i \mathbf{e}_i^T L^\dagger \mathbf{e}_i}{\alpha_i \pi_i} \mathbf{1} \boldsymbol{\pi}^T$$

Then

$$\mathbf{e}_j^T P = L^{\dagger(j)} - \frac{L_{ji}^\dagger}{\pi_i} \boldsymbol{\pi}^T - L^{\dagger(i)} + \frac{1 + \alpha_i L_{ii}^\dagger}{\alpha_i \pi_i} \boldsymbol{\pi}^T = L^{\dagger(j)} - L^{\dagger(i)} + (\gamma_{ii} + \gamma_{ji}) \boldsymbol{\pi}^T$$

As a result, we have

$$\frac{1}{\alpha_j} + \mathbf{e}_j^T P \mathbf{e}_j = \frac{1}{\alpha_j} + L_{jj}^\dagger - L_{ij}^\dagger + (\gamma_{ii} + \gamma_{ji}) u_j = (\gamma_{jj} + \gamma_{ij} + \gamma_{ii} + \gamma_{ji}) \pi_j$$

Substituting this relation into (A.4) yields

$$\begin{aligned}
\mathbf{p}_{ij}^T &= \mathbf{p}_i^T - \frac{\mathbf{p}_i^T \mathbf{e}_j \mathbf{e}_j^T P}{\alpha_j^{-1} + \mathbf{e}_j^T P \mathbf{e}_j} = \mathbf{p}_i^T - \frac{\mathbf{p}_i^T \mathbf{e}_j}{\pi_j \sum \gamma_{ij}} \mathbf{e}_j^T P \\
&= \mathbf{p}_i^T - \frac{(\gamma_{ii} + \gamma_{ij}) \pi_j}{\pi_j \sum \gamma_{ij}} (L^{\dagger(j)} + \gamma_{ji} \boldsymbol{\pi}^T + \mathbf{p}_i^T) \\
&= \mathbf{p}_i^T - \frac{\gamma_{ii} + \gamma_{ij}}{\sum \gamma_{ij}} (\gamma_{jj} \boldsymbol{\pi}^T - \mathbf{p}_j^T + \gamma_{ji} \boldsymbol{\pi}^T + \mathbf{p}_i^T) \\
&= \frac{\gamma_{jj} + \gamma_{ji}}{\sum \gamma_{ij}} \mathbf{p}_i^T + \frac{\gamma_{ii} + \gamma_{ij}}{\sum \gamma_{ij}} \mathbf{p}_j^T - \frac{(\gamma_{ii} + \gamma_{ij})(\gamma_{jj} + \gamma_{ji})}{\sum \gamma_{ij}} \boldsymbol{\pi}^T \\
&= \frac{(\gamma_{ii} + \gamma_{ij})(\gamma_{jj} + \gamma_{ji})}{\sum \gamma_{ij}} \boldsymbol{\pi}^T - \frac{\gamma_{jj} + \gamma_{ji}}{\sum \gamma_{ij}} L^{\dagger(i)} - \frac{\gamma_{ii} + \gamma_{ij}}{\sum \gamma_{ij}} L^{\dagger(j)},
\end{aligned}$$

where the third to last and the last equalities follow from the relation $\mathbf{p}_i^T = \gamma_{ii} \boldsymbol{\pi}^T - L^{\dagger(i)}$ (cf. Theorem 3.3.1). This completes the proof. \square

A.2.4 Proof of Lemma 3.5.3

First, we show that $(L_{\beta} + \Gamma_{\mathcal{S}})^{-1}$ is nonincreasing in \mathcal{S} . Let $D_{\mathcal{S}} = \text{diag}(W\mathbf{1} + \boldsymbol{\beta} + \boldsymbol{\alpha}_{\mathcal{S}})$ and note that $\rho(D_{\mathcal{S}}^{-1}W) < 1$ (cf. Lemma 3.2.4). By the absolutely convergent Neumann series $(I - D_{\mathcal{S}}^{-1}W)^{-1} = \sum_{i=0}^{\infty} (D_{\mathcal{S}}^{-1}W)^i$. Thus we have

$$(L_{\beta} + \Gamma_{\mathcal{S}})^{-1} = (D_{\mathcal{S}} - W)^{-1} = \sum_{i \geq 0} (D_{\mathcal{S}}^{-1}W)^i D_{\mathcal{S}}^{-1} \quad (\text{A.5})$$

which is clearly nonnegative. Moreover, for any $\mathcal{T} \subseteq \mathcal{V}$ such that $\mathcal{T} \supseteq \mathcal{S}$, we have $\mathbf{0}_{N \times N} \leq D_{\mathcal{T}}^{-1} \leq D_{\mathcal{S}}^{-1}$, which together with (A.5) implies that $(L_{\beta} + \Gamma_{\mathcal{T}})^{-1} \leq (L_{\beta} + \Gamma_{\mathcal{S}})^{-1}$.

Alternatively, we can also use the fact that $f(\mathbf{y}) = \mathbf{b}^T (L_{\beta} + \text{diag}(\mathbf{y} \circ \boldsymbol{\alpha}))^{-1} \mathbf{c}$ is a non-increasing function on Ω for any $\mathbf{b}, \mathbf{c} \in \mathbb{R}_+^N$ (cf. Theorem 3.4.3) to conclude the monotonicity of $(L_{\beta} + \Gamma_{\mathcal{S}})^{-1}$. This proves the second inequality in (3.45).

We now prove the first inequality in (3.45), that is, for any $v, k \in \mathcal{V} \setminus \mathcal{S}$

$$(L_\beta + \Gamma_{\mathcal{S}})^{-1} - (L_\beta + \Gamma_{\mathcal{S} \cup \{v\}})^{-1} \geq (L_\beta + \Gamma_{\mathcal{S} \cup \{k\}})^{-1} - (L_\beta + \Gamma_{\mathcal{S} \cup \{k, v\}})^{-1} \quad (\text{A.6})$$

Let $P := (L_\beta + \Gamma_{\mathcal{S}})^{-1}$ and $Q := (L_\beta + \Gamma_{\mathcal{S} \cup \{k\}})^{-1}$. By Woodbury identity (A.1), it can be shown that

$$\begin{aligned} (L_\beta + \Gamma_{\mathcal{S} \cup \{v\}})^{-1} &= P - P_{(v)}P^{(v)}(\alpha_v^{-1} + P_{vv})^{-1}, \\ (L_\beta + \Gamma_{\mathcal{S} \cup \{k, v\}})^{-1} &= Q - Q_{(v)}Q^{(v)}(\alpha_v^{-1} + Q_{vv})^{-1}. \end{aligned}$$

Thus, (A.6) is equivalent to the following matrix inequality

$$P_{(v)}P^{(v)}(\alpha_v^{-1} + P_{vv})^{-1} \geq Q_{(v)}Q^{(v)}(\alpha_v^{-1} + Q_{vv})^{-1}.$$

It suffices to show that this inequality holds element-wise, i.e.,

$$\frac{P_{iv}P_{vj}}{\alpha_v^{-1} + P_{vv}} \geq \frac{Q_{iv}Q_{vj}}{\alpha_v^{-1} + Q_{vv}}, \quad \forall i, j \in \mathcal{V}. \quad (\text{A.7})$$

Note again by Woodbury identity that

$$Q = (L_\beta + \Gamma_{\mathcal{S} \cup \{k\}})^{-1} = P - P_{(k)}P^{(k)}(\alpha_k^{-1} + P_{kk})^{-1},$$

i.e., $Q_{ij} = P_{ij} - P_{ik}P_{kj}/(\alpha_k^{-1} + P_{kk})$, $\forall i, j \in \mathcal{V}$. Therefore, we have (A.7) is equivalent

to

$$\frac{\alpha_v^{-1} + Q_{vv}}{\alpha_v^{-1} + P_{vv}} P_{iv}P_{vj} \geq \left(P_{iv} - \frac{P_{ik}P_{kv}}{\alpha_k^{-1} + P_{kk}} \right) \left(P_{vj} - \frac{P_{vk}P_{kj}}{\alpha_k^{-1} + P_{kk}} \right)$$

or, by rearranging terms,

$$\frac{P_{vk}P_{kv}P_{iv}P_{vj}}{(\alpha_v^{-1} + P_{vv})} + \frac{P_{ik}P_{kv}P_{vk}P_{kj}}{(\alpha_k^{-1} + P_{kk})} \leq P_{ik}P_{kv}P_{vj} + P_{iv}P_{vk}P_{kj}. \quad (\text{A.8})$$

We now show that (A.8) holds. To this end, first note that P is the inverse of a nonsingular M-matrix. Thus, by Lemma A.1.3 and the fact that $\alpha_v^{-1} \geq 0$, we have

$$P_{ik} \geq P_{iv}P_{vk}/P_{vv} \geq P_{iv}P_{vk}/(\alpha_v^{-1} + P_{vv}).$$

Next, multiplying both sides of the above inequality with $P_{kv}P_{kj} \geq 0$ yields

$$P_{vk}P_{kv}P_{iv}P_{vj}/(\alpha_v^{-1} + P_{vv}) \leq P_{ik}P_{kv}P_{vj}. \quad (\text{A.9})$$

Similarly we have

$$P_{ik}P_{kv}P_{vk}P_{kj}/(\alpha_k^{-1} + P_{kk}) \leq P_{iv}P_{vk}P_{kj}. \quad (\text{A.10})$$

Finally, adding (A.10) and (A.9) together results in (A.8), which then completes the proof. \square

A.2.5 Proof of Lemma 3.5.5

Let $\phi = f \circ F$. We need to show that for any $\mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{V}$

$$\phi(\mathcal{S}) + \phi(\mathcal{T}) \leq \phi(\mathcal{S} \cup \mathcal{T}) + \phi(\mathcal{S} \cap \mathcal{T}). \quad (\text{A.11})$$

First, since F is decreasing, we have

$$F(\mathcal{S} \cup \mathcal{T}) \leq F(\mathcal{S}), F(\mathcal{T}) \leq F(\mathcal{S} \cap \mathcal{T}). \quad (\text{A.12})$$

As a result, $\phi(\mathcal{S} \cup \mathcal{T}) = f(F(\mathcal{S} \cup \mathcal{T})) \leq f(F(\mathcal{S} \cap \mathcal{T})) = \phi(\mathcal{S} \cap \mathcal{T})$ since f is increasing.

This proves that ϕ is nonincreasing.

Next, we have that there exist $a_1, a_2 \in [0, 1]$ such that

$$F(\mathcal{S}) = a_1F(\mathcal{S} \cup \mathcal{T}) + (1 - a_1)F(\mathcal{S} \cap \mathcal{T}) \quad (\text{A.13})$$

$$F(\mathcal{T}) = a_1F(\mathcal{S} \cup \mathcal{T}) + (1 - a_2)F(\mathcal{S} \cap \mathcal{T}). \quad (\text{A.14})$$

Adding side-by-side of the above equations gives

$$F(\mathcal{S}) + F(\mathcal{T}) = (a_1 + a_2)F(\mathcal{S} \cup \mathcal{T}) + (2 - a_1 - a_2)F(\mathcal{S} \cap \mathcal{T}),$$

whose left side is less than $F(\mathcal{S} \cup \mathcal{T}) + F(\mathcal{S} \cap \mathcal{T})$ by supermodularity property of F . Then we have

$$(a_1 + a_2)F(\mathcal{S} \cup \mathcal{T}) + (2 - a_1 - a_2)F(\mathcal{S} \cap \mathcal{T}) \leq F(\mathcal{S} \cup \mathcal{T}) + F(\mathcal{S} \cap \mathcal{T})$$

or, by rearranging terms,

$$(1 - a_1 - a_2)(F(\mathcal{S} \cap \mathcal{T}) - F(\mathcal{S} \cup \mathcal{T})) \leq \mathbf{0}_{N \times N},$$

from which, together with (A.12), we conclude that $a_1 + a_2 \geq 1$. Now using convexity of f and (A.13) we have

$$\begin{aligned} \phi(\mathcal{S}) &= f(F(\mathcal{S})) = f(a_1 F(\mathcal{S} \cup \mathcal{T}) + (1 - a_1)F(\mathcal{S} \cap \mathcal{T})) \\ &\leq a_1 f(F(\mathcal{S} \cup \mathcal{T})) + (1 - a_1) f(F(\mathcal{S} \cap \mathcal{T})) \\ &= a_1 \phi(\mathcal{S} \cup \mathcal{T}) + (1 - a_1) \phi(\mathcal{S} \cap \mathcal{T}). \end{aligned}$$

Similarly, convexity of f and (A.14) imply

$$\phi(\mathcal{T}) \leq a_2 \phi(\mathcal{S} \cup \mathcal{T}) + (1 - a_2) \phi(\mathcal{S} \cap \mathcal{T}).$$

Adding two equations above side by side yields

$$\begin{aligned} \phi(\mathcal{S}) + \phi(\mathcal{T}) &\leq (a_1 + a_2) \phi(\mathcal{S} \cup \mathcal{T}) + (2 - a_1 - a_2) \phi(\mathcal{S} \cap \mathcal{T}) \\ &= \phi(\mathcal{S} \cup \mathcal{T}) + \phi(\mathcal{S} \cap \mathcal{T}) + (a_1 + a_2 - 1) (\phi(\mathcal{S} \cup \mathcal{T}) - \phi(\mathcal{S} \cap \mathcal{T})) \\ &\leq \phi(\mathcal{S} \cup \mathcal{T}) + \phi(\mathcal{S} \cap \mathcal{T}), \end{aligned}$$

where in the last inequality we have used the facts that $a_1 + a_2 \geq 1$ and that ϕ is nonincreasing. Thus, (A.11) is proved. \square

A.3 Omitted Proofs in Chapter 5

A.3.1 Proof of Theorem 5.3.5

For $t \geq 0$, let

$$\mathbf{s}(t) := [s_1(t), \dots, s_N(t)]^T, \quad \bar{s}(t) := \boldsymbol{\pi}^T \mathbf{s}(t), \quad \mathbf{g}_s(t) := [g_1(s_1(t)), \dots, g_N(s_N(t))]^T.$$

It follows from Theorem 5.3.4 that

$$\bar{s}(t) = s_i(t), \forall i \in \mathcal{V}, \forall t \geq \kappa. \quad (\text{A.15})$$

Thus, for any $k \geq 0$, we have

$$\begin{aligned} \bar{s}((k+1)\kappa) &= s_i((k+1)\kappa) \stackrel{(5.18)}{=} \boldsymbol{\pi}^T \mathbf{x}(k\kappa) \stackrel{(5.14a)}{=} \boldsymbol{\pi}^T (\mathbf{s}(k\kappa) - \gamma_k \mathbf{g}_s(k\kappa)), \\ &= \bar{s}(k\kappa) - \gamma_k \boldsymbol{\pi}^T \mathbf{g}_s(k\kappa). \end{aligned} \quad (\text{A.16})$$

Since W is doubly stochastic, we have $\boldsymbol{\pi} = \mathbf{1}/N$ (see, e.g., [95]). Thus we have for

any $k \geq 1$

$$N \boldsymbol{\pi}^T \mathbf{g}_s(k\kappa) \stackrel{(\text{A.15})}{=} \sum_{i=1}^N g_i(\bar{s}(k\kappa)) = g(\bar{s}(k\kappa)), \quad (\text{A.17})$$

where $g(\bar{s}(k\kappa)) \in \partial F(\bar{s}(k\kappa))$. Thus, (A.16) becomes

$$\bar{s}((k+1)\kappa) = \bar{s}(k\kappa) - \gamma_k N^{-1} g(\bar{s}(k\kappa)), \quad (\text{A.18})$$

which is the same as (5.19).

Next, it is obvious that (A.18) is the usual centralized subgradient iteration (5.3) applied to problem (5.1), where F is convex with bounded subgradient $|g| \leq \sum_{j=1}^N L_j = L_F$, by Assumption 5.3.1. Therefore, existing convergence results of the

centralized subgradient method apply; see, e.g., [117, Chap. 3]. Here, we provide analysis that suits our context to prove the main results. In particular, let

$$\bar{s}_k := \bar{s}((k+1)\kappa), \quad \bar{g}_k := g(\bar{s}(k\kappa)), \quad \bar{\gamma}_k := \gamma_k/N.$$

Then for any $x^* \in X^*$, we have

$$\begin{aligned} |\bar{s}_{k+1} - x^*|^2 &= |\bar{s}_k - x^* - \bar{\gamma}_k \bar{g}_k|^2 \\ &= |\bar{s}_k - x^*|^2 - 2\bar{\gamma}_k \bar{g}_k (\bar{s}_k - x^*) + \bar{\gamma}_k^2 \bar{g}_k^2 \\ &\leq |\bar{s}_k - x^*|^2 - 2\bar{\gamma}_k (\bar{F}_k - F^*) + \bar{\gamma}_k^2 \bar{g}_k^2 \\ &\leq |\bar{s}_1 - x^*|^2 - 2 \sum_{l=1}^k \bar{\gamma}_l (\bar{F}_l - F^*) + \sum_{l=1}^k \bar{\gamma}_l^2 \bar{g}_l^2, \end{aligned} \tag{A.19}$$

where $\bar{F}_k := F(\bar{s}(k\kappa))$, and we have used the definition of subgradient in the first inequality, and the last one follows from applying (A.19) recursively. Now, rearranging terms and using $0 \leq |\bar{s}_{k+1} - x^*|^2$ and $|g| \leq L_F$ we have

$$\sum_{l=1}^k \bar{\gamma}_l (\bar{F}_l - F^*) \leq \frac{1}{2} (|\bar{s}_1 - x^*|^2 + L_F^2 \sum_{l=1}^k \bar{\gamma}_l^2).$$

By the convexity of F , the left side is bounded below by

$$\left(\sum_{l=1}^k \bar{\gamma}_l \right) \left(\frac{\sum_{l=1}^k \bar{\gamma}_l F(\bar{s}(l\kappa))}{\sum_{l=1}^k \bar{\gamma}_l} - F^* \right) \geq \left(\sum_{l=1}^k \bar{\gamma}_l \right) \left(F(\hat{s}_k) - F^* \right)$$

Combining this and (A.22), we then have

$$F(\hat{s}_k) - F^* \leq \frac{|\bar{s}_1 - x^*|^2 + L_F^2 \sum_{l=1}^k \bar{\gamma}_l^2}{2 \sum_{l=1}^k \bar{\gamma}_l}. \tag{A.20}$$

Now we consider different choices of step size γ_k .

(i) For a constant step size $\gamma_k \equiv \gamma$, i.e., $\bar{\gamma}_k \equiv \gamma/N, \forall k \geq 1$, it follows from (A.20) that

$$F(\hat{s}_k) - F^* \leq \frac{N|\bar{s}_1 - x^*|^2}{2k\gamma} + \frac{L_F^2 \gamma}{2N}. \tag{A.21}$$

Letting $k \rightarrow \infty$ yields (5.21).

(ii) Since (A.21) holds true for any $x^* \in X^*$, $\gamma > 0$ and $k \in \mathbb{Z}_{>0}$, for any given $K \in \mathbb{Z}_{>0}$ we have

$$F(\hat{s}_K) - F^* \leq \frac{NR^2}{2K\gamma} + \frac{L_F^2\gamma}{2N}, \quad (\text{A.22})$$

Now we minimize the right hand side of (A.22) with respect to $\gamma > 0$. By application of Cauchy-Schwarz inequality $\frac{NR^2}{2K\gamma} + \frac{L_F^2\gamma}{2N} \geq \frac{RL_F}{\sqrt{K}}$, where equality holds when $\gamma = \frac{NR}{L_F\sqrt{K}}$. Thus, with this optimal step size, we have $F(\hat{s}_K) - F^* \leq \frac{RL_F}{\sqrt{K}}$.

(iii) For a non-summable but diminishing step size, it can be shown that the right hand side of (A.20) decays to 0 as $k \rightarrow \infty$; see [138] for such an argument.

Finally, consider $\gamma_k = \frac{1}{\sqrt{k}}$. It can be verified that $\sum_{l=1}^k \bar{\gamma}_l^2 \leq \frac{1+\ln(k)}{N^2}$ and $\sum_{l=1}^k \bar{\gamma}_l \geq \frac{\sqrt{k}}{2N}$, $\forall k \geq 1$. Using these bounds for (A.20), we obtain

$$F(\hat{s}_k) - F^* \leq \frac{N^2|\bar{s}_1 - x^*|^2 + L_F^2(1 + \ln(k))}{N\sqrt{k}}, \quad (\text{A.23})$$

which implies that $F(\hat{s}_k) - F^* = O(\frac{\ln(k)}{\sqrt{k}})$ as $k \rightarrow \infty$. \square

A.3.2 Proof of Theorem 5.3.12

Following the same line of proof as in Theorem 5.3.5 (see Appendix A.3.1), it can be shown that

$$\bar{s}((k+1)\kappa) = \bar{s}(k\kappa) - \gamma N^{-1} \nabla F(\bar{s}(k\kappa)), \quad (\text{A.24})$$

where $\bar{s}(t) = \frac{1}{N} \sum_{j=1}^N s_j(t)$. Clearly, (A.24) is the standard centralized gradient descent method. Thus, by [117, Thm. 2.1.14] we have for any $\gamma \in (0, \frac{2N}{L_{\nabla F}})$

$$F(\bar{s}(k\kappa)) - F^* \leq \frac{a_1}{a_2 + ka_3} \quad (\text{A.25})$$

where $a_1 = (F(\bar{s}(\kappa)) - F^*)(\bar{s}(\kappa) - x^*)^2$, $a_2 = (\bar{s}(\kappa) - x^*)^2$, and $a_3 = (F(\bar{s}(\kappa)) - F^*)(1 - \frac{L_{\nabla F} h}{2})h$ with $h = \frac{\gamma}{N}$. As a result,

$$F(s_i(k\kappa)) - F^* = O(1/k), \quad \text{as } k \rightarrow \infty. \quad (\text{A.26})$$

Finally, for each $t \geq \kappa$, there exist positive integers $k \geq 1$ and $l \in [0, \kappa - 1]$ such that $t = k\kappa + l$. Then by (5.13b),

$$F(s_i(t)) - F^* = F(s_i(k\kappa)) - F^* \stackrel{(\text{A.25})}{\leq} \frac{a_1}{a_2 + ka_3} \leq \frac{a_1 \kappa}{a_2 \kappa + ta_3}. \quad (\text{A.27})$$

The last inequality holds for sufficiently large t since $a_3 > 0$, $l \in [0, \kappa]$. Thus, $F(s_i(t)) - F^* = O(\frac{\kappa}{t})$ as $t \rightarrow \infty$. \square

A.3.3 Proof of Theorem 5.3.14

First, note that (A.24) still holds in this case, i.e., $\bar{s}((k+1)\kappa) = \bar{s}(k\kappa) - \frac{\gamma}{N} \nabla F(\bar{s}(k\kappa))$, where $\bar{s}(t) = \frac{1}{N} \sum_{j=1}^N s_j(t)$. Applying [117, Thm. 2.1.15] to this iteration (i.e., (A.24)) yields

$$\begin{aligned} |\bar{s}(k\kappa) - x^*|^2 &\leq \beta^{k-1} |\bar{s}(\kappa) - x^*|^2, \\ 2(F(\bar{s}(k\kappa)) - F^*) &\leq L_{\nabla F} \beta^{k-1} |\bar{s}(\kappa) - x^*|^2. \end{aligned}$$

Then (5.30) and (5.31) follows immediately since $s_i(t) = \bar{s}(t), \forall t \geq \kappa, \forall i \in \mathcal{V}$ (cf. Theorem 5.3.4).

Next, for each $t \geq \kappa$, there exist positive integers $k \geq 1$ and $l \in [0, \kappa - 1]$ such that $t = k\kappa + l$. Then, we have

$$\begin{aligned} |s_i(t) - x^*| &\stackrel{(5.13b)}{=} |s_i(k\kappa) - x^*| \stackrel{(5.30)}{\leq} C \beta^{\frac{k-1}{2}}, \quad \text{with } C := |\bar{s}(\kappa) - x^*| \\ &= C (\beta^{\frac{1}{2\kappa}})^{t-l-\kappa} \leq C \beta^{\frac{-1}{2}} (\beta^{\frac{1}{2\kappa}})^t, \end{aligned} \quad (\text{A.28})$$

where the last inequality holds since $l < \kappa$. Thus, $s_i(t) \rightarrow x^*$ linearly at rate $\beta^{\frac{1}{2\kappa}}$. Similarly, by using (5.31), it can be shown that $F(s_i(t)) \rightarrow F^*$ linearly at rate $\beta^{\frac{1}{\kappa}}$. \square

A.3.4 Proof of Extension to Row Stochastic Weight Matrix

First notice that the proof of Theorem 5.3.4 does not make use of (5.14a). Thus, (5.15) still holds for (5.36)-(5.37). Following the proof of Theorem 5.3.5 (see Appendix A.3.1), we let $\bar{s}(t) := s_i(t), t \geq \kappa$ and $\mathbf{g}_s(t) := [g_1(s_1(t)), \dots, g_N(s_N(t))]^T$. Recall that $\Phi = \mathbf{1}\boldsymbol{\pi}^T$. We then have

$$\begin{aligned} \bar{s}((k+1)\kappa) &= s_i((k+1)\kappa) \stackrel{(5.18)}{=} \boldsymbol{\pi}^T \mathbf{x}(k\kappa) \\ &\stackrel{(5.37)}{=} \boldsymbol{\pi}^T \left(\mathbf{s}(k\kappa) - \gamma_k (N \text{diag}(\boldsymbol{\pi}))^{-1} \mathbf{g}_s(k\kappa) \right) \\ &= \bar{s}(k\kappa) - \gamma_k N^{-1} \sum_{i=1}^N g_i(s_i(k\kappa)). \end{aligned} \tag{A.29}$$

Since $s_i(k\kappa) = \bar{s}(k\kappa), \forall i \in \mathcal{V}$ (cf. Theorem 5.3.4), (A.29) becomes

$$\bar{s}((k+1)\kappa) = \bar{s}(k\kappa) - \gamma_k N^{-1} g(\bar{s}(k\kappa)), \tag{A.30}$$

where $g(\bar{s}(k\kappa)) = \sum g_i(\bar{s}(k\kappa)) \in \partial F(\bar{s}(k\kappa))$. Now (A.30) is the same as (5.19). Therefore, the same conclusions in Theorems 5.3.5-5.3.14 hold for the convergence of the modified algorithm. \square

A.3.5 Proof of Theorem 5.5.1

First, note that the condition $\epsilon(|\mathcal{N}_i| - 1) \leq 1$ is to ensure that W is a nonnegative matrix, hence a valid weight matrix.

Now we prove that $\text{diam}(\mathcal{G}) + 1 = \deg(q_i), \forall i \in \mathcal{V}$. Define

$$\Omega^{(i)} := [\mathbf{e}_i \ L(\mathcal{G})^T \mathbf{e}_i \ \dots \ (L(\mathcal{G})^{N-1})^T \mathbf{e}_i]^T.$$

By [72, Prop. 1] we have $\deg(q_i) = D_i + 1 = \text{rank}(\Omega^{(i)})$. Next, by application of [132, Prop. 5], if the graph \mathcal{G} is distance regular, then $\text{diam}(\mathcal{G}) + 1 = \text{rank}(\Sigma^{(i)})$ where $\Sigma^{(i)}$ is the controllability matrix of the pair $(L(\mathcal{G}), \mathbf{e}_i)$, computed as

$$\Sigma^{(i)} = [\mathbf{e}_i \ L(\mathcal{G})\mathbf{e}_i \ \dots \ L(\mathcal{G})^{N-1}\mathbf{e}_i] = [\mathbf{e}_i^T \ \mathbf{e}_i^T L(\mathcal{G}) \ \dots \ \mathbf{e}_i^T (L(\mathcal{G})^{N-1})]^T = (\Omega^{(i)})^T$$

Here, the second equality follows from the symmetry of $L(\mathcal{G})$ (recalling that \mathcal{G} is undirected). Thus, $\text{diam}(\mathcal{G}) + 1 = \text{rank}((\Omega^{(i)})^T) = \text{rank}(\Omega^{(i)}) = \deg(q_i)$. \square

A.3.6 Proof of Lemma 5.4.3

For a given matrix $A \in \mathbb{R}^{N \times N}$, let q_A denote its minimal polynomial (i.e., the monic polynomial of minimum degree such that $q_A(A) = 0$). It follows from the Cayley-Hamilton theorem that $\deg(q_A) \leq N$.

Let J denote the Jordan canonical form of $W - \gamma B$, i.e., there exists a non-singular matrix $S \in \mathbb{R}^{N \times N}$ such that $W - \gamma B = SJS^{-1}$. Since similar matrices have the same minimal polynomial ([95, Corollary 3.3.3]), we have $q_{(W-\gamma B)} = q_J$.

Moreover, it can be verified that

$$\tilde{W} = \begin{bmatrix} SJS^{-1} & -\gamma B \\ \mathbf{0}_{N \times N} & I \end{bmatrix} = \begin{bmatrix} S & \Phi_\gamma \\ \mathbf{0} & I \end{bmatrix} \underbrace{\begin{bmatrix} J & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}}_{\cong K} \begin{bmatrix} S^{-1} & -S^{-1}\Phi_\gamma \\ \mathbf{0} & I \end{bmatrix} \quad (\text{A.31})$$

where Φ_γ is defined in (5.49). Thus, K is the Jordan canonical form of \tilde{W} . Under condition (5.47), i.e., $\rho(J) < 1$, the order of the largest Jordan block of K corre-

sponding to eigenvalue 1 is equal to 1. It then follows immediately that (see e.g., [95, Thm. 3.3.6]) $q_K(\xi) = (\xi - 1)q_J(\xi)$. Consequently, we have

$$q_{\tilde{W}}(\xi) = q_K(\xi) = (\xi - 1)q_J(\xi) = (\xi - 1)q_{(W-\gamma B)}(\xi). \quad (\text{A.32})$$

Since $\tilde{q}_i|q_{\tilde{W}}$ (see Lemma 4.2.3), we obtain the following,

$$\deg(\tilde{q}_i) \leq \deg(q_{\tilde{W}}) \stackrel{(\text{A.32})}{=} 1 + \deg(q_{(W-\gamma B)}) \leq 1 + N.$$

We next show that $\tilde{q}_i(1) = 0$, which then clearly implies

$$\tilde{q}_i(\xi) = (\xi - 1) \sum_{j=0}^{\tilde{D}_i} \tilde{a}_j^{(i)} \xi^j, \quad \tilde{a}_{\tilde{D}_i}^{(i)} = 1$$

for some $\tilde{\mathbf{a}}^{(i)} \in \mathbb{R}^{\tilde{D}_i+1}$. To this end, recall that for any $i = 1, \dots, N$ and the unit vector $\mathbf{e}_i \in \mathbb{R}^N$, we have

$$\begin{aligned} \mathbf{0}_{2N}^T &= [\mathbf{e}_i^T \quad \mathbf{0}_N^T] \tilde{q}_i(\tilde{W}) \\ &\stackrel{(\text{A.31})}{=} [\mathbf{e}_i^T \quad \mathbf{0}_N^T] \begin{bmatrix} S & \Phi_\gamma \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \tilde{q}_i(J) & \mathbf{0} \\ \mathbf{0} & \tilde{q}_i(1)I \end{bmatrix} \begin{bmatrix} S^{-1} & -S^{-1}\Phi_\gamma \\ \mathbf{0} & I \end{bmatrix} \\ &= [\mathbf{e}_i^T S \tilde{q}_i(J) S^{-1} \quad \tilde{q}_i(1) \mathbf{e}_i^T \Phi_\gamma - \mathbf{e}_i^T S \tilde{q}_i(J) S^{-1} \Phi_\gamma]. \end{aligned} \quad (\text{A.33})$$

As a result,

$$\tilde{q}_i(1) \mathbf{e}_i^T \Phi_\gamma = \mathbf{0}_N^T. \quad (\text{A.34})$$

Since Φ_γ is invertible (see (5.49)), none of its rows is identical to $\mathbf{0}_N^T$. Consequently, (A.34) implies that $\tilde{q}_i(1) = 0$.

Finally, (A.32) implies that if $\rho(W - \gamma B) < 1$, then 1 is the only zero of maximum modulus of $q_{\tilde{W}}$. The proof is completed by noting that \tilde{q}_i divides $q_{\tilde{W}}$. \square

A.3.7 Proof of Theorem 5.4.4

Consider again system (5.46) and assume that (5.47) holds.

First we show that Φ_γ satisfies $\Phi_\gamma \mathbf{1} = \mathbf{1}$ and $\boldsymbol{\pi}^T B \Phi_\gamma = \boldsymbol{\pi}^T B$. Note that for any $A \in \mathbb{R}^{N \times N}$ such that $(I - A)$ is invertible, we have

$$(I - A)^{-1} - I = (I - A)^{-1}A = A(I - A)^{-1}. \quad (\text{A.35})$$

Now let $E = \gamma B$ and note that $W\mathbf{1} = \mathbf{1}$ and $\boldsymbol{\pi}^T W = \boldsymbol{\pi}^T$. Then

$$\begin{aligned} \Phi_\gamma \mathbf{1} &= [I - (W - E)]^{-1} E \mathbf{1} \\ &= [I - (W - E)]^{-1} W \mathbf{1} - [I - (W - E)]^{-1} (W - E) \mathbf{1} \\ &\stackrel{(\text{A.35})}{=} [I - (W - E)]^{-1} \mathbf{1} - \left([I - (W - E)]^{-1} - I \right) \mathbf{1} = \mathbf{1}, \end{aligned}$$

$$\begin{aligned} \boldsymbol{\pi}^T E \Phi_\gamma &= \boldsymbol{\pi}^T E [I - (W - E)]^{-1} E \\ &= \boldsymbol{\pi}^T W [I - (W - E)]^{-1} E - \boldsymbol{\pi}^T (W - E) [I - (W - E)]^{-1} E \\ &\stackrel{(\text{A.35})}{=} \boldsymbol{\pi}^T [I - (W - E)]^{-1} E - \boldsymbol{\pi}^T \left([I - (W - E)]^{-1} - I \right) E \\ &= \boldsymbol{\pi}^T E. \end{aligned}$$

Therefore, $B\boldsymbol{\pi}$ and $\mathbf{1}$ are left and right eigenvectors of Φ_γ corresponding to the eigenvalue 1, respectively.

Next, by Assumption 5.2.2 and condition (5.47), $(I - (W - \gamma B))$ is an irreducible nonsingular M-matrix. Thus, $(I - (W - \gamma B))^{-1}$ is a strictly positive matrix (see, e.g., [171]). Therefore, $\Phi_\gamma = [I - (W - \gamma B)]^{-1} \gamma B$ is also strictly positive. Thus Φ_γ is also irreducible (i.e., the graph associated with Φ_γ is strongly connected; in fact

it is complete), and 1 is a simple eigenvalue, corresponding to the spectral radius of Φ_γ . In fact, by Perron's theorem for positive matrices (see, e.g., [95, Thm 8.2.11]), 1 is the unique eigenvalue of maximum modulus of Φ_γ , and

$$\lim_{k \rightarrow \infty} \Phi_\gamma^k = \mathbf{1} \boldsymbol{\pi}^T B / (\boldsymbol{\pi}^T B \mathbf{1}).$$

Hence, Φ_γ is a valid weight matrix for consensus. Moreover, the convergence is exponential with rate $|\lambda_2(\Phi_\gamma)|$, where λ_2 is an eigenvalue of second largest in modulus. □

A.3.8 Proof of Theorem 5.4.5

By (5.52) and (5.49), we have the following, which is in the same spirit as Theorem 5.2.3

$$\left(\sum_{l=0}^{\tilde{D}_i} \tilde{a}_l^{(i)} x_i(l) \right) / \left(\sum_{l=0}^{\tilde{D}_i} \tilde{a}_l^{(i)} \right) = \mathbf{e}_i^T \Phi_\gamma \mathbf{x}(0) \quad \forall i \in \mathcal{V},$$

where $\tilde{\mathbf{a}}^{(i)} = [\tilde{a}_0^{(i)}, \dots, \tilde{a}_{\tilde{D}_i}^{(i)}]^T \in \mathbb{R}^{\tilde{D}_i+1}$ satisfies (5.51). Thus,

$$\mathbf{s}((k+1)\kappa) = \Phi_\gamma \mathbf{s}(k\kappa) = \Phi_\gamma^{k+1} \mathbf{c}. \quad (\text{A.36})$$

By Theorem 5.4.4, Φ_γ is a valid consensus matrix. Since W is doubly stochastic, we have $\boldsymbol{\pi} = \mathbf{1}/N$, and thus, $\mathbf{b} = B\mathbf{1} = [b_1, \dots, b_N]^T$ is a left Perron eigenvector of Φ_γ . Thus, by (5.55)

$$\lim_{k \rightarrow \infty} \mathbf{s}(k\kappa) = \lim_{k \rightarrow \infty} \Phi_\gamma^k \mathbf{c} = \mathbf{1} \mathbf{b}^T \mathbf{c} / (\mathbf{b}^T \mathbf{1}) = \mathbf{1} x^*. \quad (\text{A.37})$$

This and (5.53)-(5.54) imply (5.56).

Moreover, since Φ_γ^k converges exponentially as $k \rightarrow \infty$, so does $\mathbf{s}(k\kappa)$, i.e., there exist $C > 0$ and $\lambda_2 \in (0, 1)$ such that

$$\|\mathbf{s}(k\kappa) - \mathbf{1}x^*\| \leq C\lambda_2^k, \quad \forall k \geq 0.$$

Now for each $t \geq \kappa$, there exist positive integers $k \geq 1$ and $l \in [0, \kappa - 1]$ such that $t = k\kappa + l$. We then have

$$\|\mathbf{s}(t) - \mathbf{1}x^*\| \stackrel{(5.53)}{=} \|\mathbf{s}(k\kappa) - \mathbf{1}x^*\| \leq C\lambda_2^k \leq C\lambda_2^{-1}\lambda_2^{\frac{t}{\kappa}}. \quad (\text{A.38})$$

where the last inequality follows from the condition that $l \in [0, \kappa - 1]$. Therefore, $\mathbf{s}(t)$ also converges exponentially with rate $\lambda_2^{\frac{1}{\kappa}}$. This concludes the proof. \square

A.3.9 Distributed Evaluation of Global Cost Function and Algorithm Local Termination

Here we provide an augmentation to our main algorithm that allows each agent i to compute $F(\hat{s}_k)$ or $F(s_i(t))$. Besides (5.13)-(5.14), let all the agents also perform (5.24)-(5.25) (in order to compute \hat{s}_k) as well as the following for all $t \geq \kappa + 1$:

$$y_i(t) = \begin{cases} f_i(\hat{s}_k) & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} y_j(t-1) & \text{if } t \neq k\kappa \end{cases} \quad (\text{A.39a})$$

$$y_i(t) = \begin{cases} f_i(\hat{s}_k) & \text{if } t = k\kappa \\ \sum_{j \in \mathcal{N}_i} w_{ij} y_j(t-1) & \text{if } t \neq k\kappa \end{cases} \quad (\text{A.39b})$$

and

$$F_k^{(i)} = \frac{\sum_{\tau=0}^{D_i} a_\tau^{(i)} y_i(t - \kappa + \tau)}{\sum_{l=0}^{D_i} a_l^{(i)}}, \quad \text{if } t = k\kappa \quad (\text{A.40})$$

Claim: Let the assumptions of Theorem 5.3.5 hold. Then, $F(\hat{s}_k) = NF_k^{(i)}$ for any $k \geq 1$ and $\forall i \in \mathcal{V}$.

Proof of Claim: First recall from Theorem 5.3.4 that $s_i(t) = s_j(t) =: \bar{s}(t)$ for all $t \geq \kappa$ and $i, j \in \mathcal{V}$. Thus, each agent can locally find \hat{s}_k using (5.24)-(5.25). Next, by (A.39b), we have

$$\mathbf{y}(t) = W\mathbf{y}(t-1), \quad \forall t = k\kappa + 1, \dots, (k+1)\kappa - 1.$$

At time $t = (k+1)\kappa$, we have

$$\begin{aligned} & \frac{\sum_{\tau=0}^{D_i} a_\tau^{(i)} y_i(k\kappa + \tau)}{\sum_{\tau=0}^{D_i} a_\tau^{(i)}} \stackrel{(\text{Thm.5.2.3})}{=} \mathbf{e}_i^T \Phi \mathbf{y}(k\kappa) = N^{-1} \mathbf{1}^T \mathbf{y}(k\kappa) \\ & = N^{-1} \sum_{i=1}^N f_i(\hat{s}_k) = N^{-1} F(\hat{s}_k). \end{aligned} \tag{A.41}$$

where the second equality follows from (5.7) with $\boldsymbol{\pi} = \mathbf{1}/N$ and the third one from (A.39a). Therefore the claim holds. \square

As a result, each agent can compute $F(\hat{s}_k)/N$ in a distributed manner (and hence $F(\hat{s}_k)$ if N is known). Similarly, if (A.39a) is replaced by $y(t) = f_i(s_i(k\kappa))$ for $t = k\kappa$, the same argument holds for $F(s_i(k\kappa))$, that is, $F(s_i(k\kappa)) = NF_k^{(i)}$ for any $k \geq 1$ and $\forall i \in \mathcal{V}$.

Therefore, all the agents can stop at the same time if they agree to use a common stopping criterion such as (5.27), which is based on absolute convergence error. Note that other criteria of the same type can also be employed, for example, local relative convergence tolerance

$$|F_k^{(i)} - F_{k-1}^{(i)}| \leq \epsilon |F_k^{(i)}|, \tag{A.42}$$

is obviously equivalent to global relative convergence tolerance $|F(\hat{s}_k) - F(\hat{s}_{k-1})| \leq \epsilon |F(\hat{s}_k)|$ or $|F(s_i(k\kappa)) - F(s_i(k\kappa - \kappa))| \leq \epsilon |F(s_i(k\kappa))|$ if s_i is used instead.

A.4 Omitted Proofs in Chapter 6

A.4.1 Proof of Theorem 6.3.6 for Algorithm 6.2.

Recall that for Algorithm 6.2, the projection error $\phi_i(t)$ is given by (6.15). Thus, for any $\mathbf{v} \in X$, we have

$$\|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 = \left\| \sum_{j \in \mathcal{N}_i} w_{ij} \left(\mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right) - \mathbf{v} + \phi_i(t) \right\|^2.$$

Expanding the right side and using the fact that

$$\phi_i(t)^T \left(\sum_{j \in \mathcal{V}} w_{ij} \left(\mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right) - \mathbf{v} \right) \leq -\|\phi_i(t)\|^2$$

for any $\mathbf{v} \in X \subseteq X_i$ (cf. Lemma 6.3.1(a)), we obtain

$$\begin{aligned} \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq \left\| \sum_{j \in \mathcal{N}_i} w_{ij} \left(\mathbf{x}_j(t) - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right) - \mathbf{v} \right\|^2 - \|\phi_i(t)\|^2 \\ &\leq \sum_{j \in \mathcal{N}_i} w_{ij} \left\| \mathbf{x}_j(t) - \mathbf{v} - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right\|^2 - \|\phi_i(t)\|^2, \end{aligned} \quad (\text{A.43})$$

where (A.43) follows from $\sum_{j \in \mathcal{V}} w_{ij} = 1$ and Lemma 6.3.2. Hence,

$$\begin{aligned} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq \sum_{i \in \mathcal{V}} \pi_i \sum_{j \in \mathcal{N}_i} w_{ij} \left\| \mathbf{x}_j(t) - \mathbf{v} - \gamma(t) \frac{\mathbf{g}_j(t)}{z_{jj}(t)} \right\|^2 - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2 \\ &= \sum_{i \in \mathcal{V}} \pi_i \left\| \mathbf{x}_i(t) - \mathbf{v} - \gamma(t) \frac{\mathbf{g}_i(t)}{z_{ii}(t)} \right\|^2 - \sum_{i \in \mathcal{V}} \pi_i \|\phi_i(t)\|^2, \end{aligned} \quad (\text{A.44})$$

where (A.44) holds since $\boldsymbol{\pi}^T W = \boldsymbol{\pi}^T$. Expanding the first term on the right side of (A.44) yields

$$\sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 - 2\gamma(t) \sum_{i \in \mathcal{V}} \frac{\pi_i}{z_{ii}(t)} \mathbf{g}_i(t)^T (\mathbf{x}_i(t) - \mathbf{v}) + \gamma^2(t) \sum_{i \in \mathcal{V}} \frac{\pi_i}{z_{ii}^2(t)} \|\mathbf{g}_i(t)\|^2. \quad (\text{A.45})$$

We now derive upper bounds for the last two terms in (A.45). First, by (6.5), (6.11) and the fact that $\sum_{i \in \mathcal{V}} \pi_i = 1$, we have

$$\gamma^2(t) \sum_{i \in \mathcal{V}} \frac{\pi_i}{z_{ii}^2(t)} \|\mathbf{g}_i(t)\|^2 \leq \gamma^2(t) L_f^2 \eta^2. \quad (\text{A.46})$$

Second, using the facts that $\mathbf{g}_i(t) \in \partial f_i(\mathbf{x}_i(t))$ and f_i is Lipschitz continuous on $\text{conv}(\cup_{i=1}^N X_i)$ (cf. Assumption 6.2.1(b)), it can be shown that

$$\mathbf{g}_i(t)^T (\mathbf{v} - \mathbf{x}_i(t)) \leq f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t)) + L_f \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|. \quad (\text{A.47})$$

As a result, the second term in (A.45) can be bounded as follows (ignoring the factor $2\gamma(t)$):

$$\begin{aligned} \sum_{i \in \mathcal{V}} \frac{-\pi_i}{z_{ii}(t)} \mathbf{g}_i(t)^T (\mathbf{x}_i(t) - \mathbf{v}) &\stackrel{(\text{A.47})}{\leq} \sum_{i \in \mathcal{V}} \frac{\pi_i}{z_{ii}(t)} \left(f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t)) + L_f \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| \right) \\ &\stackrel{(\text{6.11})}{\leq} \sum_{i \in \mathcal{V}} \frac{\pi_i}{z_{ii}(t)} \left(f_i(\mathbf{v}) - f_i(\bar{\mathbf{x}}(t)) \right) + L_f \eta \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|, \\ &\stackrel{(\text{6.23})}{\leq} F(\mathbf{v}) - F(\bar{\mathbf{x}}(t)) + NCL_f \eta \lambda^t \|\bar{\mathbf{x}}(t) - \mathbf{v}\| \\ &\quad + L_f \eta \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\|. \end{aligned} \quad (\text{A.48})$$

Finally, returning to the argument in (A.44) and using (A.45) together with the bounds in (A.46) and (A.48), we obtain

$$\begin{aligned} \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t+1) - \mathbf{v}\|^2 &\leq \sum_{i \in \mathcal{V}} \pi_i \|\mathbf{x}_i(t) - \mathbf{v}\|^2 - 2\gamma(t) \left(F(\bar{\mathbf{x}}(t)) - F^* \right) \\ &\quad + 2\gamma(t) NCL_f \eta \lambda^t \|\bar{\mathbf{x}}(t) - \mathbf{v}\| \\ &\quad + 2\gamma(t) L_f \eta \sum_{i \in \mathcal{V}} \|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\| + \gamma^2(t) L_f^2 \eta^2 \end{aligned}$$

which is the same as (6.25). Therefore, (6.13) readily follows by the same constants D_i as in the case of Algorithm 6.1. \square

Bibliography

- [1] M. H. DeGroot, “Reaching a consensus,” *J. American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.
- [2] J. N. Tsitsiklis, “Problems in decentralized decision making and computation,” Ph.D. dissertation, MIT, 1984.
- [3] N. Friedkin and E. Johnsen, “Social influence and opinions,” *J. Mathematical Sociology*, vol. 15, pp. 193–206, 1990.
- [4] P. DeMarzo, D. Vayanos, and J. Zwiebel, “Persuasion bias, social influence, and unidimensional opinions,” *Quarterly J. Economics*, vol. 118, no. 3, pp. 909–968, 2003.
- [5] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [6] J. Fax and R. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, 2004.
- [7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Trans. Robot. Automat.*, vol. 20, no. 2, pp. 243–255, 2004.
- [8] L. Moreau, “Stability of multiagent systems with time-dependent communication links,” *IEEE Trans. Autom. Control*, vol. 50, pp. 169–182, 2005.
- [9] W. Ren, R. W. Beard, and T. W. McLain, “Coordination variables and consensus building in multiple vehicle systems,” in *Cooperative Control*. Springer, 2005, pp. 171–188.
- [10] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, 2008.

- [11] B. Golub and M. O. Jackson, “Naive learning in social networks and the wisdom of crowds,” *American Economic J.: Microeconomics*, pp. 112–149, 2010.
- [12] D. Acemoglu and A. Ozdaglar, “Opinion dynamics and learning in social networks,” *Dynamic Games and Applications*, vol. 1, no. 1, pp. 3–49, 2011.
- [13] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, “Opinion fluctuations and disagreement in social networks,” *Mathematics of Operations Research*, vol. 38, no. 1, pp. 1–27, 2013.
- [14] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, 2007.
- [15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [16] Y. Cao, W. Yu, W. Ren, and G. Chen, “An overview of recent progress in the study of distributed multi-agent coordination,” *IEEE Trans Ind. Informat.*, vol. 9, no. 1, pp. 427–438, 2013.
- [17] S. Strogatz, *Sync: The Emerging Science of Spontaneous Order*. Hyperion, 2003.
- [18] M. Ji, A. Muhammad, and M. Egerstedt, “Leader-based multi-agent coordination: Controllability and optimal control,” in *Proc. American Control Conf.*, 2006, pp. 1358–1363.
- [19] E. Yildiz, D. Acemoglu, A. E. Ozdaglar, A. Saberi, and A. Scaglione, “Discrete opinion dynamics with stubborn agents,” 2011. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.1744113>
- [20] J. Ghaderi and R. Srikant, “Opinion dynamics in social networks: a local interaction game with stubborn agents,” in *Proc. American Control Conf. IEEE*, 2013, pp. 1982–1987.
- [21] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, “Consensus-based distributed intrusion detection for multi-robot systems,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2008, pp. 120–127.
- [22] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterative strategies in the presence of malicious agents,” *IEEE Trans. Autom. Control*, no. 7, pp. 1495–1508, 2011.
- [23] H. J. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos, “Consensus of multi-agent networks in the presence of adversaries using only local information,” in *Proc. 1st Int. Conf. High Confidence Networked Systems*, 2012, pp. 1–10.

- [24] M. Ji, A. Muhammad, and M. Egerstedt, “Leader-based multi-agent coordination: Controllability and optimal control,” in *Proc. American Control Conf. IEEE*, 2006, pp. 6–pp.
- [25] F. Sorrentino, M. di Bernardo, F. Garofalo, and G. Chen, “Controllability of complex networks via pinning,” *Physical Rev. E*, vol. 75, no. 4, p. 046103, 2007.
- [26] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, “Controllability of multi-agent systems from a graph-theoretic perspective,” *SIAM J. Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.
- [27] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, “Controllability of complex networks,” *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.
- [28] G. Parlangeli and G. Notarstefano, “On the reachability and observability of path and cycle graphs,” *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 743–748, 2012.
- [29] F. Pasqualetti, S. Zampieri, and F. Bullo, “Controllability metrics, limitations and algorithms for complex networks,” *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 40–52, 2014.
- [30] A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, “Controllability and observability of network-of-networks via cartesian products,” *IEEE Trans. Autom. Control*, vol. 59, no. 10, pp. 2668–2679, 2014.
- [31] A. J. Whalen, S. N. Brennan, T. D. Sauer, and S. J. Schiff, “Observability and controllability of nonlinear networks: The role of symmetry,” *Physical Rev. X*, vol. 5, no. 1, p. 011005, 2015.
- [32] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Proc. 3rd Int. Sympo. Inf. Process. Sensor Networks*. ACM, 2004, pp. 20–27.
- [33] S. Kar and J. M. Moura, “Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise,” *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, 2009.
- [34] S. Bolognani, S. Del Favero, L. Schenato, and D. Varagnolo, “Consensus-based distributed sensor calibration and least-square parameter identification in WSNs,” *Int. J. Robust and Nonlinear Control*, vol. 20, no. 2, pp. 176–193, 2010.
- [35] L. Xiao and S. Boyd, “Optimal scaling of a gradient method for distributed resource allocation,” *J. Optim. Theory App.*, vol. 129, no. 3, pp. 469–488, 2006.

- [36] P. Di Lorenzo and S. Barbarossa, “A bio-inspired swarming algorithm for decentralized access in cognitive radio,” *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6160–6174, 2011.
- [37] G. Mateos, J. A. Bazerque, and G. B. Giannakis, “Distributed sparse linear regression,” *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct 2010.
- [38] P. A. Forero, A. Cano, and G. B. Giannakis, “Consensus-based distributed support vector machines,” *J. Machine Learning Research*, vol. 11, no. May, pp. 1663–1707, 2010.
- [39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends. Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [40] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, “Novel type of phase transition in a system of self-driven particles,” *Physical Rev. Lett.*, vol. 75, no. 6, p. 1226, 1995.
- [41] V. Borkar and P. Varaiya, “Asymptotic agreement in distributed estimation,” *IEEE Trans. Autom. Control*, vol. 27, no. 3, pp. 650–655, 1982.
- [42] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [43] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., 1989.
- [44] L. Scardovi and R. Sepulchre, “Synchronization in networks of identical linear systems,” *Automatica*, vol. 45, no. 11, pp. 2557–2562, 2009.
- [45] Z. Li, Z. Duan, G. Chen, and L. Huang, “Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint,” *IEEE Trans. Circuits Syst. I*, vol. 57, no. 1, pp. 213–224, 2010.
- [46] U. Krause, “A discrete nonlinear and nonautonomous model of consensus formation,” *Commun. Difference Equations*, pp. 227–236, 2000.
- [47] N. Chopra and M. W. Spong, “Passivity-based control of multi-agent systems,” in *Advances in Robot Control*. Springer, 2006, pp. 107–134.
- [48] J. Zhou, J.-a. Lu, and J. Lu, “Adaptive synchronization of an uncertain complex dynamical network,” *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 652–656, 2006.
- [49] M. Arcak, “Passivity as a design tool for group coordination,” *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1380–1390, 2007.

- [50] G.-B. Stan and R. Sepulchre, “Analysis of interconnected oscillators by dissipativity theory,” *IEEE Trans. Autom. Control*, vol. 52, no. 2, pp. 256–270, 2007.
- [51] N. Chopra and M. W. Spong, “On exponential synchronization of Kuramoto oscillators,” *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 353–357, 2009.
- [52] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [53] A. Kashyap, T. Başar, and R. Srikant, “Quantized consensus,” *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.
- [54] S. Kar and J. M. Moura, “Distributed consensus algorithms in sensor networks: Quantized data and random link failures,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1383–1400, 2010.
- [55] G. Shi and K. H. Johansson, “Persistent graphs and consensus convergence,” in *IEEE 51st Conf. Decision and Control*, 2012, pp. 2046–2051.
- [56] J. Wolfowitz, “Products of indecomposable, aperiodic, stochastic matrices,” *Proc. American Mathematical Society*, vol. 14, no. 5, pp. 733–737, 1963.
- [57] J. M. Hendrickx and V. D. Blondel, “Convergence of linear and non-linear versions of Vicsek’s model,” in *Proc. 17th Int. Sympo. Mathematical Theory of Networks and Systems*, 2005, pp. 1229–1240.
- [58] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking,” in *44th IEEE Conf. Decision and Control/ 2005 European Control Conf.*, 2005, pp. 2996–3000.
- [59] B. Touri and A. Nedic, “Product of random stochastic matrices,” *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 437–448, Feb 2014.
- [60] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*. ACM, 2003, pp. 137–146.
- [61] S. Patterson and B. Bamieh, “Leader selection for optimal network coherence,” in *Proc. 49th IEEE Conf. Decision and Control*. IEEE, 2010, pp. 2692–2697.
- [62] M. Fardad, F. Lin, X. Zhang, and M. R. Jovanovic, “On new characterizations of social influence in social networks,” in *Proc. American Control Conf.*, 2013, pp. 4777–4782.
- [63] A. Gionis, E. Terzi, and P. Tsaparas, “Opinion maximization in social networks,” in *Proc. SIAM Int. Conf. Data Mining*. SIAM, 2013, pp. 387–395.

- [64] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, and A. Scaglione, “Binary opinion dynamics with stubborn agents,” *ACM Trans. Econ. Comp.*, vol. 1, no. 4, p. 19, 2013.
- [65] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, “Minimizing convergence error in multi-agent systems via leader selection: A supermodular optimization approach,” *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1480–1494, 2014.
- [66] L. Vassio, F. Fagnani, P. Frasca, and A. Ozdaglar, “Message passing optimization of harmonic influence centrality,” *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 109–120, 2014.
- [67] V. S. Borkar, A. Karnik, J. Nair, and S. Nalli, “Manufacturing consent,” *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 104–117, 2015.
- [68] G. Shi, K. C. Sou, H. Sandberg, and K. H. Johansson, “A graph-theoretic approach on optimizing informed-node selection in multi-agent tracking control,” *Physica D: Nonlinear Phenomena*, vol. 267, pp. 104–111, 2014.
- [69] K. Fitch and N. E. Leonard, “Information centrality and optimal leader selection in noisy networks,” in *IEEE 52nd Conf. Decision and Control*. IEEE, 2013, pp. 7510–7515.
- [70] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, “Minimizing convergence error in multi-agent systems via leader selection: A supermodular optimization approach,” *CoRR*, vol. abs/1306.4949, 2013.
- [71] S. Sundaram and C. N. Hadjicostis, “Finite-time distributed consensus in graphs with time-invariant topologies,” in *Proc. American Control Conf.*, 2007, pp. 711–716.
- [72] Y. Yuan, G.-B. Stan, L. Shi, M. Barahona, and J. Goncalves, “Decentralised minimum-time consensus,” *Automatica*, vol. 49, no. 5, pp. 1227–1235, 2013.
- [73] B. Ho and R. E. Kalman, “Effective construction of linear state-variable models from input/output functions,” *Automatisierungstechnik*, vol. 14, no. 1-12, pp. 545–548, 1966.
- [74] A. Tether, “Construction of minimal linear state-variable models from finite input-output data,” *IEEE Trans. Autom. Control*, vol. 15, no. 4, pp. 427–436, 1970.
- [75] L. Silverman, “Realization of linear dynamical systems,” *IEEE Trans. Autom. Control*, vol. 16, no. 6, pp. 554–567, 1971.
- [76] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, “Subgradient methods and consensus algorithms for solving convex optimization problems,” in *Proc. 47th IEEE Conf. Decision and Control*, 2008, pp. 4185–4190.

- [77] A. Jadbabaie, A. Ozdaglar, and M. Zargham, “A distributed Newton method for network optimization,” in *Proc. 48th IEEE Conf. Decision and Control / 28th Chinese Control Conf.* IEEE, 2009, pp. 2736–2741.
- [78] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [79] D. Jakovetic, J. Xavier, and J. M. Moura, “Cooperative convex optimization in networked systems: Augmented Lagrangian algorithms with directed gossip communication,” *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 3889–3902, 2011.
- [80] I. Lobel and A. Ozdaglar, “Distributed subgradient methods for convex optimization over random networks,” *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1291–1306, 2011.
- [81] J. C. Duchi, A. Agarwal, and M. J. Wainwright, “Dual averaging for distributed optimization: convergence analysis and network scaling,” *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, 2012.
- [82] K. I. Tsianos, S. Lawlor, and M. G. Rabbat, “Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning,” in *Proc. 50th Annu. Allerton Conf. Commun. Control Comp.* IEEE, 2012, pp. 1543–1550.
- [83] E. Wei and A. Ozdaglar, “On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers,” *arXiv preprint arXiv:1307.8254*, 2013.
- [84] B. Gharesifard and J. Cortés, “Distributed continuous-time convex optimization on weight-balanced digraphs,” *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 781–786, 2014.
- [85] A. Nedic and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, March 2015.
- [86] K. Yuan, Q. Ling, and W. Yin, “On the convergence of decentralized gradient descent,” *arXiv preprint arXiv:1310.7063*, 2013.
- [87] D. Jakovetic, J. Xavier, and J. M. Moura, “Fast distributed gradient methods,” *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, 2014.
- [88] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.
- [89] S. S. Ram, A. Nedic, and V. V. Veeravalli, “Incremental stochastic subgradient algorithms for convex optimization,” *SIAM J. Optim.*, vol. 20, no. 2, pp. 691–717, 2009.

- [90] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. Int. Conf. Information Processing in Sensor Networks*, 2005, pp. 63–70.
- [91] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, “Newton-Raphson consensus for distributed convex optimization,” in *Proc. 50th IEEE Conf. Decision / 2011 Control and European Control Conf.*, 2011, pp. 5917–5922.
- [92] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, “Consensus in ad hoc wsns with noisy linkspart i: Distributed estimation of deterministic signals,” *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, 2008.
- [93] K. Tsianos and M. Rabbat, “Distributed dual averaging for convex optimization under communication delays,” in *Proc. American Control Conf.*, Jun. 2012, pp. 1067–1072.
- [94] P. Lin, W. Ren, and Y. Song, “Distributed multi-agent optimization subject to nonidentical constraints and communication delays,” *Automatica*, vol. 65, pp. 120–131, 2016.
- [95] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
- [96] V. S. Mai and E. H. Abed, “Opinion dynamics with persistent leaders,” in *IEEE 53rd Conf. Decision and Control*, 2014, pp. 2907–2913.
- [97] H. Minc, *Nonnegative Matrices*. John Wiley and Sons, New York, 1988.
- [98] J. Lorenz, “A stabilization theorem for dynamics of continuous opinions,” *Physica A*, vol. 355, no. 1, pp. 217–223, 2005.
- [99] D. J. Hartfiel, *Nonhomogeneous Matrix Products*. World Scientific, 2002.
- [100] A. Olshevsky and J. N. Tsitsiklis, “Convergence speed in distributed consensus and averaging,” *SIAM Rev.*, vol. 53, no. 4, pp. 747–772, 2011.
- [101] W. Wang and J.-J. Slotine, “A theoretical study of different leader roles in networks,” *IEEE Trans. Autom. Control*, vol. 51, no. 7, pp. 1156–1161, 2006.
- [102] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, “Controllability of multi-agent systems from a graph-theoretic perspective,” *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 162–186, 2009.
- [103] S. Joshi and S. Boyd, “Sensor selection via convex optimization,” *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, 2009.
- [104] F. Lin, M. Fardad, and M. R. Jovanović, “Algorithms for leader selection in large dynamical networks: Noise-corrupted leaders,” in *Proc. 50th IEEE Conf. Decision Control, European Control Conf.*, 2011, pp. 2932–2937.

- [105] J. Long, S. O. Memik, and M. Grayson, "Optimization of an on-chip active cooling system based on thin-film thermoelectric coolers," in *Proc. Conf. Design, Automation and Test in Europe*, 2010, pp. 117–122.
- [106] V. S. Borkar, J. Nair, and N. Sanketh, "Manufacturing consent," in *48th Allerton Conf. Commun. Control Comp.*, 2010, pp. 1550–1555.
- [107] V. S. Mai and E. H. Abed, "Dynamic consensus measure and optimal selection of direct followers in multiagent networks," in *Proc. American Control Conf. IEEE*, 2016, pp. 2880–2885.
- [108] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions –I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [109] M. Conforti and G. Cornuéjols, "Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem," *Discrete Applied Mathematics*, vol. 7, no. 3, pp. 251–274, 1984.
- [110] G. Ranjan and Z.-L. Zhang, "Geometry of complex networks and topological centrality," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 17, pp. 3833–3845, 2013.
- [111] K. Stephenson and M. Zelen, "Rethinking centrality: Methods and examples," *Social Networks*, vol. 11, no. 1, pp. 1–37, 1989.
- [112] I. Poulakakis, L. Scardovi, and N. E. Leonard, "Node classification in networks of stochastic evidence accumulators," *arXiv preprint arXiv:1210.4235*, 2012.
- [113] M. Brand, "A random walks perspective on maximizing satisfaction and profit," in *SDM*. SIAM, 2005, pp. 12–19.
- [114] S. J. Wright, *Primal-dual interior-point methods*. Siam, 1997.
- [115] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [116] F. A. Potra and S. J. Wright, "Interior-point methods," *J. Computational and Applied Mathematics*, vol. 124, no. 1–2, pp. 281–302, 2000.
- [117] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Springer Science & Business Media, 2004, vol. 87.
- [118] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [119] J. Long, D. Li, S. O. Memik, and S. Ulgen, "Theory and analysis for optimization of on-chip thermoelectric cooling systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1628–1632, 2013.

- [120] A. Ganesan, S. R. Ross, and B. R. Barmish, “An extreme point result for convexity, concavity and monotonicity of parameterized linear equation solutions,” *Linear Algebra Appl.*, vol. 390, pp. 61–73, 2004.
- [121] D. M. Topkis, “Minimizing a submodular function on a lattice,” *Operations Research*, vol. 26, no. 2, pp. 305–321, 1978.
- [122] P. Milgrom and J. Roberts, “Rationalizability, learning, and equilibrium in games with strategic complementarities,” *Econometrica*, vol. 58, no. 6, pp. 1255–1277, 1990.
- [123] J. Leskovec, D. Huttenlocher, and J. Kleinberg, “Predicting positive and negative links in online social networks,” in *Proc. 19th Int. Conf. World Wide Web*. ACM, 2010, pp. 641–650.
- [124] J. Currie and D. I. Wilson, “OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User,” in *Found. Comp.-Aided Process Operations*, N. Sahinidis and J. Pinto, Eds., Savannah, Georgia, USA, 8–11 January 2012.
- [125] P. Colaneri, R. H. Middleton, Z. Chen, D. Caporale, and F. Blanchini, “Convexity of the cost functional in an optimal control problem for a class of positive switched systems,” *Automatica*, vol. 50, no. 4, pp. 1227–1234, 2014.
- [126] B. P. Lathi, *Linear Systems and Signals*. Oxford University Press, 2009.
- [127] D. G. Luenberger, *Optimization by Vector Space Methods*. John Wiley & Sons, 1997.
- [128] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, “Distributed finite-time average consensus in digraphs in the presence of time delays,” *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 4, pp. 370–381, 2015.
- [129] C. D. Godsil, G. Royle, and C. Godsil, *Algebraic Graph Theory*. Springer New York, 2001, vol. 8.
- [130] S. Martini, M. Egerstedt, and A. Bicchi, “Controllability analysis of multi-agent systems using relaxed equitable partitions,” *Int. J. Syst. Control Commun.*, vol. 2, no. 1-3, pp. 100–121, 2010.
- [131] S. Zhang, M. Cao, and M. K. Camlibel, “Upper and lower bounds for controllable subspaces of networks of diffusively coupled agents,” *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 745–750, 2014.
- [132] S. Zhang, M. K. Camlibel, and M. Cao, “Controllability of diffusively-coupled multi-agent systems with general and distance regular coupling topologies,” in *Proc. 50th IEEE Conf. Decision / 2011 Control and European Control Conf.*, 2011, pp. 759–764.

- [133] J. Wang and N. Elia, “A control perspective for centralized and distributed convex optimization,” in *Proc. 50th IEEE Conf. Decision and Control / 2011 European Control Conf.*, 2011, pp. 3800–3805.
- [134] A. Olshevsky, “Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control,” *arXiv preprint arXiv:1411.4186v6*, 2016.
- [135] I.-A. Chen, “Fast distributed first-order methods,” Master’s thesis, MIT, 2012.
- [136] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: An exact first-order algorithm for decentralized consensus optimization,” *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.
- [137] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [138] S. Boyd, L. Xiao, and A. Mutapcic, “Subgradient methods,” *Lecture notes of EE392o, Stanford University, Autumn Quarter*, 2003. [Online]. Available: http://web.mit.edu/6.976/www/notes/subgrad_method.pdf
- [139] B. Nejad, S. Attia, and J. Raisch, “Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies,” in *Int. Sympo. Info. Commun. Automation Tech.*, 2009, pp. 1–7.
- [140] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation and consensus using linear iterative strategies,” *IEEE J. Select. Areas Commun.*, vol. 26, no. 4, pp. 650–660, 2008.
- [141] T. Charalambous, M. G. Rabbat, M. Johansson, and C. N. Hadjicostis, “Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum,” *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 2, pp. 137–148, 2016.
- [142] A. E. Brouwer, A. M. Cohen, , and A. Neumaier, *Distance-Regular Graphs*. New York: Springer-Verlag, 1989, vol. 18.
- [143] E. R. van Dam, J. H. Koolen, and H. Tanaka, “Distance-regular graphs,” *Electronic J. Combinatorics*, vol. DS22, 2014.
- [144] J. Wang and N. Elia, “Control approach to distributed optimization,” in *Proc. 48th Annu. Allerton Conf. Commun. Control Comp.*, 2010, pp. 557–561.
- [145] A. Nedic and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, March 2015.

- [146] D. Kempe, A. Dobra, and J. Gehrke, “Gossip-based computation of aggregate information,” in *Proc. 44th Annu. IEEE Symp. Found. Comp. Sci.*, 2003, pp. 482–491.
- [147] A. Makhdoumi and A. Ozdaglar, “Graph balancing for distributed subgradient methods over directed graphs,” in *Proc. 54th IEEE Conf. Decision and Control*, 2015, pp. 1364–1371.
- [148] S. S. Ram, A. Nedić, and V. V. Veeravalli, “Distributed stochastic subgradient projection algorithms for convex optimization,” *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.
- [149] S. Lee and A. Nedic, “Distributed random projection algorithm for convex optimization,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 221–229, 2013.
- [150] V. S. Mai and E. H. Abed, “Distributed optimization over weighted directed graphs using row stochastic matrix,” in *Proc. American Control Conf.*, 2016, pp. 7165–7170.
- [151] I. Lobel, A. Ozdaglar, and D. Feijer, “Distributed multi-agent optimization with state-dependent communication,” *Mathematical Programming*, vol. 129, no. 2, pp. 255–284, 2011.
- [152] A. Hoffmann, “The distance to the intersection of two convex sets expressed by the distances to each of them,” *Mathematische Nachrichten*, vol. 157, no. 1, pp. 81–98, 1992.
- [153] H. H. Bauschke and J. M. Borwein, “On projection algorithms for solving convex feasibility problems,” *SIAM Rev.*, vol. 38, no. 3, pp. 367–426, 1996.
- [154] Z. Qu, C. Li, and F. Lewis, “Cooperative control based on distributed estimation of network connectivity,” in *Proc. American Control Conf.*, 2011, pp. 3441–3446.
- [155] A. Priolo, A. Gasparri, E. Montijano, and C. Sagues, “A decentralized algorithm for balancing a strongly connected weighted digraph,” in *Proc. American Control Conf.*, Jun. 2013, pp. 6547–6552.
- [156] I. Matei and J. S. Baras, “A comparison between upper bounds on performance of two consensus-based distributed optimization algorithms,” in *Estimation and Control of Networked Systems*, vol. 3, no. 1, 2012, pp. 168–173.
- [157] H. Robbins and D. Siegmund, “A convergence theorem for nonnegative almost supermartingales and some applications,” *Methods in Statistics*, pp. 233–257, 1971.

- [158] C. Xi, V. S. Mai, E. H. Abed, and U. A. Khan, “Linear convergence in directed optimization with row-stochastic matrices,” *arXiv preprint arXiv:1611.06160*, 2016.
- [159] W. B. Gragg and A. Lindquist”, “On the partial realization problem,” *Linear Algebra Appl.*, vol. 50, pp. 277 – 319, 1983.
- [160] L. Ljung, *System Identification*. Wiley Online Library, 1999.
- [161] S. Beheshti and M. A. Dahleh, “Noisy data and impulse response estimation,” *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 510–521, 2010.
- [162] Y. Yuan, G.-B. Stan, L. Shi, M. Barahona, and J. Gonçalves, “Minimal-time uncertain output final value of unknown DT-LTI systems with application to the decentralised network consensus problem,” in *Proc. Int. Sympo. Mathematical Theory of Netw. Syst.*, 2010.
- [163] F. Pasqualetti, A. Bicchi, and F. Bullo, “Distributed intrusion detection for secure consensus computations,” in *Proc. 46th IEEE Conf. Decision and Control*. IEEE, 2007, pp. 5594–5599.
- [164] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, “Distributed fault detection for interconnected second-order systems,” *Automatica*, vol. 47, no. 12, pp. 2757–2764, 2011.
- [165] F. Pasqualetti, A. Bicchi, and F. Bullo, “Consensus computation in unreliable networks: A system theoretic approach,” *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, 2012.
- [166] F. Pasqualetti, F. Dörfler, and F. Bullo, “Attack detection and identification in cyber-physical systems,” *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, 2013.
- [167] J. McDonald, M. Neumann, H. Schneider, and M. Tsatsomeros, “Inverse M -matrix inequalities and generalized ultrametric matrices,” *Linear Algebra Appl.*, vol. 220, pp. 321–341, 1995.
- [168] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*. Siam, 2002.
- [169] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*. Springer, 2003, vol. 13.
- [170] C. Meyer, Jr., “Generalized inversion of modified matrices,” *SIAM J. Applied Math.*, vol. 24, no. 3, pp. 315–323, 1973.
- [171] R. Plemmons, “ M -matrix characterizations. I–Nonsingular M -matrices,” *Linear Algebra Appl.*, vol. 18, no. 2, pp. 175 – 188, 1977.