Multi-Agent Consensus and Averaging on General Network Topology

by

Kai Cai

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology

Abstract

Multi-Agent Consensus and Averaging on General Network Topology

Kai Cai

Doctor of Philosophy

Department of Computational Intelligence and Systems Science

Tokyo Institute of Technology

2011

We study two fundamental problems, consensus and averaging, in multi-agent systems where component agents are interconnected through a network. Such problems relate intimately to many phenomena and applications (e.g., bird flocking, oscillator synchronization, and local balancing), in which all agents need to reach an agreement on some state of their common interest. Central to solving these problems is the topology of the interconnection network among agents, which determines who is coupled to whom. Our principal objective is to find the most general topological conditions, under which we design distributed algorithms to solve both problems.

First, we deal with the averaging problem in the setup where agents' states are real-valued, and networks static or dynamic. Novel algorithms are proposed where additional variables, called surplus, are associated to individual agents to keep track of their state updates. Under these algorithms, it is justified that state averaging is ensured on general network topology. Second, we investigate both consensus and averaging problems when agents' states are quantized, and networks gossip-type randomized. For consensus, an algorithm is designed and a necessary and sufficient topological condition derived to guarantee convergence. For averaging, a counterpart surplus-based algorithm is developed, which is proved again to converge to the average on general topology. Furthermore, we analyze the convergence time of these quantized gossip algorithms, and obtain that the upper bounds are of polynomial orders.

Acknowledgements

During the past three years, it has been my great pleasure to study under the supervision of Professor Hideaki Ishii. I wish to express my sincere gratitude for his patience and guidance throughout the course of study, which has led me to the accomplishment of this thesis.

I would like to thank the students and secretary in the group for their friendship and administrative help. Special appreciation dues to Kogure-kun and Nakamura-san.

I am also grateful to the thesis committee members – Professors Kaoru Hirota, Yoshiyuki Kabashima, Masayuki Fujita, Yoshihiro Miyake, and Misako Takayasu – for their helpful discussions and comments.

Financial support from the Ministry of Education, Culture, Sports, Science and Technology of Japan is gratefully acknowledged.

Finally, I am forever indebted to my wife Akiko, for her constant love and inspiration for life, as well as our parents, for their understanding and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

From the systems theoretic viewpoint, a *multi-agent system* is an arbitrary collection of more-or-less autonomous agents, usually interconnected through a network. Such systems are pervasive, across from natural phenomena (e.g., bird flocking, fish schooling, and biochemical reactions), to scientific disciplines (e.g., oscillator synchronization, distributed artificial intelligence, and game theory), as well as engineering practice (e.g., robot teams, sensor networks, and multi-core processors). In the engineering applications, particular attention is paid to the *design* of individual strategies for component agents, so that they cooperatively interact with neighboring peers in pursuit of a common collective goal (mimicking the prototypical insect colony).

In this thesis we deal with multi-agent systems potential for engineering applications. Among various issues arising in such systems, our work is devoted to two fundamental problems, *consensus and averaging*, which can be described as follows. Consider that every agent is associated with a numerical value, often referred to as state. The value may represent any physical quantity of concern depending on context, for instance speed, frequency, and temperature measurement. Through sensing/communication, agents exchange state information with their neighbors, and based on the obtained neighbors' states they update their own values according to a prescribed algorithm. The objective of consensus is to design appropriate algorithms by which agents may update their values in an iterative fashion so that eventually

they reach an agreement on some common value. On the other hand, the averaging problem is of a special form of consensus: It requires the agreed common value to be the average of the values initially associated with agents. It can be seen from the above description that the consensus and averaging problems are indeed *abstractions* which potentially subsume, or at least closely relate to, a variety of concrete issues in multi-agent systems (see Literature Survey below). Thus by studying these abstractions, our aim is to explore underlying mechanisms common to a class of issues, meanwhile without being bogged down with issue-specific details.

The algorithms used by agents to update their values are called consensus and averaging algorithms, respectively. We restrict our attention to such algorithms of *distributed* type: There is no agent taking the role of a leader, and everyone executes the same algorithm. Central to the design of distributed consensus and averaging algorithms is the *topology* of the interconnection network among agents, which determines for individual agents who are their neighbors. The principal objective of this thesis is to derive the most general conditions on network topology that permit the existence of distributed algorithms solving the consensus and averaging problems, respectively; and under those general conditions, construct provably correct solution algorithms.

1.1 Literature Survey

We review the key references relevant to the main topic of this thesis. We start by illustrating a set of subjects taken from diverse fields that may be considered from the perspective of consensus and averaging.

Animal group behavior is the phenomenon that very simple individual actions expands into highly sophisticated collective behavior, like bird flocking, fish schooling, and fly swarming. Typically individual animals are not aware of the global picture, and evidently there does not exist any sort of intentional, centralized supervision. Emergent behavior may be viewed as a form of multi-agent consensus, because each

animal must collaborate with its neighbors so as to achieve agreement on the position, orientation, and speed of their motion. In 1987, Reynolds [68] wrote the celebrated program called boids, which vividly simulates the emergent behavior in a flock of birds. This is achieved by programming each virtual bird with merely three simple rules: separation, alignment, and cohesion with respect to local flocking mates. More recently, in [77] Vicsek et al. proposed a compelling model for a collection of self-driven particles all moving in the plane with the same speed but with different headings. Each particle's heading is updated towards the average of its own heading plus those of its neighbors. This local updating rule gives rise to an emergent behavior – all particles eventually move in the same direction – which is supported by a series of intriguing simulations.

Synchronization is a phenomenon ubiquitous in biology and physics, in which a system of coupled oscillators spontaneously locks to a common frequency, despite the differences in the natural frequencies of individual oscillators. Examples include networks of pacemaker cells in the heart, groups of synchronously flashing fireflies, and arrays of lasers. As described, synchronization may also be considered from the consensus viewpoint, the state of interest being frequency. In the seminal work [47], the Kuramoto model was developed to study collective synchronization of coupled oscillators; it was found that if the coupling strength exceeds a certain threshold, there occurs a phase transition: some oscillators synchronize while others remain incoherent. A summary of this work is provided in [72]. It is also worth noting that synchronization has been addressed from the perspective of stability of dynamical systems [55, 62].

Load balancing is a fundamental task in a multi-core processor or a computer cluster, where individuals cooperatively equalize the distribution of workloads to achieve optimal resource utilization. Each core or computer may initially be loaded with different amounts of tasks queued for processing, and each balances the uneven distribution by means of transferring tasks to its neighbors through available channels. As such, load balancing corresponds to the multi-agent averaging problem. There is

a large volume of literature on load balancing; notably in [1, 36] balancing algorithms that rely only on local information were proposed and analyzed.

Information fusion is an essential task in sensor networks. Suppose that each sensor takes a measurement of some environment parameter (e.g., temperature, humidity). The average of these measurements provides a sufficient statistic for many problems of interest in sensor networks, including minimum variance estimation and optimal detection of the parameter [63]. To compute the average measurement in a distributed way, individual sensors must exchange information among local peers and update their measurements based on those received from neighbors – this corresponds again to the multi-agent averaging problem. Important issues of information fusion in sensor networks include signal quantization owing to typically wireless digital networks, limited capacity in storage memories and communication channels, and robust strategies against link failure, node malfunction, and asynchronous local clocks; see [34, 80] and the references therein.

Distributed mobile robotics is an engineering filed where many basic problems may be regarded as consensus or averaging of certain sorts. In [2], local strategies were proposed to gather together a group of mobile robots at a common location, or rendezvous; this is consensus on the robots' positions. In [73], protocols were studied to get mobile robots to form circular and polygon formations; in this case, robots agree on their relative distances and orientations. In [25], distributed algorithms were designed for a team of vehicles to uniformly cover an area; thus in effect, the target area is partitioned in the average sense. A comprehensive list of interesting problems in distributed mobile robotics is documented in [19].

So far we have seen that the concepts of consensus and averaging appear, with one form or another, in diverse areas of study. Rigorous theoretic treatments of these two problems are being undertaken in the systems control community, which have recently become a very active research topic. We turn next to introducing the extensive developments witnessed in this community.

In systems control, research on the multi-agent consensus and averaging prob-

lems dates back to the pioneering work [9, 76]. This work is done in the context of parallel and distributed computing, where multiple processors iteratively perform computations and exchange messages so that they all compute a desired value. Then in the past decade, stimulated by new applications especially like sensor networks and mobile robots, substantial work addressing both problems has been carried out; refer to the survey papers [59, 67] and the monographs [13, 50, 66].

Early efforts [11, 42, 51, 56, 61, 65, 79] primarily adopt the following setup. Every agent is assumed to possess a real-valued state. Agents can communicate with and/or sense the precise states of their neighbors, and based on that information they update their own states according to a prescribed algorithm. The communication or sensing networks connecting agents may be either static or deterministic time-varying. For this fundamental setting, basic solution algorithms are proposed, instrumental analysis tools introduced, and essential conditions on network topology derived to ensure consensus and averaging.

In order to further tackle the problems in more realistic scenarios, the work that follows frequently takes into account a variety of constraints in the communication networks among agents. A notable constraint is state quantization (as contrasted with real valued), which is common in digital communication channels often of limited data rate. Quantization may be due also to that agents' physical storage memories are sometimes of finite capacity. Many quantization strategies are studied (e.g., uniform, logarithmic, and dynamic quantization), and particular attention is given to the tradeoffs between the coarseness of quantization and the precision of consensus and averaging [3, 20, 21, 23, 28, 33, 49, 57].

Another constraint often considered is random networks (as contrasted with deterministic), which potentially models many stochastic phenomena in practical networks, including noise, packet loss, link failure, and node malfunction. Different random models are proposed and analyzed (e.g., Erdős Rényi model, Markovian model, and gossiping), and the design of randomized algorithms is also a popular approach [12, 29, 39, 54, 64, 74, 78].

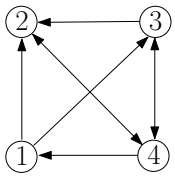
More recently, the setup that deals simultaneously with both state quantization and random networks has appeared and attracted much attention [16, 18, 22, 32, 45, 48, 81]. These efforts have extended the early results on consensus and averaging to fairly practical situations. We shall discuss in more detail the above cited references in the corresponding chapters.

1.2 Overview of the Thesis and Contributions

In this thesis, we explore the most general conditions on network topology for the multi-agent consensus and averaging problems, and develop corresponding distributed algorithms. For our setup, we consider both real and quantized states, and both deterministic and random networks. The thesis is outlined as follows.

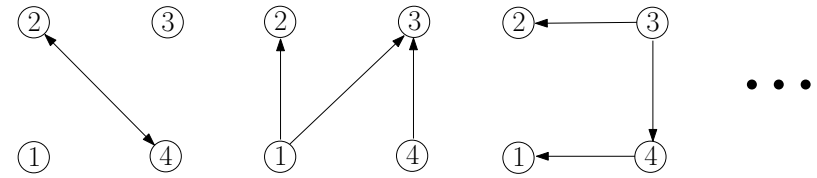
In Chapter 2, we collect basic notions in graph theory and important results in nonnegative matrix theory which are to be referred to in the later chapters.

We begin our investigation from Chapter 3, by studying the averaging problem in the basic setting where states are real valued, networks deterministic, and also agents' clocks synchronized. In particular, we consider that the interconnection topology among agents is fixed for all time. In Fig. 1.1(a) we use a *graph* to model the interconnection topology: Each *node* stands for an agent, and each *edge* stands for an existing connection between two agents; here the graph is invariant at all time. For this basic setup, we propose a distributed algorithm which is proved to guarantee state averaging on general networks (Theorem 3.1). The essence of the algorithm is to keep local records of individual state updates, thereby ensuring consensus on the average despite that the state sum of agents is not preserved. This is achieved by augmenting a new variable for each agent, which we call "surplus". Moreover, we derive bounds, in both general and special topologies, for a parameter of the algorithm which is essential to guarantee convergence. For analysis tools, we rely on graph theory and nonnegative matrix theory (summarized in Chapter 2), with the eigenvalue perturbation theory playing a crucial role.



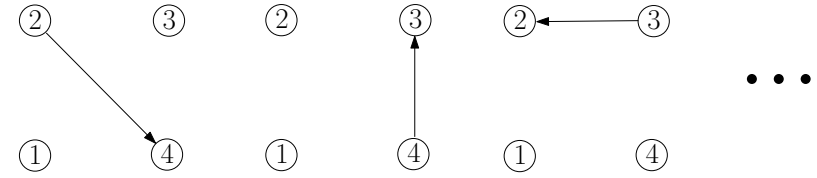
time $k = 0, 1, 2, \cdots$

(a) Deterministic static network, Chapter 3



time $k = 0$ $k = 1$ $k = 2$ $\cdots$

(b) Deterministic time-varying network, Chapter 4



time $k = 0$ $k = 1$ $k = 2$ $\cdots$

(c) Random gossip network, Chapters 4, 5, and 6

Figure 1.1: Network models in the thesis.

Chapter 4 progresses to target dynamic networks, where the interconnection topology among agents is time-varying. The time-varying mechanism can be either deterministic (see Fig. 1.1(b)) or random. In the random case, we consider that agents asynchronously "gossip" with one another in the sense that only a single interaction randomly occurs at a time (see Fig. 1.1(c)). We propose distributed algorithms in both deterministic and random scenarios. These algorithms are based again on surplus variables, and are justified to ensure state averaging on general networks (Theorems 4.1 and 4.2). The analysis tools for the random case are the same as those in Chapter 3, while for the deterministic case Lyapunov-type arguments are used.

In Chapters 5 and 6, we study both consensus and averaging problems in the new setting where states are quantized, and networks randomized in the gossip sense (Fig. 1.1(c)). To model quantization effects, each agent's state is abstracted to be

an integer. In Chapter 5 we propose a class of consensus and averaging algorithms, respectively; for each, we derive a necessary and sufficient condition on network topology to guarantee the corresponding convergence (Theorems 5.1 and 5.2). In particular, the averaging algorithm is again surplus-based, and the characterizing condition on network topology is general. Further, we discuss two parameters of the averaging algorithm, and remark on their respective relations to convergence as well as performance. For analysis, we employ tools from finite Markov chain theory.

Chapter 6 moves on to study the convergence time of the consensus and averaging algorithms studied in Chapter 5. To this end, we investigate the shrinking time of the smallest interval that contains all states for the consensus algorithm, and the decay time of a suitable Lyapunov function for the averaging algorithm. The investigation leads us to characterizing the convergence time by the hitting time in certain special Markov chains. We simplify the structures of state transition by considering the special case of complete networks, and derive polynomial upper bounds with respect to the number of agents on convergence time (Theorems 6.1 and 6.2).

Finally, we conclude in Chapter 7 by summarizing the thesis, and propose potential future research topics.

The contributions of the thesis are summarized as follows, and the relations to other works in the literature shown in Table 1.1. First, in the static network model, we propose a novel distributed algorithm to solve the multi-agent averaging problem. The novelty lies in the augmentation of additional surplus variables which collectively keep track of state updates. Under this algorithm, a necessary and sufficient condition on network topology is derived to guarantee state averaging; the condition turns out to be more general than those previously reported in the literature, in the sense that it does not require symmetric or balanced topological structures. In addition, certain useful tools from matrix perturbation theory are introduced to analyze and establish convergence.

Second, in the setup of dynamic networks, we again design original distributed algorithms based on surplus variables to solve the averaging problem. In particular,

we consider only unidirectional (as contrasted with bidirectional) information flow for averaging on randomized gossip-type networks; this setting seems to be new. The topological condition we derived to ensure averaging is general: While in the literature it is required that the network be symmetric or balanced at *every* moment, our condition does not require so at *any* moment. This feature would potentially simplify the implementation of our designed algorithms, and also in this sense, the algorithms may be robust with respect to possible perturbations on network structures (e.g., random packet loss and node failure)

Third, we pose the problems of solving both consensus and averaging in the setup where networks are randomized (gossip type with unidirectional information flow) and states quantized (all integers). For the consensus problem, we propose a class of algorithms, under which we derive a necessary and sufficient condition on network topology that guarantees convergence to some common value. For the averaging problem, we develop again a novel surplus-based algorithm which provably ensures state averaging on general networks. Owing to the integer constraint, the set of all states turns out finite, and therefore the use of certain tools from finite Markov chain theory is essential. Moreover, to demonstrate the efficiency of both consensus and averaging algorithms, we find polynomial upper bounds for their convergence time on complete networks. For the averaging algorithm, in particular, by studying the corresponding transition structure we clarify the relation between the behavior of surplus and the speed of convergence.

Table 1.1: Relations to other works in the literature.

	Balanced & strongly connected network topology	General strongly connected network topology
Real-valued states	[12], [39], [42], [54], [57], [61], [65], [74], [79]	Our work in Chapters 3 and 4
Quantized states	[21], [22], [23], [33], [45], [48], [49], [57], [81]	Our work in Chapters 5 and 6

Chapter 2

Mathematical Preliminaries

For convenience of reference, this chapter collects basic notions in graph theory and important results in nonnegative matrix theory which will be used throughout the rest of the thesis. As the material is standard, we shall omit proofs and refer to textbooks for detailed developments.

2.1 Graph Theory [4, 50]

Directed and undirected graphs. A *directed graph* (or simply *digraph*) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a *node set* $\mathcal{V}$ and an *edge set* $\mathcal{E}$ of ordered pairs of nodes. By convention, the direction of an edge (j, i) is from node j to i; and selfloop edges are excluded, i.e., $(i, i) \notin \mathcal{E}$. A *path* in a digraph is a finite sequence of edges: $(i_1, i_2)(i_2, i_3) \cdots (i_{k-1}, i_k)$. The *local* structure of a digraph $\mathcal{G}$ is described by *neighbor sets*: For each node $i \in \mathcal{V}$, let $\mathcal{N}_i^+ := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ denote the set of its *in-neighbors*, and $\mathcal{N}_i^- := \{h \in \mathcal{V} : (i, h) \in \mathcal{E}\}$ the set of its *out-neighbors*. An *undirected* (or symmetric) graph $\mathcal{G}$ is such that $(j, i) \in \mathcal{E}$ implies $(i, j) \in \mathcal{E}$. Clearly undirected graphs are special cases of digraphs; and unless otherwise stated, we shall deal exclusively with digraphs.

As an illustration, in Fig. 2.1 (a) is a digraph, where for node 1, $\mathcal{N}_1^+ = \{2\}$ and $\mathcal{N}_1^- = \{2, 3\}$. Here (b) is an undirected graph.

Connectivity in digraphs. A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *complete* if $(j, i) \in \mathcal{E}$ exists between every pair of distinct nodes $j, i \in \mathcal{V}$. A node i is *reachable* from another node

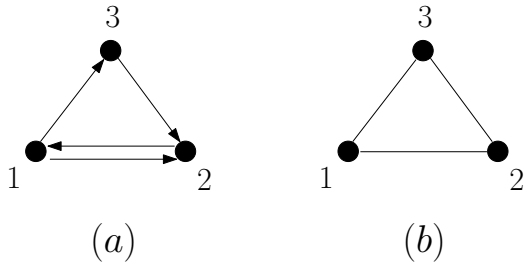


Figure 2.1: Directed and undirected graphs.

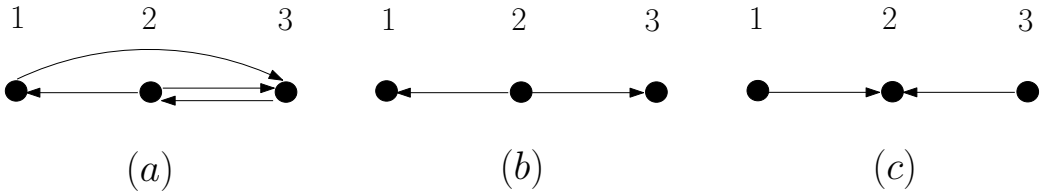


Figure 2.2: Digraphs connectivity.

j if there exists a path from j to i. A digraph is *strongly connected* if every node is reachable from every other node. Now let $\mathcal{U}$ be a nonempty subset of $\mathcal{V}$. The subset $\mathcal{U}$ is said to be *closed* if every node u in $\mathcal{U}$ is not reachable from any node v in $\mathcal{V}-\mathcal{U}$. Intuitively, there is no edge pointing in the subset $\mathcal{U}$ from outside. Also, the digraph $\mathcal{G}_{\mathcal{U}} = (\mathcal{U}, \mathcal{E} \cap (\mathcal{U} \times \mathcal{U}))$ is called the *induced subdigraph* by $\mathcal{U}$. A *strong component* of $\mathcal{G}$ is a maximal induced subdigraph of $\mathcal{G}$ which is strongly connected. Note that a maximal induced subdigraph need not be unique in general. Lastly, a node $i \in \mathcal{V}$ is said a *globally reachable node* if every other node is reachable from i. Evidently a digraph is strongly connected if and only if every node is globally reachable.

In Fig. 2.2, digraph (a) is strongly connected, while (b) is not, for only node 2 is globally reachable. Digraph (c) does not have a globally reachable node.

We shall need the following result from [50, Theorem 2.1], which reveals an important relation between digraph connectivity and its structure.

Lemma 2.1. A digraph $\mathcal{G}$ has a globally reachable node if and only if it has a unique closed strong component. Furthermore, this unique closed strong component is the induced subdigraph by the set of all globally reachable nodes.

Dynamic digraphs. A *dynamic digraph* $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$, $k \in \mathbb{Z}_+$, is one

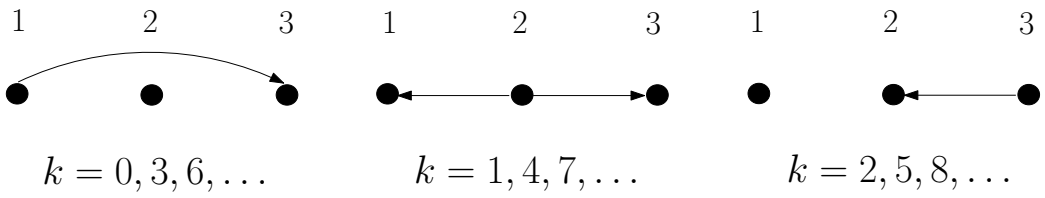


Figure 2.3: Dynamic digraph.

whose node set is fixed while edge set time varying. The time-varying mechanism can be deterministic or random; in either case, the connectivity of $\mathcal{G}(k)$ changes over time. We introduce the notion of joint connectivity over some time interval $[k_1, k_2]$. Define the *union digraph* $\mathcal{G}([k_1, k_2]) := \left(\mathcal{V}, \bigcup_{k\in[k_1,k_2]} \mathcal{E}(k)\right)$; namely, the edge set of $\mathcal{G}([k_1, k_2])$ is the union of those over the interval $[k_1, k_2]$. A dynamic digraph $\mathcal{G}(k)$ is *jointly strongly connected* if there exists a finite $k_1 \in \mathbb{Z}_+$ such that for every $k_0 \in \mathbb{Z}_+$, the union digraph $\mathcal{G}([k_0, k_0 + k_1])$ is strongly connected.

Consider the dynamic digraph $\mathcal{G}(k)$ displayed in Fig. 2.3 with three topologies switching periodically. For example, the union digraph over the time interval $[1, 2]$ is $\mathcal{G}([1, 2]) = \{\{1, 2, 3\}, \{(2, 1), (2, 3), (3, 2)\}\}$. This dynamic digraph $\mathcal{G}(k)$ is jointly strongly connected, because fixing an arbitrary finite $k_1 \geq 2$, the union digraph $\mathcal{G}([k_0, k_0 + k_1])$ is strongly connected for every $k_0 \in \mathbb{Z}_+$.

2.2 Nonnegative Matrix Theory

Important matrices (e.g., adjacency matrix, Laplacian matrix, to be introduced in Chapter 3) associated to digraphs are nonnegative. In this section, all matrices are real and square. Main references are [8, 40].

A matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is *nonnegative* if all $a_{ij} \geq 0$, and is *positive* if all $a_{ij} > 0$. These are similarly defined for vectors. The *spectrum* of A, denoted by $\sigma(A)$, is the set of all eigenvalues; the *spectral radius*, $\rho(A)$, is the maximum modulus of all eigenvalues. Let A be nonnegative, and have constant row (resp. column) sums, then $\rho(A)$ is equal to the row (resp. column) sum. The well-known result below provides an estimation for $\sigma(A)$ [40, Theorem 6.1.1].

Lemma 2.2. (Geršgorin) All the eigenvalues of a nonnegative matrix A are located in the union of n discs:

$$\bigcup_{i=1}^{n}\{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j=1, j\neq i}^{n} a_{ij}\}.$$

A matrix A is said *reducible* if either $n = 1$, or there is a permutation matrix P (resulting from permuting the rows of the identity matrix) such that

$$PAP^T = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix},$$

where B, D are nonempty square matrices. Otherwise A is *irreducible*. A positive matrix is always irreducible; the following fact [8, Theorem 2.7 of Chapter 2] characterizes when a nonnegative matrix is so.

Lemma 2.3. A nonnegative matrix $A = (a_{ij})$ is irreducible if and only if the digraph $\mathcal{G}(A)$ is strongly connected. Here $\mathcal{G}(A)$ is constructed from A with n nodes and with an edge (i, j) if and only if $a_{ij} > 0$.

For nonnegative and irreducible matrices, the celebrated theorem [40, Theorem 8.4.4] below states crucial properties of spectral radius.

Lemma 2.4. (Perron-Frobenius) If A is nonnegative and irreducible, then

(i) $\rho(A)$ is a simple eigenvalue;

(ii) $\rho(A) > 0$, and has a positive eigenvector.

It is convenient to state here an easy corollary of the Perron-Frobenius Theorem.

Lemma 2.5. (cf. [35, Chapter XIII]) Let A be nonnegative and irreducible, and λ be an eigenvalue of A. If there is a positive vector v such that $Av = \lambda v$, then $\lambda = \rho(A)$.

Finally, we turn to a special class of nonnegative matrices which we shall frequently encounter. A nonnegative matrix A is said *row stochastic* if every row sums up to one, *column stochastic* if every column sums up to one, and further *doubly stochastic* if it is both row and column stochastic. The spectral radius of a row or column stochastic matrix is 1; and the products of row (resp. column) stochastic matrices are again

row (resp. column) stochastic matrices. By Lemma 2.4, if A is row (resp. column) stochastic and irreducible, then the spectral radius 1 is a simple eigenvalue and a corresponding right (resp. left) eigenvector is $\mathbf{1} := [1 \cdots 1]^T$.

Chapter 3

Averaging on Static Digraphs

3.1 Introduction

In this chapter, we study multi-agent average consensus in its most fundamental setting: Agents's clocks are *synchronized*, and their states *real valued*. For this setup, there has been an extensive and growing literature; many basic results can be found in early work [9, 42, 51, 56, 61, 65, 79]. A common feature of the distributed algorithms developed in these references is that individual agents are assumed to execute their local protocols in a synchronized fashion (in the sense that they can act all together at an arbitrary specified time), and be able to simultaneously sense and/or communicate with all the neighbors within their interaction range. In particular, Olfati-Saber and Murray [61] studied algorithms of such type to achieve average consensus on digraphs, and justified that a *balanced* and *strongly connected* topology is necessary and sufficient to guarantee convergence.

We generalize the result of [61] by proposing a novel synchronous algorithm, and prove that it guarantees state averaging on *arbitrary* strongly connected digraphs. In particular, the balanced topological requirement in [61] is dropped, and hence individual agents need not maintain identical amounts of flow-in and flow-out information. The primary challenge of average consensus on arbitrary strongly connected digraphs lies in that the state sum of the agents cannot be preserved in general, thereby causing

a shift in the average value. To handle this problem, the key novelty in our approach is to augment an additional variable for each agent, which we call "surplus", whose function is to record every state change of the associated agent; thus in effect, these variables locally maintain the information of the amount of average shift. Surpluses are then communicated among peers across the network, and provided that the topology is strongly connected, surplus information is accessible, either directly or indirectly, by all the component agents.

We notice that references [7, 46] presented an alternative way of employing an auxiliary variable to achieve averaging on general digraphs. The idea is based on computing the stationary distribution for the Markov chain characterized by the agent network. We note however that the algorithm is quite different from that of consensus type. By contrast, our algorithm to be designed is consensus based, with additional surplus variables to keep track of state updates. On the other hand, our algorithm differs also from the usual ones [9, 42, 51, 56, 61, 65, 79] in that the associated matrices contain negative entries. Consequently for our analysis tools, besides nonnegative matrix theory and algebraic graph theory, it is found that the *matrix perturbation theory* is instrumental in analyzing the convergence properties. Specifically, the surpluses used in updating the states will be viewed as a perturbation term, and it turns out that the states, being suitably perturbed, will eventually average out. Finally, in [30, 31] the authors proposed a *broadcast gossip* algorithm, with an additional variable augmented, to achieve average consensus on general digraphs. The augmented variable is similar in mechanism to surplus, but the convergence to average was not proved; by contrast, we provide a rigorous justification for our convergence results.

The rest of this chapter is organized as follows. In Section 3.2 we formulate the distributed average consensus problem. In Sections 3.3 and 3.4 we present our novel solution algorithm, and justify that it guarantees state averaging on general strongly connected digraphs. Further, in Section 3.5 we explore certain special graph topologies and in Section 3.6 we provide numerical examples for demonstration.

3.2 Problem Formulation

We model a network of n (> 1) agents by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Each *node* in $\mathcal{V} = \{1, ..., n\}$ stands for an agent, and each directed *edge* (j, i) in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes that agent j communicates to agent i (namely, the information flow is from j to i). In this chapter we assume that the digraph $\mathcal{G}$ is static; that is, the communication structure among agents is fixed. We will study dynamic, or time-varying, digraphs in subsequent chapters.

At time $k \in \mathbb{Z}_+$ each agent $i \in \mathcal{V}$ has a scalar state $x_i(k) \in \mathbb{R}$; the aggregate state is denoted by $x(k) = [x_1(k) \cdots x_n(k)]^T \in \mathbb{R}^n$. The *average consensus problem* aims at designing distributed algorithms, where individual agents update their states using only the local information of their neighboring agents in the digraph $\mathcal{G}$ such that every state $x_i(k)$ eventually converges to the initial average value $x_a := \mathbf{1}^T x(0)/n$.

To achieve state averaging on general digraphs, the main difficulty is that the state sum $\mathbf{1}^T x$ need not remain invariant, which can result in losing track of the initial average x_a. To deal with this problem, we propose associating to each agent i an additional variable $s_i(k) \in \mathbb{R}$, called *surplus*; write $s(k) = [s_1(k) \cdots s_n(k)]^T \in \mathbb{R}^n$ and set $s(0) = 0$. The function of surplus is to locally record the state changes of individual agents such that $\mathbf{1}^T(x(k) + s(k)) = \mathbf{1}^T x(0)$ for all time k; in other words, surplus keeps the quantity $\mathbf{1}^T(x + s)$ constant over time. The rules of how to utilize and communicate surplus mark the distinctive feature of our averaging algorithm compared to those in the literature [9, 42, 51, 56, 61, 65, 79], as detailed in Section 3.3.

Definition 3.1. A network of agents achieves *average consensus* if for every initial condition $(x(0), 0)$, it holds that $(x(k), s(k)) \to (x_a \mathbf{1}, 0)$ as $k \to \infty$.

Problem 3.1. Design a distributed algorithm such that the agents achieve average consensus on general (strongly connected) digraphs.

To solve this problem, we will propose in Section 3.3 a surplus-based distributed algorithm, under which we will justify in Section 3.4 that average consensus is achieved for general digraphs.

3.3 Distributed Algorithm

In this section, we first review the standard (discrete-time) consensus algorithm in the literature [9, 42, 79]. We then propose our novel algorithm based on surplus, which may be seen as an extension of the standard one.

3.3.1 Standard Algorithm

Consider a system of n agents each interacting with neighbors with the following protocol [9, 42, 79]:

$$x_i(k+1) = x_i(k) + \sum_{j \in \mathcal{N}_i^+} a_{ij}(x_j(k) - x_i(k)), \quad i \in \mathcal{V}. \tag{3.1}$$

Here $\mathcal{N}_i^+$ is the set of *in-neighbors* of agent i, and a_{ij} are the *updating weights* such that $a_{ij} \in (0,1)$ if $j \in \mathcal{N}_i^+$, $a_{ij} = 0$ otherwise, and $\sum_{j \in \mathcal{N}_i^+} a_{ij} < 1$.

Define the *adjacency matrix* A of the digraph $\mathcal{G}$ by

$$A := [a_{ij}] \in \mathbb{R}^{n \times n}; \tag{3.2}$$

namely the entries of A correspond to the updating weights. Then define the *degree matrix* D by

$$D := \mathrm{diag}(d_1, \ldots, d_n), \text{ where } d_i := \sum_{j=1}^{n} a_{ij}. \tag{3.3}$$

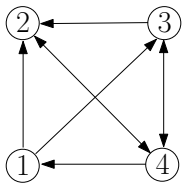
Thus the *Laplacian matrix* L is defined to be

$$L := D - A. \tag{3.4}$$

One observes that L's diagonal entries are nonnegative, off-diagonal entries nonpositive, and row sums zero. With these matrices defined, the collective dynamics of n agents can be written in a matrix form as

$$x(k+1) = (I - L)x(k). \tag{3.5}$$

We henceforth refer to Equation (3.5) the standard algorithm. It is easily seen that the updating matrix $I - L$ is nonnegative (by $\sum_{j \in \mathcal{N}_i^+} a_{ij} < 1$), and every row sums up to one; i.e., $I - L$ is *row stochastic*.



$$\begin{aligned} \mathcal{N}_1^+ &= \{4\} & \mathcal{N}_1^- &= \{2,3\} \\ \mathcal{N}_2^+ &= \{1,3,4\} & \mathcal{N}_2^- &= \{4\} \\ \mathcal{N}_3^+ &= \{1,4\} & \mathcal{N}_3^- &= \{2,4\} \\ \mathcal{N}_4^+ &= \{2,3\} & \mathcal{N}_4^- &= \{1,2,3\} \end{aligned}$$

Figure 3.1: Illustrating example of 4 agents: topology $\mathcal{G}$ and neighbor sets.

Definition 3.2. A digraph $\mathcal{G}$ is *balanced* if for every $i \in \mathcal{V}$, $\sum_{j=1}^n a_{ij} = \sum_{j=1}^n a_{ji}$.

One readily verifies that the digraph $\mathcal{G}$ is balanced if and only if the matrix $I - L$ is *column stochastic* (therefore *doubly stochastic*). For the standard algorithm (3.5), the fundamental convergence result is [61, 79]: A network of agents achieves average consensus if and only if the digraph $\mathcal{G}$ is both strongly connected and balanced.

Remark 3.1. "Balanced", together with "strongly connected", provides a clean characterization for average consensus under the standard algorithm (3.5). The balanced requirement on topology may very well be strong, however, because it demands that every agent keeps precisely identical amounts of input and output information. Even though one supposes that a balanced network might be designed off line, the balance could be easily destroyed by time delay, packet loss, or link failure in practice. Nevertheless a balanced topology is necessary for the algorithm (3.5) to average; therefore we should, and will, explore a new algorithm, one that does not rely on the balanced property to achieve average consensus.

Example 3.1. We illustrate that under an unbalanced digraph, the standard algorithm (3.5) fails to achieve average consensus. Consider the network of 4 agents with topology $\mathcal{G}$ and neighbor sets displayed in Fig. 3.1. Fixing $i \in [1,4]$, let $a_{ij} = 1/\big(\text{card}(\mathcal{N}_i^+) + 1\big)$ for every $j \in \mathcal{N}_i^+$. One verifies that $\mathcal{G}$ is not balanced, and computes the matrices L and $I - L$ as follows:

$$L = \begin{bmatrix} 1/2 & 0 & 0 & -1/2 \\ -1/4 & 3/4 & -1/4 & -1/4 \\ -1/3 & 0 & 2/3 & -1/3 \\ 0 & -1/3 & -1/3 & 2/3 \end{bmatrix}, \quad I - L = \begin{bmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/3 & 0 & 1/3 & 1/3 \\ 0 & 1/3 & 1/3 & 1/3 \end{bmatrix}.$$

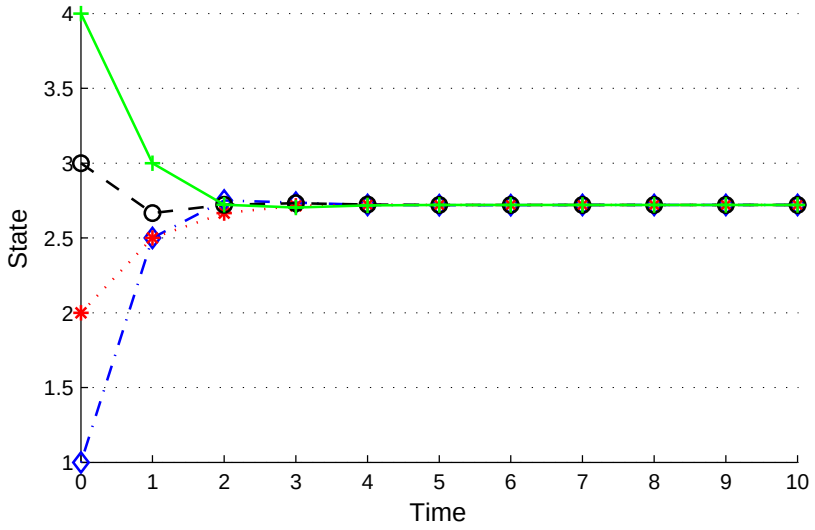


Figure 3.2: Standard algorithm (3.5) fails to achieve average consensus on unbalanced topology.

Observe that $I - L$ is row stochastic, but not column stochastic. Now consider the initial state $x(0) = [1\ 2\ 3\ 4]^T$; so the desired average is $x_a = 2.5$. As displayed in Fig. 3.2, one sees that the states indeed reach a consensus, the value of which is, however, 2.72.

3.3.2 Surplus-Based Algorithm

Having revealed the dependence on balanced topology of the standard algorithm (3.5), we embark on the design of a novel distributed algorithm, with the purpose to achieve average consensus on arbitrary strongly connected digraphs.

Consider again the system of n agents represented by the digraph $\mathcal{G}$. There are three operations that every node $i \in \mathcal{V}$ performs at time $k \in \mathbb{Z}_+$.

(1) First (sending stage), node i sends its state information $x_i(k)$ and weighted surplus $b_{ih}s_i(k)$ to each *out-neighbor* $h \in \mathcal{N}_i^-$; here the *sending weight* b_{ih} is such that $b_{ih} \in (0, 1)$ if $h \in \mathcal{N}_i^-$, $b_{ih} = 0$ otherwise, and $\sum_{h\in\mathcal{N}_i^-} b_{ih} < 1$.

(2) Second (receiving stage), node i receives state information $x_j(k)$ and weighted surplus $b_{ji}s_j(k)$ from each in-neighbor $j \in \mathcal{N}_i^+$.

(3) Third (updating stage), node i updates its own state $x_i(k)$ and surplus $s_i(k)$ as follows:

$$x_i(k+1) = x_i(k) + \sum_{j\in\mathcal{N}_i^+} a_{ij}(x_j(k) - x_i(k)) + \epsilon s_i(k), \tag{3.6}$$

$$s_i(k+1) = \Big((1 - \sum_{h\in\mathcal{N}_i^-} b_{ih})s_i(k) + \sum_{j\in\mathcal{N}_i^+} b_{ji}s_j(k)\Big) - \Big(x_i(k+1) - x_i(k)\Big). \tag{3.7}$$

Here a_{ij} are the updating weights, and the parameter ϵ is a positive number which adjusts the amount of surplus used to update the state.

Let $B := [b_{ih}]^T \in \mathbb{R}^{n\times n}$, where the entries are the sending weights (note that the transpose in the notation is needed because $h \in \mathcal{N}_i^-$ for b_{ih}). Define the matrix $S := (I - \tilde{D}) + B$, where $\tilde{D} = \mathrm{diag}(\tilde{d}_1, \ldots, \tilde{d}_n)$ with $\tilde{d}_i = \sum_{h=1}^n b_{ih}$. One verifies that S is nonnegative (by $\sum_{h\in\mathcal{N}_i^-} b_{ih} < 1$), and every column sums up to one; i.e., S is column stochastic. As can be observed from (3.7), the matrix S captures the part of update induced by sending and receiving surplus. Now write the iteration of states (3.6) and surpluses (3.7) in a matrix form as

$$\begin{bmatrix} x(k+1) \\ s(k+1) \end{bmatrix} = M \begin{bmatrix} x(k) \\ s(k) \end{bmatrix}, \quad \text{where } M := \begin{bmatrix} I - L & \epsilon I \\ L & S - \epsilon I \end{bmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{3.8}$$

Notice that (i) the matrix M has negative entries due to the presence of the Laplacian matrix L in the $(2,1)$-block; (ii) the column sums of M are equal to one, which implies that the quantity $x(k) + s(k)$ is a constant for all $k \in \mathbb{Z}_+$; and (iii) the state evolution specified by the $(1,1)$-block of M is the standard consensus algorithm (3.5). We henceforth refer to (3.8) as the surplus-based algorithm, and will analyze its convergence properties in Section 3.4.

Example 3.2. We demonstrate that the surplus-based algorithm (3.8) achieves average consensus on the unbalanced topology in Fig. 3.1. Fixing $i \in [1,4]$, let a_{ij} be the same as in Example 3.1, and $b_{ih} = 1/\big(\mathrm{card}(\mathcal{N}_i^-) + 1\big)$ for every $h \in \mathcal{N}_i^-$. One

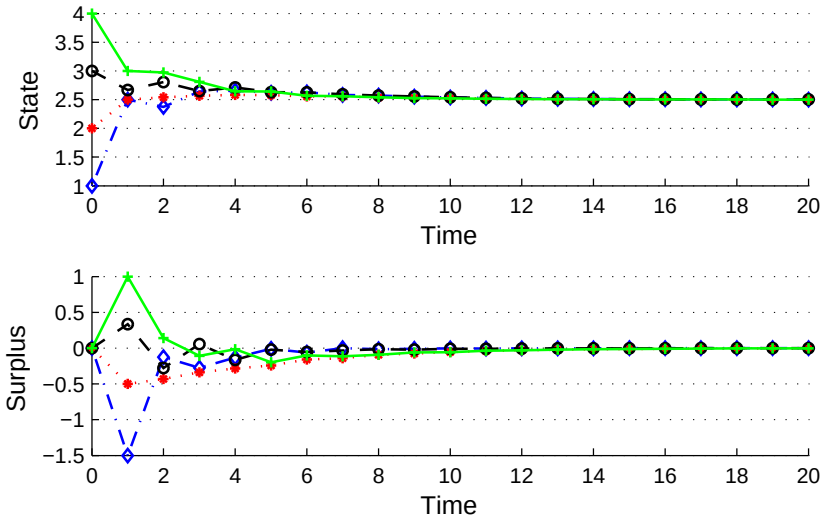


Figure 3.3: Surplus-based algorithm (3.8) achieves average consensus on unbalanced topology.

computes that the matrix S as follows:

$$S = \begin{bmatrix} 1/3 & 0 & 0 & 1/4 \\ 1/3 & 1/2 & 1/3 & 1/4 \\ 1/3 & 0 & 1/3 & 1/4 \\ 0 & 1/2 & 1/3 & 1/4 \end{bmatrix};$$

so S is column stochastic. Then the matrix M is as follows:

$$M = \left[\begin{array}{cccc|cccc} 1/2 & 0 & 0 & 1/2 & \epsilon & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & \epsilon & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & \epsilon & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & \epsilon \\ \hline 1/2 & 0 & 0 & -1/2 & 1/3-\epsilon & 0 & 0 & 1/4 \\ -1/4 & 3/4 & -1/4 & -1/4 & 1/3 & 1/2-\epsilon & 1/3 & 1/4 \\ -1/3 & 0 & 2/3 & -1/3 & 1/3 & 0 & 1/3-\epsilon & 1/4 \\ 0 & -1/3 & -1/3 & 2/3 & 0 & 1/2 & 1/3 & 1/4-\epsilon \end{array}\right].$$

Observe that M has negative entries, and every column sums up to one. Set the parameter $\epsilon = 0.25$ and consider again the initial state $x(0) = [1\ 2\ 3\ 4]^T$; Fig. 3.3

shows that every state converges to the desired average value 2.5, and every surplus vanishes.

3.4 Convergence Results on General Digraphs

We present the central result of this chapter.

Theorem 3.1. Using the surplus-based algorithm (3.8) with the parameter $\epsilon > 0$ sufficiently small, a network of agents achieves average consensus if and only if the digraph $\mathcal{G}$ is strongly connected.

In other words, our new algorithm does not rely on balanced topology, as contrasted with the standard algorithm (3.5). As a result, the graphical condition ensuring average consensus is generalized to *arbitrary strongly connected* – a primary contribution of this thesis. We should note, however, the assumption that ϵ is sufficiently small, which will be seen crucial to guarantee convergence.

Before providing the proof of Theorem 3.1, we state a necessary and sufficient condition for average consensus in terms of the spectrum of the matrix M.

Proposition 3.1. The surplus-based algorithm (3.8) achieves average consensus if and only if 1 is a simple eigenvalue of M, and all other eigenvalues have moduli smaller than one.

Proof. (Sufficiency) Since every column of M sums up to one, 1 is an eigenvalue of M and $[\mathbf{1}^T\ \mathbf{1}^T]^T$ is a corresponding left eigenvector. Note also that $M[\mathbf{1}^T\ 0]^T = [\mathbf{1}^T\ 0]^T$; so $[\mathbf{1}^T\ 0]^T \in \mathbb{R}^{2n}$ is a right eigenvector corresponding to the eigenvalue 1. Write M in Jordan canonical form as

$$M = VJV^{-1} = \begin{bmatrix} y_1 & \cdots & y_{2n} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & J' \end{bmatrix} \begin{bmatrix} z_1^T \\ \vdots \\ z_{2n}^T \end{bmatrix},$$

where $y_i,\ z_i \in \mathbb{C}^{2n}$, $i \in [1, 2n]$, are respectively the (generalized) right and left eigenvectors of M; and $J' \in \mathbb{C}^{(2n-1)\times(2n-1)}$ contains the Jordan block matrices corresponding to those eigenvalues with moduli smaller than one. For the eigenvalue 1 choose

$y_1 = [\mathbf{1}^T\ 0]^T$ and $z_1 = (1/n)[\mathbf{1}^T\ \mathbf{1}^T]^T$; thus $z_1^T y_1 = 1$. Now the kth power of M is

$$M^k = VJ^kV^{-1} = V\begin{bmatrix} 1 & 0 \\ 0 & (J')^k \end{bmatrix} V^{-1} \to y_1 z_1^T = \begin{bmatrix} \frac{1}{n}\mathbf{1}\mathbf{1}^T & \frac{1}{n}\mathbf{1}\mathbf{1}^T \\ 0 & 0 \end{bmatrix}, \quad \text{as } k \to \infty.$$

Therefore

$$\begin{bmatrix} x(k) \\ s(k) \end{bmatrix} = M^k \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} \to \begin{bmatrix} \frac{1}{n}\mathbf{1}\mathbf{1}^T & \frac{1}{n}\mathbf{1}\mathbf{1}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(0) \\ s(0) \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{n}\mathbf{1}\mathbf{1}^T x(0) \\ 0 \end{bmatrix} = \begin{bmatrix} x_a \mathbf{1} \\ 0 \end{bmatrix}, \quad \text{as } k \to \infty.$$

(Necessity) First we claim that the eigenvalue 1 of M is always simple. Suppose on the contrary that the algebraic multiplicity of 1 equals two. The corresponding geometric multiplicity, however, equals one; this can be shown by verifying $\text{rank}(M - I) = 2n - 1$. Thus there exists a generalized right eigenvector $u = [u_1^T\ u_2^T]^T \in \mathbb{R}^{2n}$ such that $(M - I)^2 u = 0$, and $(M - I)u$ is a right eigenvector with respect to the eigenvalue 1. Since $[\mathbf{1}^T\ 0]^T$ is also a right eigenvector corresponding to the eigenvalue 1, it must hold that

$$\begin{aligned} & (M - I)u = c[\mathbf{1}^T\ 0]^T, \quad \text{for some scalar } c \neq 0 \\ \Rightarrow\ & \begin{bmatrix} -L & \epsilon I \\ L & S - I - \epsilon I \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = c \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \\ \Rightarrow\ & \begin{cases} -Lu_1 + \epsilon u_2 = c\mathbf{1} \\ Lu_1 + (S - I)u_2 - \epsilon u_2 = 0 \end{cases} \\ \Rightarrow\ & (S - I)u_2 = c\mathbf{1}. \end{aligned}$$

One may verify that $\text{rank}(S - I) = n - 1$ but $\text{rank}([S - I\ \ c\mathbf{1}]) = n$. Hence there is no solution for u_2, which in turn implies that the generalized right eigenvector u cannot exist. This proves our claim.

Now suppose that there is an eigenvalue λ of M such that $\lambda \neq 1$ and $|\lambda| \geq 1$. But this immediately implies that $\lim_{k\to\infty} M^k$ does not exist (cf. [79]). Therefore, average consensus cannot be achieved. ■

We are ready to prove Theorem 3.1. The necessity part will justify that the class of strongly connected digraphs characterizes the existence of any distributed algorithm that can solve average consensus. For the sufficiency part, let

$$M_0 := \begin{bmatrix} I - L & 0 \\ L & S \end{bmatrix} \quad \text{and} \quad E := \begin{bmatrix} 0 & I \\ 0 & -I \end{bmatrix}. \tag{3.9}$$

Then $M = M_0 + \epsilon E$, and we view M as being obtained by "perturbing" M_0 via the term ϵE. Concretely, we show that the eigenvalues λ_i of the unperturbed matrix M_0 satisfy

$$1 = \lambda_1 = \lambda_2 > |\lambda_3| \geq \cdots \geq |\lambda_{2n}|; \tag{3.10}$$

and that after a small perturbation ϵE, the obtained matrix M has only a simple eigenvalue 1 and all other eigenvalues have moduli smaller than one. Hence by Proposition 3.1, average consensus is achieved. It should be pointed out that, unlike the standard algorithm (3.5), the tools in nonnegative matrix theory cannot be used directly to analyze the spectrum of M owing to the existence of negative entries.

Proof of Theorem 3.1. (Necessity) Suppose that $\mathcal{G}$ is not strongly connected. Then at least one node of $\mathcal{G}$ is not globally reachable. Let $\mathcal{V}_g^*$ denote the set of non-globally reachable nodes, and write its cardinality $\text{card}(\mathcal{V}_g^*) = r,\ r \in [1, n]$. If $r = n$, i.e. $\mathcal{G}$ does not have a globally reachable node, then $\mathcal{G}$ has at least two distinct closed strong components (by Lemma 2.1). In this case, if the nodes in different components have different initial states, then average consensus cannot be achieved. It is left to consider $r < n$. Let $\mathcal{V}_g := \mathcal{V} - \mathcal{V}_g^*$ denote the set of all globally reachable nodes; thus $\mathcal{V}_g$ is the unique closed strong component in $\mathcal{G}$ (again by Lemma 2.1). Consider an initial condition $(x(0), 0)$ such that all nodes in $\mathcal{V}_g$ have the same state $c \in \mathbb{R}$, and not all the states of the nodes in $\mathcal{V}_g^*$ equal c. Hence $x_a \neq c$. But no state or surplus update is possible for the nodes in $\mathcal{V}_g$ because it is closed, and therefore average consensus cannot be achieved.

(Sufficiency) First, we prove the assertion (3.10). Since M_0 is block (lower) triangular, its spectrum is $\sigma(M_0) = \sigma(I - L) \cup \sigma(S)$. Recall that the matrices $I - L$ and S are row and column stochastic, respectively; so their spectral radii satisfy

$\rho(I-L)=\rho(S)=1$. Now owing to that $\mathcal{G}$ is strongly connected, $I-L$ and S are both irreducible by Lemma 2.3. Thus it follows from the Perron-Frobenius Theorem (Lemma 2.4) that $\rho(I-L)$ (resp. $\rho(S)$) is a simple eigenvalue of $I-L$ (resp. S). This implies (3.10).

Next, we will qualify the changes to the two eigenvalues $\lambda_1=\lambda_2=1$ of M_0 under a small perturbation ϵE. For this we need to find their corresponding left and right eigenvectors. One may quickly verify $\text{rank}(M_0-I)=2n-2$, and hence the geometric multiplicity of this eigenvalue 1 equals two. Thus the matrix M_0 can be written in the following Jordan canonical form:

$$M_0=VJV^{-1}=\begin{bmatrix} y_1 & y_2 & \cdots & y_{2n}\end{bmatrix}\left[\begin{array}{cc|c} \lambda_1 & 0 & \\ 0 & \lambda_2 & 0 \\ \hline & & \\ 0 & & J' \\ & & \end{array}\right]\begin{bmatrix} z_1^T \\ z_2^T \\ \vdots \\ z_{2n}^T\end{bmatrix},$$

where $y_i,\ z_i\in\mathbb{C}^{2n}$, $i\in[1,2n]$, are respectively the (generalized) right and left eigenvectors of M_0; and $J'\in\mathbb{C}^{(2n-2)\times(2n-2)}$ contains the Jordan block matrices corresponding to $\lambda_3,\ldots,\lambda_{2n}$. Also write $J_0:=\text{diag}(\lambda_1,\lambda_2)$. Choose

$$Y:=\begin{bmatrix} y_1 & y_2\end{bmatrix}=\begin{bmatrix} 0 & \mathbf{1} \\ v_2 & -nv_2\end{bmatrix},\quad Z:=\begin{bmatrix} z_1^T \\ z_2^T\end{bmatrix}=\begin{bmatrix} \mathbf{1}^T & \mathbf{1}^T \\ v_1^T & 0\end{bmatrix}.$$

Here $v_1\in\mathbb{R}^n$ is a left eigenvector of $I-L$ with respect to $\rho(I-L)$ such that it is positive and scaled to satisfy $v_1^T\mathbf{1}=1$; and $v_2\in\mathbb{R}^n$ is a right eigenvector of S corresponding to $\rho(S)$ such that it is positive and scaled to satisfy $\mathbf{1}^Tv_2=1$. The fact that positive eigenvectors v_1 and v_2 exist follows again from the Perron-Frobenius Theorem (Lemma 2.4). With this choice one may check that $ZY=I$.

We now proceed to a perturbation analysis. It is well known that the eigenvalues of a matrix are continuous functions of its entries ([10, Section VI.1], [71, Section IV.1]). Hence for sufficiently small $\epsilon>0$, there must exist exactly two eigenvalues $\lambda_1(\epsilon)$, $\lambda_2(\epsilon)$ of M corresponding respectively to λ_1, λ_2 of M_0. Write $J_0(\epsilon):=\text{diag}(\lambda_1(\epsilon),\lambda_2(\epsilon))$,

and denote the associated right eigenvectors by $Y(\epsilon) = \begin{bmatrix} y_{11}(\epsilon) & y_{12}(\epsilon) \\ y_{21}(\epsilon) & y_{22}(\epsilon) \end{bmatrix}$. It can be verified that $J_0(\epsilon) \to J_0$ and $Y(\epsilon) \to Y$ as $\epsilon \to 0$ [69, Section 2.8]. Then

$$
\begin{aligned}
& MY(\epsilon) = Y(\epsilon)J_0(\epsilon) \\
\Rightarrow\ & M_0X(\epsilon) + \epsilon EX(\epsilon) = Y(\epsilon)J_0(\epsilon) && \text{(by } M = M_0 + \epsilon E\text{)} \\
\Rightarrow\ & ZM_0Y(\epsilon) + \epsilon ZEY(\epsilon) = ZY(\epsilon)J_0(\epsilon) && \text{(left multiplying } Z \text{ on both sides)} \\
\Rightarrow\ & ZY(\epsilon)J_0(\epsilon) - J_0ZY(\epsilon) = \epsilon ZEY(\epsilon) && \text{(by } ZM_0 = J_0Z\text{)} \\
\Rightarrow\ & \begin{bmatrix} (\lambda_1(\epsilon) - \lambda_1)\mathbf{1}^T(y_{11}(\epsilon) + y_{21}(\epsilon)) & (\lambda_2(\epsilon) - \lambda_1)\mathbf{1}^T(y_{12}(\epsilon) + y_{22}(\epsilon)) \\ (\lambda_1(\epsilon) - \lambda_2)v_1^Ty_{11}(\epsilon) & (\lambda_2(\epsilon) - \lambda_2)v_1^Ty_{12}(\epsilon) \end{bmatrix} \\
& = \epsilon \begin{bmatrix} 0 & 0 \\ v_1^Ty_{21}(\epsilon) & v_1^Ty_{22}(\epsilon) \end{bmatrix}.
\end{aligned}
$$

Equating the $(2,2)$ entries on both sides gives

$$v_1^Ty_{12}(\epsilon)\frac{\lambda_2(\epsilon) - \lambda_2}{\epsilon} = v_1^Ty_{22}(\epsilon).$$

Let $\epsilon \to 0$ in the above equation. Then the left hand side is $v_1^Ty_{12}(\epsilon)(\lambda_2(\epsilon) - \lambda_2)/\epsilon \to v_1^T\mathbf{1}\dot{\lambda_2} = \dot{\lambda_2}$, where the derivative $\dot{\lambda_2}$ is given by $\dot{\lambda_2} := \lim_{\epsilon\to 0}(\lambda_2(\epsilon) - \lambda_2)/\epsilon$; the existence of this derivative is due to that for the eigenvalue $(\lambda_1 = \lambda_2 =)1$ of M_0, its algebraic multiplicity equals its geometric multiplicity [69]. On the other hand, the right hand side is $v_1^Ty_{22}(\epsilon) \to -nv_1^Tv_2$. As v_1 and v_2 are positive vectors, we have $\dot{\lambda_2} < 0$, which indicates that the eigenvalue $\lambda_2(\epsilon)$ moves to the left along the real axis when ϵ is small. Likewise, equating the $(1,1)$ entries on both sides yields $\dot{\lambda_1} := \lim_{\epsilon\to 0}(\lambda_1(\epsilon) - \lambda_1)/\epsilon = 0$, showing that the eigenvalue $\lambda_1(\epsilon)$ stays put. One may also check that the $(1,2)$ and $(2,1)$ entries on both sides vanish as $\epsilon \to 0$. Hence by continuity there must exist a positive δ_1 such that $\lambda_1(\delta_1) = 1$ and $\lambda_2(\delta_1) < 1$. On the other hand, by the eigenvalue continuity there exists a positive δ_2 such that $|\lambda_i(\delta_2)| < 1$ for all $i \in [3, 2n]$. Thus for any sufficiently small $\epsilon \in (0, \min\{\delta_1, \delta_2\})$, the matrix M has a simple eigenvalue 1 and all other eigenvalues have moduli smaller than one. Therefore, from Proposition 3.1, the conclusion that average consensus is achieved follows. ■

Remark 3.2. (Convergence Speed) Assuming that the surplus-based algorithm (3.8) converges to the initial average, the speed of its convergence is governed by the second largest (in modulus) eigenvalue of the updating matrix M. We denote this particular eigenvalue by λ_2^*, and refer to it as the *convergence factor* of the algorithm (3.8). Note that $\lambda_2^* < 1$ is equivalent to average consensus (by Proposition 3.1); and the value of λ_2^* depends not only on the digraph topology $\mathcal{G}$ but also on the algorithm parameter ϵ. We will illustrate this latter point in Section 3.6.

3.4.1 Bound on Parameter ϵ

Having shown that the surplus-based algorithm (3.8) solves average consensus for sufficiently small parameter ϵ, we now aim to give an upper bound on ϵ. For this, we borrow a fact from matrix perturbation theory (e.g., [10,71]) which relates the size of ϵ to the distance between perturbed and unperturbed eigenvalues. Below is the main result of this investigation.

Proposition 3.2. Suppose that the digraph $\mathcal{G}$ is strongly connected. The surplus-based algorithm (3.8) achieves average consensus if the parameter ϵ satisfies $\epsilon \in (0, \bar{\epsilon})$, where

$$\bar{\epsilon} := \frac{1}{(20+8n)^n}\,(1-|\lambda_3|)^n, \quad \text{with } \lambda_3 \text{ as in (3.10).} \tag{3.11}$$

We stress that the above bound $\bar{\epsilon}$ ensures average consensus for arbitrary strongly connected topologies. Due to the power n, however, the bound is rather conservative. This power is unavoidable for any perturbation bound result with respect to general matrices, as is well known in matrix perturbation literature [10,71]. In Section 3.5, we will exploit structures of some special topologies, which yield less conservative bounds on ϵ.

Some preliminaries will be presented first, based on which Proposition 3.2 follows immediately. Henceforth in this subsection, the digraph $\mathcal{G}$ is assumed to be strongly connected. We begin by introducing a metric for the distance between the spectrums of M_0 and M; here $M = M_0 + \epsilon E$, with M_0 and E in (3.9). Let $\sigma(M_0) := \{\lambda_1, \ldots, \lambda_{2n}\}$

(where the numbering is the same as that in (3.10)) and $\sigma(M) := \{\lambda_1(\epsilon), \ldots, \lambda_{2n}(\epsilon)\}$. The *optimal matching distance* $d\left(\sigma(M_0), \sigma(M)\right)$ [10, 71] is defined by

$$d\left(\sigma(M_0), \sigma(M)\right) := \min_{\pi} \max_{i \in [1,2n]} |\lambda_i - \lambda_{\pi(i)}(\epsilon)|, \tag{3.12}$$

where π is taken over all permutations of $\{1, \ldots, 2n\}$. Thus if we draw $2n$ identical circles centered respectively at $\lambda_1, \ldots, \lambda_{2n}$, then $d\left(\sigma(M_0), \sigma(M)\right)$ is the smallest radius such that these circles include all $\lambda_1(\epsilon), \ldots, \lambda_{2n}(\epsilon)$. Here is an upper bound on the optimal matching distance [10, Theorem VIII.1.5].

Lemma 3.1. $d\left(\sigma(M_0), \sigma(M)\right) \leq 4\ (||M_0||_\infty + ||M||_\infty)^{1-1/n}\ ||\epsilon E||_\infty^{1/n}$.

Next, we are concerned with the eigenvalues $\lambda_3(\epsilon), \ldots, \lambda_{2n}(\epsilon)$ of M.

Lemma 3.2. If the parameter $\epsilon \in (0, \bar{\epsilon})$ with $\bar{\epsilon}$ in (3.11), then $|\lambda_3(\epsilon)|, \ldots, |\lambda_{2n}(\epsilon)| < 1$.

Proof. Since $L = D - A$ and $S = (I - \tilde{D}) + B$, one can compute $||L||_\infty = 2 \max_{i \in [1,n]} d_i < 2$ and $||S||_\infty < n$. Then $||M_0||_\infty \leq ||L||_\infty + ||S||_\infty < 2 + n$ and $||E||_\infty \leq 1$. By Lemma 3.1,

$$\begin{aligned} d\left(\sigma(M_0), \sigma(M)\right) &\leq 4\ (2||M_0||_\infty + \epsilon||E||_\infty)^{1-1/n}\ (\epsilon||E||_\infty)^{1/n} \\ &< 4\ (4 + 2n + \epsilon)^{1-1/n}\ \epsilon^{1/n} \\ &< 4\ (4 + 2n + \epsilon)\ \epsilon^{1/n} < 1 - |\lambda_3|. \end{aligned}$$

The last inequality is due to $\epsilon < \bar{\epsilon}$ in (3.11). Now recall from the proof of Theorem 3.1 that the unperturbed eigenvalues $\lambda_3, \ldots, \lambda_{2n}$ of M_0 lie strictly inside the unit circle; in particular, (3.10) holds. Therefore, perturbing the eigenvalues $\lambda_3, \ldots, \lambda_{2n}$ by an amount less than $\bar{\epsilon}$, the resulting eigenvalues $\lambda_3(\epsilon), \ldots, \lambda_{2n}(\epsilon)$ will remain inside the unit circle. ■

It is left to consider the eigenvalues $\lambda_1(\epsilon)$ and $\lambda_2(\epsilon)$ of M. Since every column sum of M equals one for an arbitrary ϵ, we obtain that 1 is always an eigenvalue of M. Hence $\lambda_1(\epsilon)$ must be equal to 1 for any ϵ. On the other hand, for $\lambda_2(\epsilon)$ the following is true.

Lemma 3.3. If the parameter $\epsilon \in (0, \bar{\epsilon})$ with $\bar{\epsilon}$ in (3.11), then $|\lambda_2(\epsilon)| < 1$.

Proof. First recall from the proof of Theorem 3.1 that $\lambda_2 = 1$ and $\dot{\lambda_2} < 0$; so for sufficiently small $\epsilon > 0$, it holds that $|\lambda_2(\epsilon)| < 1$. Now suppose that there exists $\delta \in (0, \bar{\epsilon})$ such that $|\lambda_2(\delta)| \geq 1$. Owing to the continuity of eigenvalues, it suffices to consider $|\lambda_2(\delta)| = 1$. There are three such possibilities, for each of which we derive a contradiction.

Case 1: $\lambda_2(\delta)$ is a complex number with nonzero imaginary part and $|\lambda_2(\delta)| = 1$. Since M is a real matrix, there must exist another eigenvalue $\lambda_i(\delta)$, for some $i \in [3, 2n]$, such that $\lambda_i(\delta)$ is a complex conjugate of $\lambda_2(\delta)$. Then $|\lambda_i(\delta)| = |\lambda_2(\delta)| = 1$, which is in contradiction to that all the eigenvalues $\lambda_3(\delta), \ldots, \lambda_{2n}(\delta)$ stay inside the unit circle as $\delta \in (0, \bar{\epsilon})$ by Lemma 3.2.

Case 2: $\lambda_2(\delta) = -1$. This implies at least $d\left(\sigma(M_0), \sigma(M)\right) = 2$, which contradicts $d\left(\sigma(M_0), \sigma(M)\right) < 1 - |\lambda_3| < 1$ when (3.11) holds.

Case 3: $\lambda_2(\delta) = 1$. This case is impossible because the eigenvalue 1 of M is always simple, as we have justified in the necessity proof of Proposition 3.1. ■

Summarizing Lemmas 3.2 and 3.3, we obtain that if the parameter $\epsilon \in (0, \bar{\epsilon})$ with $\bar{\epsilon}$ in (3.11), then $\lambda_1(\epsilon) = 1$ and $|\lambda_2(\epsilon)|$, $|\lambda_3(\epsilon)|$, …, $|\lambda_{2n}(\epsilon)| < 1$. Therefore, by Proposition 3.1 the surplus-based algorithm (3.8) achieves average consensus; this establishes Proposition 3.2.

3.4.2 Weighted Average Consensus

It may be natural to quest whether the surplus-based algorithm (3.8) can be adapted to achieve consensus on some specified value other than the initial average. Here we show that under mild assumptions, the algorithm (3.8) can be readily modified to compute linear combinations of the initial states.

Consider a vector $\tilde{v} \in \mathbb{R}^n$ such that $\tilde{v}_i \neq 0$ for all i and $\mathbf{1}^T\tilde{v} = n$. Let the desired consensus value be $\tilde{x}_a := \tilde{v}^T x(0)/n$; this is a linear combination of the initial states where every one is (more or less) weighted. We say that a network of agents achieves *weighted average consensus* if for every initial condition $(x(0), 0)$, $(x(k), s(k)) \to (\tilde{x}_a\mathbf{1}, 0)$ as $k \to \infty$.

Now assume that each agent i knows its own weight $\tilde{v}_i$, as well as the weights $\tilde{v}_h$ of its out-neighbors $h \in \mathcal{N}_i^-$. Then revise the surplus iteration equation (3.7) as follows:

$$s_i(k+1) = \Big((1 - \frac{1}{\tilde{v}_i} \sum_{h \in \mathcal{N}_i^-} \tilde{v}_h b_{ih}) s_i(k) + \sum_{j \in \mathcal{N}_i^+} b_{ji} s_j(k)\Big) - \Big(x_i(k+1) - x_i(k)\Big); \quad (3.13)$$

thereby we obtain a modified surplus-based algorithm (cf. (3.8)):

$$\begin{bmatrix} x(k+1) \\ s(k+1) \end{bmatrix} = \tilde{M} \begin{bmatrix} x(k) \\ s(k) \end{bmatrix}, \quad \text{where } \tilde{M} := \begin{bmatrix} I - L & \epsilon I \\ L & \tilde{S} - \epsilon I \end{bmatrix}. \quad (3.14)$$

Here the matrix $\tilde{S}$ is such that $\tilde{v}^T \tilde{S} = \tilde{v}^T$.

Corollary 3.1. Using the modified surplus-based algorithm (3.14) with the parameter $\epsilon > 0$ sufficiently small, the agents achieve weighted average consensus if and only if the digraph $\mathcal{G}$ is strongly connected.

Proof. Note that for the updating matrix $\tilde{M}$, 1 is an eigenvalue with a left eigenvector $[\tilde{v}^T \ \tilde{v}^T]^T$. The rest is the same as the proof for Theorem 3.1. ■

To achieve consensus on some more general functions of the initial states, for example the *power or geometric mean* [6, 60] which are nonlinear functions, it seems necessary to introduce corresponding more general protocols than just linear ones in the state and surplus update equations. Such an extension involves exploring new tools for analysis, as in [24]; and it is an interesting direction we would like to pursue.

3.4.3 Continuous-Time Algorithm

We end this section with a brief account for the continuous-time counterpart of the algorithm (3.8). Let $t \in [0, \infty)$ denote the continuous time index. Given a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, for each node $i \in \mathcal{V}$, the updating rule is as follows:

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i^+} a_{ij}(x_j(t) - x_i(t)) + \epsilon s_i(t), \quad (3.15)$$

$$\dot{s}_i(t) = \Big(- \sum_{h \in \mathcal{N}_i^-} b_{ih} s_i(t) + \sum_{j \in \mathcal{N}_i^+} b_{ji} s_j(t)\Big) - \dot{x}_i(t). \quad (3.16)$$

Here the updating weight $a_{ij} > 0$ if $j \in \mathcal{N}_i^+$, and $a_{ij} = 0$ otherwise; the sending weight $b_{ij} > 0$ if $j \in \mathcal{N}_i^-$, and $b_{ij} = 0$ otherwise. Different from those in the algorithm (3.8), the weights here are allowed to be any positive numbers. Also the equations (3.15) and (3.16) may be written in the matrix form as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{s}(t) \end{bmatrix} = M_c \begin{bmatrix} x(t) \\ s(t) \end{bmatrix}, \quad \text{where } M_c = \begin{bmatrix} -L & \epsilon I \\ L & (S-I) - \epsilon I \end{bmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{3.17}$$

Note that the matrix M_c has zero column sums.

Corollary 3.2. Using the continuous-time algorithm (3.17) with the parameter $\epsilon > 0$ sufficiently small, the agents achieve average consensus if and only if the digraph $\mathcal{G}$ is strongly connected.

Proof. The necessity proof is the same as Theorem 3.1. Now let $M_{c,0} := \begin{bmatrix} -L & 0 \\ L & S-I \end{bmatrix}$ and $E := \begin{bmatrix} 0 & I \\ 0 & -I \end{bmatrix}$ as before. Then $M_c = M_{c,0} + \epsilon E$; and for strongly connected digraphs the eigenvalues of $M_{c,0}$ satisfy $\lambda_1 = \lambda_2 = 0$, and $\lambda_3, \ldots, \lambda_{2n}$ with negative real parts. We then conduct an analogous perturbation argument as in the proof of Theorem 3.1, which yields that the matrix M_c has a simple eigenvalue 0 and all other eigenvalues have negative real parts for sufficiently small ϵ. Moreover, corresponding to the simple eigenvalue 0, a right eigenvector is $y_1 = [\mathbf{1}^T \ 0]^T$ and a left eigenvector $z_1 = (1/n)[\mathbf{1}^T \ \mathbf{1}^T]^T$. Therefore

$$\begin{bmatrix} x(t) \\ s(t) \end{bmatrix} = \mathrm{e}^{tM_c} \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} \to y_1 z_1^T \begin{bmatrix} x(0) \\ s(0) \end{bmatrix} = \begin{bmatrix} \frac{1}{n}\mathbf{1}\mathbf{1}^T x(0) \\ 0 \end{bmatrix} = \begin{bmatrix} x_a \mathbf{1} \\ 0 \end{bmatrix}, \quad \text{as } t \to \infty.$$

■

3.5 Special Graph Topologies

We turn now to a special class of topologies – strongly connected and balanced digraphs – and investigate the required upper bound on the parameter ϵ. Furthermore,

when these digraphs are restricted to symmetric or cyclic respectively, we derive less conservative ϵ bounds for the surplus-based algorithm (3.8).

Given a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, its *degree* d is defined by $d := \max_{i\in\mathcal{V}} \text{card}(\mathcal{N}_i^+)$. In the surplus-based algorithm (3.8) choose the updating and sending weights to be respectively $a_{ij} = 1/(2dn)$ and $b_{ij} = 1/(dn)$, for every $(j, i) \in \mathcal{E}$. This choice renders the two matrices $I - 2L$ and S identical, when the digraph $\mathcal{G}$ is balanced. We will see that the equality $I - 2L = S$ supports a similarity transformation in dealing with the cyclic case below.

Lemma 3.4. Suppose that the parameter ϵ satisfies $\epsilon \in (0, 2)$, and the zeros of the following polynomial for every $\mu \neq 0$ with $|\mu - 1/(2n)| \leq 1/(2n)$ lie strictly inside the unit circle:

$$p(\lambda) := \lambda^2 + \alpha_1\lambda + \alpha_0, \tag{3.18}$$

where $\alpha_0 := 2\mu^2 - 3\mu - \epsilon + 1$, $\alpha_1 := 3\mu + \epsilon - 2$. Then the surplus-based algorithm (3.8) achieves average consensus on strongly connected and balanced digraphs.

Proof. We analyze the spectral properties of the matrix M in terms of those of the Laplacian matrix L. Let μ_i, $i = 1, \ldots, n$, be the ith eigenvalue of L. Since $\mathcal{G}$ is balanced and all the updating weights are $a_{ij} = 1/(2dn)$, it follows from the Gershgorin Theorem (Lemma 2.2) that $|\mu_i - 1/(2n)| \leq 1/(2n)$. Further, as $\mathcal{G}$ is strongly connected, by the Perron-Frobenius Theorem (Lemma 2.4) we get that $\mu_1 = 0$ is simple. Now substituting the equality $S = I - 2L$ into (3.8) one obtains

$$M = \begin{bmatrix} I - L & \epsilon I \\ L & I - 2L - \epsilon I \end{bmatrix}.$$

Consider the characteristic polynomial of M:

$$\begin{aligned} \det(\lambda I - M) &= \det\left(\begin{bmatrix} (\lambda - 1)I + L & -\epsilon I \\ -L & (\lambda - 1 + \epsilon)I + 2L \end{bmatrix}\right) \\ &= \det\left(((\lambda - 1)I + L)((\lambda - 1 + \epsilon)I + 2L) - \epsilon L\right) \\ &= \det\left((\lambda - 1)(\lambda - 1 + \epsilon)I + 3(\lambda - 1)L + 2L^2\right). \end{aligned}$$

Here the second equality is due to that $(\lambda-1)I+L$ and $-L$ commute [70]. By spectral mapping one derives that the $2n$ eigenvalues of M can be obtained by solving the following n equations:

$$(\lambda-1)(\lambda-1+\epsilon)+3(\lambda-1)\mu_i+2\mu_i^2=0, \quad i=1,\ldots,n. \tag{3.19}$$

For $\mu_1=0$ we have from (3.19) that $\lambda_1=1$ and $\lambda_2=1-\epsilon$. Since $\epsilon\in(0,2)$, $\lambda_2\in(-1,1)$. Now fix $i\in[2,n]$ so that $\mu_i\neq 0$ and $|\mu_i-1/(2n)|\leq 1/(2n)$. Note that the left hand side of (3.19) can be arranged into the polynomial $p(\lambda)$ in (3.18), whose zeros are inside the unit circle. It follows that 1 is a simple eigenvalue of M, and all other eigenvalues have moduli smaller than one. Therefore, by Proposition 3.1 we conclude that average consensus is achieved. ■

Now we investigate the values of ϵ that ensure the zeros of the polynomial $p(\lambda)$ in (3.18) inside the unit circle, which in turn guarantee average consensus on strongly connected and balanced digraphs by Lemma 3.4. For this, we view the polynomial $p(\lambda)$ as *interval polynomials* [5] by letting μ take any value in the square: $0\leq \mathrm{Re}(\mu)\leq 1/n$, $-1/(2n)\leq \mathrm{Im}(\mu)\leq 1/(2n)$. Applying the bilinear transformation we obtain a new family of interval polynomials:

$$\tilde{p}(\gamma):=(\gamma-1)^2 p\left(\frac{\gamma+1}{\gamma-1}\right)=(1+\alpha_0+\alpha_1)\gamma^2+(2-2\alpha_0)\gamma+(1+\alpha_0-\alpha_1). \tag{3.20}$$

Then by Kharitonov's result for the complex-coefficient case, the stability of $\tilde{p}(\gamma)$ (its zeros have negative real parts) is equivalent to the stability of eight extreme polynomials [5, Section 6.9], which in turn suffices to guarantee that the zeros of $p(\lambda)$ lie strictly inside the unit circle. Checking the stability of eight extreme polynomials results in upper bounds on ϵ in terms of n. This is displayed in Fig. 3.4 as the solid curve. We see that the bounds grow linearly, which is in contrast with the general bound $\bar{\epsilon}$ in Proposition 3.2 that decays exponentially and is known to be conservative. This is due to that, from the robust control viewpoint, the uncertainty of μ in the polynomial coefficients becomes smaller as n increases.

Alternatively, we employ the Jury stability test [44] to derive that the zeros of the

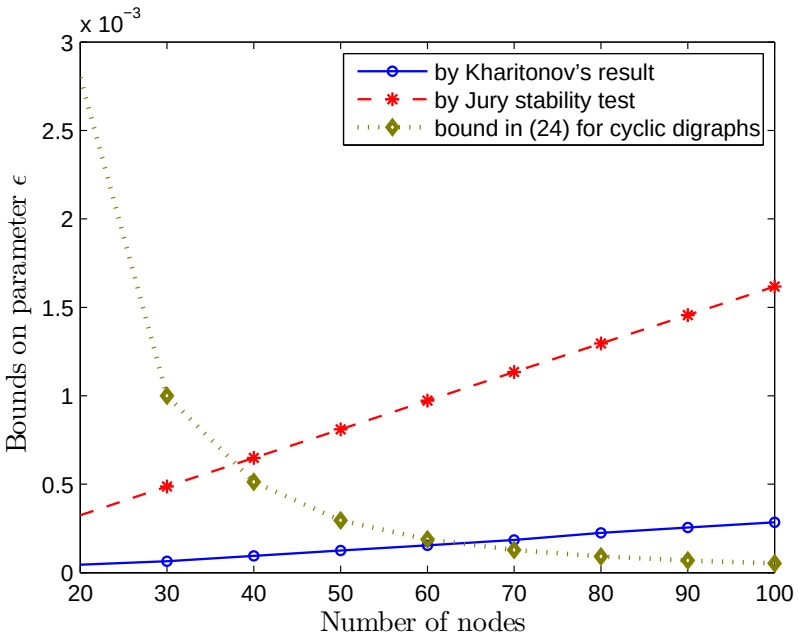


Figure 3.4: Upper bounds on parameter ϵ such that surplus-based algorithm (3.8) achieves average consensus on general strongly connected balanced digraphs (solid and dashed curves) and cyclic digraphs (dotted curve).

polynomial $p(\lambda)$ are strictly inside the unit circle if and only if

$$\beta_0 := \begin{vmatrix} 1 & \alpha_0 \\ \bar{\alpha}_0 & 1 \end{vmatrix} > 0, \quad \beta_1 := \begin{vmatrix} \begin{vmatrix} 1 & \alpha_0 \\ \bar{\alpha}_0 & 1 \end{vmatrix} & \begin{vmatrix} 1 & \alpha_1 \\ \bar{\alpha}_0 & \bar{\alpha}_1 \end{vmatrix} \\ \begin{vmatrix} 1 & \bar{\alpha}_1 \\ \alpha_0 & \alpha_1 \end{vmatrix} & \begin{vmatrix} 1 & \bar{\alpha}_0 \\ \alpha_0 & 1 \end{vmatrix} \end{vmatrix} > 0. \tag{3.21}$$

Here β_0 and β_1 turn out to be polynomials in ϵ of second and fourth order, respectively; the corresponding coefficients are functions of μ and n. Thus selecting μ such that $\mu \neq 0$ and $|\mu - 1/(2n)| \leq 1/(2n)$, we can solve the inequalities in (3.21) for ϵ in terms of n. Thereby we obtain the dashed curve in Fig. 3.4, each plotted point being the minimum value of ϵ over 1000 random samples such that the inequalities in (3.21) hold. This simulation confirms that the true bound on ϵ for the zeros of $p(\lambda)$ to be inside the unit circle is between the solid and dashed curves. Since the discrepancy of these two curves is relatively small, it is suggested that our previous analysis based on Kharitonov's result is not very conservative.

Here ends our discussion on ϵ bounds for arbitrary balanced (and strongly connected) digraphs. In the sequel, we will further specialize the balanced digraph $\mathcal{G}$ to be symmetric or cyclic, respectively, and provide analytic ϵ bounds less conservative than (3.11) for the general case. In particular, the exponent n is not involved.

3.5.1 Connected Undirected Graphs

A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *symmetric* if $(j, i) \in \mathcal{E}$ implies $(i, j) \in \mathcal{E}$. That is, $\mathcal{G}$ is undirected.

Proposition 3.3. Consider a general connected undirected graph $\mathcal{G}$. Then the surplus-based algorithm (3.8) achieves average consensus if the parameter ϵ satisfies $\epsilon \in (0, (1 - (1/n))(2 - (1/n))$.

Proof. The symmetry of the undirected graph $\mathcal{G}$ results in the symmetry of its Laplacian matrix L. So all the eigenvalues μ_i of L are real, and satisfy $\mu_1 = 0$ and $(\forall i \in [2, n])\ \mu_i \in (0, 1/n]$ ($\mathcal{G}$ is connected). For $\mu_1 = 0$ we know from (3.19) that $\lambda_1 = 1$, and $\lambda_2 \in (-1, 1)$ since $0 < \epsilon < (1 - (1/n))(2 - (1/n)) < 2$. For $\mu_i \in (0, 1/n]$, $i \in [2, n]$, consider again the polynomial $p(\lambda)$ in (3.18). According to the Jury stability test for real-coefficient case [43], the zeros of $p(\lambda)$ are strictly inside the unit circle if and only if

$$1 + \alpha_0 + \alpha_1 > 0, \quad 1 + \alpha_0 - \alpha_1 > 0, \quad |\alpha_0| < 1.$$

Straightforward calculations show that these conditions hold provided $\epsilon \in (0, (1 - (1/n))(2 - (1/n))$. Hence, the matrix M has a simple eigenvalue $\lambda_1 = 1$ and all others $\lambda_2, \ldots, \lambda_{2n} \in (0, 1)$. Therefore, by Proposition 3.1 the surplus-based algorithm (3.8) achieves average consensus. ■

It is noted that for connected undirected graphs, the upper bound on ϵ ensuring average consensus grows as n increases. This characteristic is in agreement with that of the bounds for the more general class of balanced digraphs as we observed in Fig. 3.4.

3.5.2 Cyclic Digraphs

A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *cyclic* if $\mathcal{V} = \{1, \ldots, n\}$ and $\mathcal{E} = \{(1,2), (2,3), \ldots, (n-1, n), (n, 1)\}$. So a cyclic digraph is strongly connected.

Proposition 3.4. Suppose that the digraph $\mathcal{G}$ is cyclic. Then the surplus-based algorithm (3.8) achieves average consensus if the parameter ϵ satisfies

$$\epsilon \in \left(0, \frac{\sqrt{2}}{3+\sqrt{5}}\,(1 - |\lambda_3|)\right), \quad \text{with } \lambda_3 \text{ as in (3.10).} \tag{3.22}$$

Further, in this case $\lambda_3 = \sqrt{1 - (1/n) + (1/(2n^2)) + (1/n)(1 - 1/(2n))\cos 2\pi/n}$.

Before providing the proof, we state a perturbation result for diagonalizable matrices (e.g., [40, Section 6.3]). Recall that the matrix M in (3.8) can be written as $M = M_0 + \epsilon E$, with M_0 and E in (3.9). Throughout this subsection, write $\lambda_i(\epsilon)$ for the eigenvalues of M, and λ_i for those of M_0.

Lemma 3.5. Suppose that M_0 is diagonalizable; i.e., there exist a nonsingular matrix $V \in \mathbb{C}^{2n \times 2n}$ and a diagonal matrix $J = \mathrm{diag}(\lambda_1, \ldots, \lambda_{2n})$ such that $M_0 = VJV^{-1}$. If $\lambda(\epsilon)$ is an eigenvalue of M, then there is an eigenvalue λ_i of M_0, for some $i \in [1, 2n]$, such that $|\lambda(\epsilon) - \lambda_i| \leq ||V||_2 \, ||V^{-1}||_2 \, ||\epsilon E||_2$.

In other words, every eigenvalue of the perturbed matrix M lies in a circle centered at some eigenvalue of the unperturbed matrix M_0 of the radius $(||V||_2 \, ||V^{-1}||_2 \, ||\epsilon E||_2)$. We now present the proof of Proposition 3.4.

Proof of Proposition 3.4. Since the digraph $\mathcal{G}$ is cyclic, we derive its Laplacian matrix $L = \mathrm{circ}(1/(2n), 0, \ldots, 0, -1/(2n))$ – a *circulant matrix* [27] with the first row $[1/(2n)\ 0\ \cdots\ 0\ -1/(2n)] \in \mathbb{R}^{1 \times n}$. Let $\omega := \mathrm{e}^{2\pi\iota/n}$ with $\iota := \sqrt{-1}$. Then the eigenvalues μ_i of L are $\mu_i = (1/(2n))(1 - \omega^{i-1})$, $i = 1, \ldots, n$. Rewrite the equation (3.19) as $(\lambda(\epsilon) - 1)(\lambda(\epsilon) - 1 + \epsilon) + 3(\lambda(\epsilon) - 1)\mu_i + 2\mu_i^2 = 0$. Then for $\mu_1 = 0$, we have $\lambda_1(\epsilon) = 1$ and $\lambda_2(\epsilon) = 1 - \epsilon$, corresponding respectively to the eigenvalues λ_1, λ_2 of M_0. Evidently the upper bound in (3.22) is strictly smaller than 2; so $\lambda_2(\epsilon) \in (-1, 1)$.

We turn next to investigating the rest of the eigenvalues $\lambda_3(\epsilon), \ldots, \lambda_{2n}(\epsilon)$, for which we employ Lemma 3.5. Let F denote the $n \times n$ *Fourier matrix* given by

$$F := \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix}.$$

Note that F is unitary, i.e., $F^{-1} = F^*$ (the conjugate transpose of F). It is a fact that every circulant matrix can be (unitarily) diagonalized by F [27, Theorem 3.2.1]. Now let $V := \begin{bmatrix} F & 0 \\ F & F \end{bmatrix}$, and consider

$$\begin{aligned} V^{-1} M_0 V &= \begin{bmatrix} F^* & 0 \\ -F^* & F^* \end{bmatrix} \begin{bmatrix} I-L & 0 \\ L & S \end{bmatrix} \begin{bmatrix} F & 0 \\ F & F \end{bmatrix} \\ &= \begin{bmatrix} F^*(I-L)F & 0 \\ 0 & F^*SF \end{bmatrix}. \end{aligned}$$

The last equality is due to $S = I - 2L$. Hence M_0 is diagonalizable via V, and its spectrum is

$$\sigma(M_0) = \sigma(I-L) \cup \sigma(S) = \left\{ 1 - \frac{1}{2n}(1-\omega^{i-1}), 1 - \frac{1}{n}(1-\omega^{i-1}) : i = 1, \ldots, n \right\}.$$

Also, by a direct calculation we get $||V||_2 = ||V^{-1}||_2 = \sqrt{(3+\sqrt{5})/2}$ and $||E||_2 = \sqrt{2}$. It then follows from Lemma 3.5 that for every eigenvalue $\lambda_l(\epsilon)$ of M there is an eigenvalue $\lambda_{l'}$ of M_0, $l, l' \in [3, 2n]$, such that $|\lambda_l(\epsilon) - \lambda_{l'}| \leq ||V||_2 \, ||V^{-1}||_2 \, ||\epsilon E||_2 = \left((3+\sqrt{5})/2\right) \sqrt{2}\, \epsilon$. So the upper bound of ϵ in (3.22) guarantees $|\lambda_l(\epsilon) - \lambda_{l'}| < 1 - |\lambda_3|$; namely, the perturbed eigenvalues still lie within the unit circle. Summarizing the above we have $\lambda_1(\epsilon) = 1$ and $|\lambda_2(\epsilon)|, |\lambda_3(\epsilon)|, \ldots, |\lambda_{2n}(\epsilon)| < 1$; therefore, the surplus-based algorithm (3.8) achieves average consensus by Proposition 3.1. Further, one

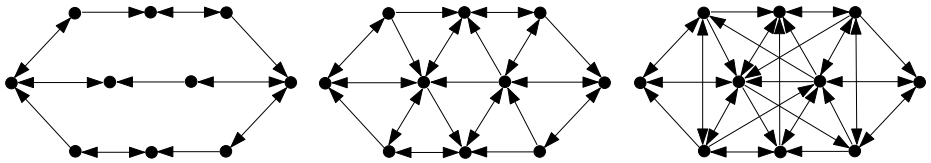


Figure 3.5: Three examples of strongly connected but unbalanced digraphs.

computes that

$$|\lambda_3| = \max_{i\in[2,n]} \left\{ \left|1 - \frac{1}{2n}(1-\omega^{i-1})\right|, \left|1 - \frac{1}{n}(1-\omega^{i-1})\right| \right\}$$
$$= \left|1 - \frac{1}{2n} + \frac{1}{2n}\omega\right| = \sqrt{1 - \frac{1}{n} + \frac{1}{2n^2} + \frac{1}{n}\left(1 - \frac{1}{2n}\right)\cos\frac{2\pi}{n}}.$$

■

Finally, in Fig. 3.4 we plot the upper bound on ϵ in (3.22) for the class of cyclic digraphs. We see that this bound decays as the number n of nodes increases, which contrasts with the bound characteristic of the more general class of balanced digraphs. This may indicate the conservativeness of our current approach based on perturbation theory. Nevertheless, since the perturbation result used here is specific only to diagonalizable matrices, the derived upper bound in (3.22) is less conservative than the general one in (3.11).

3.6 Numerical Examples

Let us illustrate, by simulation, that the surplus-based algorithm (3.8) indeed converges to the desired average value, as well as the corresponding convergence speed.

Example 3.3. Consider the three digraphs displayed in Fig. 3.5, with 10 nodes and respectively 17, 29, and 38 edges. Note that all the digraphs are strongly connected, and in the case of uniform weights they are unbalanced (indeed, no single node is balanced). We apply the surplus-based algorithm (3.8) by setting uniform weights $a = 1/(2\text{card}(\mathcal{E}))$ and $b = 1/\text{card}(\mathcal{E})$.

Table 3.1: Convergence factor λ_2^* with respect to different values of parameter ϵ.

	$\epsilon = 0.2$	$\epsilon = 0.7$	$\epsilon = 2.15$
$\mathcal{G}_a$	0.9963	0.9993	1.0003
$\mathcal{G}_b$	0.9951	0.9969	0.9985
$\mathcal{G}_c$	0.9883	0.9930	0.9966

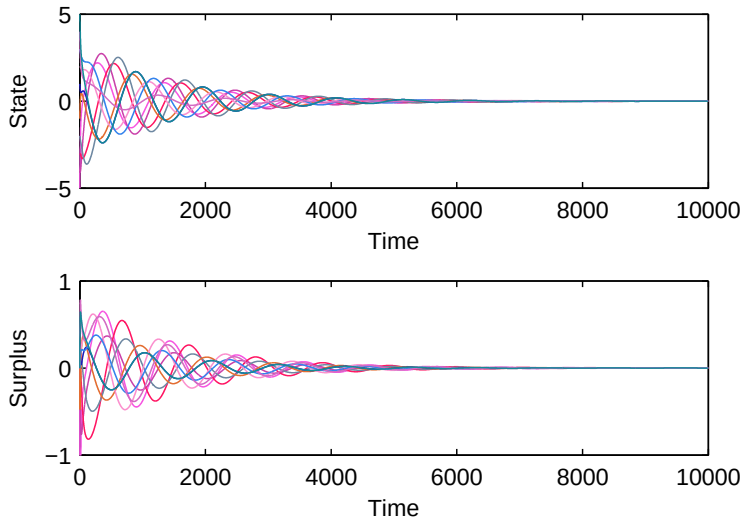


Figure 3.6: Convergence paths of states and surpluses: Obtained by applying surplus-based algorithm (3.8) with parameter $\epsilon = 0.7$ on digraph $\mathcal{G}_a$.

The convergence factor λ_2^* (see Remark 3.2) for three different values of the parameter ϵ are summarized in Table 3.1. We see that small ϵ ensures convergence of the algorithm (3.8) ($\lambda_2^* < 1$), whereas large ϵ can lead to instability. Moreover, in those converging cases the factor λ_2^* decreases as the number of edges increases from $\mathcal{G}_a$ to $\mathcal{G}_c$, which indicates faster convergence when there are more communication channels available for information exchange. We also see that the algorithm (3.8) is more robust on digraphs with more edges, in the sense that a larger range of values of ϵ is allowed.

For a random initial state $x(0)$ with the average $x_a = 0$ and the initial surplus $s(0) = 0$, we display in Fig. 3.6 the trajectories of both states and surpluses when

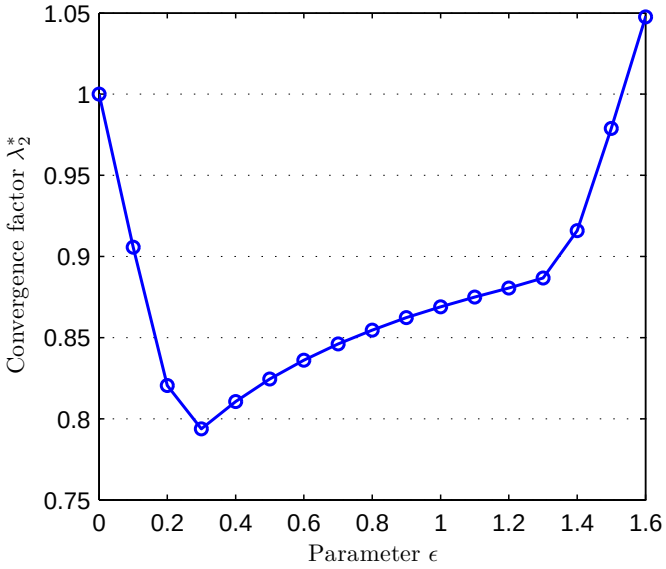


Figure 3.7: Convergence factor λ_2^* of surplus-based algorithm (3.8) with respect to parameter ϵ.

the surplus-based algorithm (3.8) is applied on digraph $\mathcal{G}_a$ with parameter $\epsilon = 0.7$. Observe that asymptotically, state averaging is achieved and surplus vanishes.

Example 3.4. We demonstrate the influence of ϵ on the speed of convergence, specifically the convergence factor λ_2^* (Remark 3.2). To reduce the effect of network topology in this demonstration, we employ a type of random digraphs where an edge between every pair of nodes can exist with probability 1/2, independent across the network and invariant over time; we take only those that are strongly connected.

For the surplus-based algorithm (3.8), consider random digraphs of 50 nodes and uniform weights $a = b = 1/50$. Fig. 3.7 displays the curve of convergence factor λ_2^* with respect to the parameter ϵ, each plotted point being the mean value of λ_2^* over 100 random digraphs. To account for the trend of this curve, first recall from the perturbation argument for Theorem 3.1 that the matrix M in (3.8) has two (maximum) eigenvalues 1 when $\epsilon = 0$, and small ϵ causes that one of them (denote its modulus by λ_{in}) moves into the unit circle. Meanwhile, some other eigenvalues of M inside the unit circle move outward; denote the maximum modulus among these by λ_{out}. In our simulation it is observed that when ϵ is small, $\lambda_2^* = \lambda_{\text{in}}$ ($> \lambda_{\text{out}}$)

and λ_{in} moves further inside as perturbation becomes larger; so λ_2^* decreases (faster convergence) as ϵ increases in the beginning. Since the eigenvalues move continuously, there exists some ϵ such that $\lambda_{\text{in}} = \lambda_{\text{out}}$, corresponding to the fastest convergence speed. After that, $\lambda_2^* = \lambda_{\text{out}}$ ($> \lambda_{\text{in}}$) and λ_{out} moves further outside as ϵ increases; hence λ_2^* increases and convergence becomes slower, and finally divergence occurs.

Chapter 4

Averaging on Dynamic Digraphs

4.1 Introduction

In the preceding chapter we studied multi-agent average consensus when the interaction structure among agents is static. This chapter progresses to explore a more challenging, and more interesting, scenario where this structure is time-varying. For this, we employ the *dynamic digraph* model introduced in Section 2.1. There can be many practical factors causing an interaction topology time-varying. There are unpredictable communication issues such as random packet loss, link failure, and node malfunction. There can also be deterministic, supervisory switchings among different modes of the network. In the sequel, we shall investigate average consensus under both randomized and deterministic dynamic topologies.

First, we consider the randomized case. Randomization-based techniques have been recently studied in the field of systems and control [75]. Here we adopt a simple but compelling "gossip" algorithm [12], which provides an asynchronous approach to treat average consensus. This is in contrast with the synchronized model studied in Chapter 3. The original gossip algorithm [12] assumes that at each time instant, exactly one agent wakes up, contacts only one of its neighbors selected at random, and then these two agents average out their states. The graph model that the original algorithm bases is undirected, and it turns out that average consensus is ensured as

long as the topology is *connected* [12]. Since then, the gossip approach has been widely employed [22, 45, 48, 81] in tackling average consensus on undirected graphs, with additional constraints on quantized information flow; see also [41] for related distributed computation problems in search engines.

We generalize the original gossip algorithm in [12] from undirected graphs to digraphs. Owing to unidirectional information flow, the generalization restricts that only one agent (rather than one pair) is allowed to update its state at each time. Consequently, the state sum of agents cannot be kept invariant during iterations. We propose again augmenting surplus variables, one for each agent, to locally record state updates, and for the new gossip algorithm, it is found that an arbitrary strongly connected digraph *in expectation* is necessary and sufficient to achieve average consensus *in mean-square* and (thus) *almost surely*. For the proof technique here, similar to the previous chapter we will analyze the spectral properties of the corresponding updating matrix, and establish convergence results again by matrix perturbation theory. We note that designing a gossip algorithm for digraphs was also reported in [29]; the algorithm achieves consensus at some value, however, not at the average in general, and the difference from the average is estimated. By contrast, our algorithm provably ensures, by design, converging to the average.

The second part of this chapter is devoted to digraphs that vary in some deterministic fashion. It will be seen that the *joint connectivity* notion (in Section 2.1) is essential. This setup has been extensively studied in the literature [52, 53, 56, 61, 65]; in particular, Olfati-Saber and Murray [61] justified that for the standard algorithm (3.5) to achieve average consensus under a dynamic digraph, a sufficient condition is that the digraph at every time instant is both strongly connected and balanced. Here supported by surplus variables, we are able to largely weaken the required graphical condition, and our result is that average consensus can be achieved if and only if the dynamic digraph is *jointly strongly connected*. Thus for one, we drop the "balanced" requirement completely; for the other, we need "strongly connected" only in a joint sense. As to the proof, the previous matrix spectrum analysis is no longer applicable

because the system under current setting is (deterministic) time-varying. Instead, we resort to the Lyapunov method, as in [53, 56]. This involves deriving an appropriate Lyapunov function for state evolution, and proving that the function decays monotonically.

The rest of this chapter is organized as follows. In Section 4.2 we formulate the time-varying average consensus problem. In Section 4.3 we present a novel gossip algorithm, and justify that it guarantees mean-square and almost sure state averaging when the digraph is general strongly connected in expectation. In Section 4.4 we present a new dynamic algorithm, and show that it ensures average consensus on jointly strongly connected digraphs. Moreover, in Section 4.5 we provide numerical examples for illustration.

4.2 Problem Formulation

Given a network of n (> 1) agents, we model their time-varying interconnection structure by a dynamic digraph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$, $k \in \mathbb{Z}_+$. The time-varying mechanism can be either random or deterministic. In the random case, we adopt "gossiping" of the following i.i.d. fashion: At each time k, exactly one edge $(j, i) \in \mathcal{E}(k)$ is activated at random, independently from all earlier instants and with a constant, strictly positive probability $p_{ij} \in (0, 1)$. Along this activated edge, node $j \in \mathcal{V}$ sends its state and surplus to i, while i receives the information and makes a corresponding update. According to the probability distribution, one may take expectation on $\mathcal{G}(k)$, denoted by $\overline{\mathcal{G}} = (\mathcal{V}, \overline{\mathcal{E}})$ where $\overline{\mathcal{E}}$ contains all edges whose probability to be activated is nonzero. Also, it is clear that $\sum_{(j,i)\in\overline{\mathcal{E}}} p_{ij} = 1$.

As before, for each agent i we denote by $x_i(k), s_i(k) \in \mathbb{R}$ its state and surplus at time k, respectively. Introduce the aggregate state $x(k) = [x_1(k) \cdots x_n(k)]^{\mathrm{T}} \in \mathbb{R}^n$ and the aggregate surplus $s(k) = [s_1(k) \cdots s_n(k)]^{\mathrm{T}} \in \mathbb{R}^n$. As surplus variables record state changes, we require $\mathbf{1}^{\mathrm{T}}(x(k) + s(k)) = \mathbf{1}^{\mathrm{T}}x(0)$ for all time k.

There are two problems to be solved in order. The first part will address randomly

varying topologies of the gossip type.

Definition 4.1. A network of agents achieves

(i) *mean-square average consensus* if for every initial condition $(x(0), 0)$, it holds that $E\left[||x(k) - x_a\mathbf{1}||_2^2\right] \to 0$ and $E\left[||s(k)||_2^2\right] \to 0$ as $k \to \infty$;

(ii) *almost sure average consensus* if for every initial condition $(x(0), 0)$, it holds that $(x(k), s(k)) \to (x_a\mathbf{1}, 0)$ as $k \to \infty$ with probability one.

As defined, the mean-square convergence is concerned with the second moments of the state and surplus evolution processes, whereas the almost sure convergence is with respect to the corresponding sample paths. It should be noted that in general there is no implication between these two convergence notions (e.g., [37, Section 7.2]).

Problem 4.1. Design a gossip algorithm such that the agents achieve mean-square and almost sure average consensus on general digraphs.

For this problem, we will propose in Section 4.3 a surplus-based gossip algorithm, under which we justify that both mean-square and almost sure average consensus can be achieved when the expected digraph $\overline{\mathcal{G}}$ is arbitrary strongly connected.

The second part is concerned with digraphs that vary in some deterministic way.

Problem 4.2. Design a dynamic algorithm such that the agents achieve average consensus (in the sense of Definition 3.1) on general digraphs.

To solve this problem, we will propose in Section 4.4 a surplus-based dynamic algorithm, under which we prove that average consensus may be achieved on jointly strongly connected digraphs.

4.3 Convergence Results on Randomized Dynamic Digraphs

In this section, we first present a surplus-based gossip algorithm, which extends those in [12, 45, 48] from undirected graphs to digraphs. Then we justify mean-square and almost sure convergence to average consensus for general topologies.

4.3.1 Gossip Algorithm

Consider again a network of n agents modeled by a dynamic digraph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ which varies in the gossip fashion. Suppose that at time k, exactly one edge $(j,i) \in \mathcal{E}(k)$ is activated (precisely $(j,i) \in \overline{\mathcal{E}}$ and $\mathcal{E}(k) = \{(j,i)\}$) at random with probability $p_{ij} \in (0,1)$. Along this edge, the state $x_j(k)$ and surplus $s_j(k)$ are transmitted from node j to i. The induced update is described as follows:

(1) For node i,

$$x_i(k+1) = x_i(k) + w_{ij}(x_j(k) - x_i(k)) + \epsilon w_{ij} s_i(k), \tag{4.1}$$

$$s_i(k+1) = s_i(k) + s_j(k) - (x_i(k+1) - x_i(k)) \tag{4.2}$$

where the updating weight $w_{ij} \in (0,1)$ and the parameter $\epsilon > 0$.

(2) For node j,

$$x_j(k+1) = x_j(k), \quad s_j(k+1) = 0.$$

(3) For other nodes $l \in \mathcal{V} - \{i,j\}$,

$$x_l(k+1) = x_l(k), \quad s_l(k+1) = s_l(k).$$

We emphasize that in the whole network at time k, only node j sends information, and only node i receives information and performs an update. Also note that the sender j always sets its surplus variable to be zero, meaning that all of its surpluses are transmitted to the receiver i.

Let A_{ji} be the adjacency matrix of the digraph $\mathcal{G}_{ji} = (\mathcal{V}, \{(j,i)\})$ given by $A_{ji} = w_{ij} f_i f_j^T$, where f_i, f_j are unit vectors of the standard basis of $\mathbb{R}^n$. Then the Laplacian matrix L_{ji} is given by $L_{ji} := D_{ji} - A_{ji}$, where $D_{ji} = w_{ij} f_i f_i^T$. Thus L_{ji} has zero row sums, and the matrix $I - L_{ji}$ is row stochastic. Also define $S_{ji} := I - (f_j - f_i) f_j^T$; it is clear that S_{ji} is column stochastic.

With these matrices defined, the iteration of states and surpluses when edge (j,i) is activated at time k can be written in the matrix form as

$$\begin{bmatrix} x(k+1) \\ s(k+1) \end{bmatrix} = M(k) \begin{bmatrix} x(k) \\ s(k) \end{bmatrix}, \text{ where } M(k) = M_{ji} := \begin{bmatrix} I - L_{ji} & \epsilon D_{ji} \\ L_{ji} & S_{ji} - \epsilon D_{ji} \end{bmatrix}. \tag{4.3}$$

We have several remarks regarding this algorithm. (i) The matrix $M(k)$ has negative entries due to the presence of the Laplacian matrix L_{ji} in the $(2,1)$-block. (ii) The column sums of $M(k)$ are equal to one, which implies that the quantity $x(k)+s(k)$ is constant for all k. (iii) By the assumption on the probability distribution of activating edges, the sequence $M(k)$, $k = 0, 1, \ldots$, is independent and identically distributed (i.i.d.). Henceforth we refer to (4.3) as the gossip algorithm, and establish its mean-square and almost sure convergence in the sequel.

Example 4.1. Consider again the network of four nodes in Fig. 3.1. We give two instances of the matrix $M(k)$ when edges $(3,2)$ and $(2,4)$ are activated respectively. For the associated updating weights, let $w_{23} = w_{42} = 1/2$.

$$M_{32} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & \epsilon/2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1/2 & -1/2 & 0 & 0 & 1-\epsilon/2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right],$$

$$M_{24} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & \epsilon/2 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1/2 & 0 & 1/2 & 0 & 1 & 0 & 1-\epsilon/2 \end{array}\right].$$

We see that in both cases $M(k)$ has negative entries, and every column sums up to one.

4.3.2 Convergence Results

We present the main result of this section.

Theorem 4.1. Using the gossip algorithm (4.3) with the parameter $\epsilon > 0$ sufficiently small, the agents achieve mean-square average consensus if and only if the expected digraph $\overline{\mathcal{G}}$ is strongly connected.

This result can be seen as an generalization of the one in [12] from undirected graphs to digraphs. The problem of achieving average consensus on arbitrary digraphs is, however, more difficult in that the state sum of agents need not be invariant at each iteration. The key in our extension is to augment surplus variables which keep track of individual state updates, thereby ensuring mean-square state averaging for general strongly connected digraphs (cf. Theorem 3.1). We note again we the algorithm parameter ϵ which specifies the amount of surpluses used in updating the states. It turns out that the parameter ϵ again should be small enough in order to ensure convergence.

Before proving Theorem 4.1, we need to establish two preliminary results. The first is a necessary and sufficient condition for mean-square average consensus characterized by the spectrum of the matrix $E[M(k) \otimes M(k)]$, where $\otimes$ stands for the Kronecker product. Since the matrices $M(k)$ are i.i.d., we denote $E[M(k) \otimes M(k)]$ by $E[M \otimes M]$. This result corresponds to Proposition 3.1 for the surplus-based algorithm (3.8) in Chapter 3.

Proposition 4.1. The gossip algorithm (4.3) achieves mean-square average consensus if and only if 1 is a simple eigenvalue of $E[M \otimes M]$, and all other eigenvalues have moduli smaller than one.

Proof. (Sufficiency) Define the consensus error $e(k)$, $k \geq 0$, as

$$e(k) := \begin{bmatrix} x(k) \\ s(k) \end{bmatrix} - \begin{bmatrix} x_a \mathbf{1} \\ 0 \end{bmatrix} \in \mathbb{R}^{2n}. \tag{4.4}$$

We must show that $E\left[e(k)^T e(k)\right] \to 0$ as $k \to \infty$. Since $\mathbf{1}^T(x(k)+s(k)) = \mathbf{1}^T x(0)$ for every $k \geq 0$, $e(k)$ is orthogonal to $[\mathbf{1}^T \ \mathbf{1}^T]^T$ (i.e., $[\mathbf{1}^T \ \mathbf{1}^T]e(k) = 0$). Also it is easy to

check $e(k+1) = M(k)e(k)$; thus $e(k+1)e(k+1)^T = M(k)e(k)e(k)^T M(k)^T$. Collect the entries of $e(k)e(k)^T$, drawn column wise, into a vector $\tilde{e}(k) \in \mathbb{R}^{4n^2}$. It then suffices to show that $E[\tilde{e}(k)] \to 0$ as $k \to \infty$.

Now it follows that $\tilde{e}(k+1) = (M(k) \otimes M(k))\,\tilde{e}(k)$ (cf. [12]). Hence $E[\tilde{e}(k+1)|\tilde{e}(k)] = E[M \otimes M]\,\tilde{e}(k)$, and condition repeatedly to obtain $E[\tilde{e}(k)] = E[M \otimes M]^k\,\tilde{e}(0)$. Note that every column of $E[M \otimes M]$ sums up to one, and

$$E[M \otimes M]\left(\begin{bmatrix}\mathbf{1}\\0\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\0\end{bmatrix}\right) = \begin{bmatrix}\mathbf{1}\\0\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1}\\0\end{bmatrix};$$

so 1 is an eigenvalue of $E[M \otimes M]$, with $[\mathbf{1}^T\ \mathbf{1}^T]^T \otimes [\mathbf{1}^T\ \mathbf{1}^T]^T$ and $[\mathbf{1}^T\ 0]^T \otimes [\mathbf{1}^T\ 0]^T$ as associated left and right eigenvectors, respectively. Write $E[M \otimes M]$ in Jordan canonical form as

$$E[M \otimes M] = VJV^{-1} = \begin{bmatrix} y_1 & \cdots & y_{4n^2}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & J'\end{bmatrix}\begin{bmatrix} z_1^T \\ \vdots \\ z_{4n^2}^T\end{bmatrix},$$

where J' contains the Jordan block matrices corresponding to those eigenvalues with moduli smaller than one. For the eigenvalue 1 choose $y_1 = [\mathbf{1}^T\ 0]^T \otimes [\mathbf{1}^T\ 0]^T$ and $z_1 = 1/n^2[\mathbf{1}^T\ \mathbf{1}^T]^T \otimes [\mathbf{1}^T\ \mathbf{1}^T]^T$; thus $z_1^T y_1 = 1$. Then the kth power of $E[M \otimes M]$ is

$$E[M \otimes M]^k = VJ^kV^{-1} = V\begin{bmatrix}1 & 0\\ 0 & (J')^k\end{bmatrix}V^{-1} \to y_1 z_1^T, \quad \text{as } k \to \infty.$$

Therefore we obtain

$$E[\tilde{e}(k)] \to y_1 z_1^T \tilde{e}(0) = y_1 \sum_{i=1}^{2n}\left(e_i(0)\sum_{j=1}^{2n} e_j(0)\right) = y_1 \sum_{i=1}^{2n} e_i(0)\cdot 0 = 0,$$

where the second equality is due to $e(k) \perp [\mathbf{1}^T\ \mathbf{1}^T]^T$.

(Necessity) Suppose $E\left[e(k)^T e(k)\right] \to 0$ as $k \to \infty$. Then $E[e_i(k)^2] \to 0$ for all i. It thus follows from the Cauchy-Schwartz inequality (e.g., [37]) that $E\left[|e_i(k)e_j(k)|\right]^2 \le E[e_i(k)^2]\,E[e_j(k)^2] \to 0$, for every $i, j \in [1, 2n]$. This implies $E[\tilde{e}(k)] \to 0$; so $\lim_{k\to\infty} E[M \otimes M]^k\,\tilde{e}(0) = 0$. Also, it is known [79] that $\lim_{k\to\infty} E[M \otimes M]^k$ ex-

ists if and only if there is a nonsingular V such that

$$E[M \otimes M] = VJV^{-1} = \begin{bmatrix} y_1 & \cdots & y_{4n^2} \end{bmatrix} \begin{bmatrix} I_\kappa & 0 \\ 0 & J' \end{bmatrix} \begin{bmatrix} z_1^T \\ \vdots \\ z_{4n^2}^T \end{bmatrix},$$

where $\kappa \in [1, 2n]$ and $\rho(J') < 1$. Hence $\lim_{k\to\infty} E[M \otimes M]^k \tilde{e}(0) = \left(\sum_{i=1}^{\kappa} y_i z_i^T\right) \tilde{e}(0) = 0$. Now suppose $\kappa > 1$. Choose as before $z_1 = 1/n^2[\mathbf{1}^T \ \mathbf{1}^T]^T \otimes [\mathbf{1}^T \ \mathbf{1}^T]^T$, and recall $z_1^T e(0) = 0$. We know from the structure of J that for every $j \in [2, \kappa]$, z_j is linearly independent of z_1, which indicates $z_j^T e(0) \neq 0$ and consequently $\left(\sum_{i=1}^{\kappa} y_i z_i^T\right) \tilde{e}(0) \neq 0$. Therefore $\kappa = 1$, i.e., the eigenvalue 1 of $E[M \otimes M]$ is simple and all others have moduli smaller than one. ■

The second preliminary is on the spectral properties of the following four matrices $E[(I - L) \otimes (I - L)]$, $E[(I - L) \otimes S]$, $E[S \otimes (I - L)]$, and $E[S \otimes S]$.

Lemma 4.1. Suppose that the expected digraph $\overline{\mathcal{G}} = (\mathcal{V}, \overline{\mathcal{E}})$ is strongly connected. Then each of the four matrices $E[(I - L) \otimes (I - L)]$, $E[(I - L) \otimes S]$, $E[S \otimes (I - L)]$, and $E[S \otimes S]$ has a simple eigenvalue 1 and all other eigenvalues with moduli smaller than one.

Proof. First observe that all the four matrices are nonnegative, for $I - L_{ji}$ and S_{ji} are for every $(j, i) \in \overline{\mathcal{E}}$. Then since $(I - L_{ji})\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^T S_{ji} = \mathbf{1}^T$ for every $(j, i) \in \overline{\mathcal{E}}$, a short calculation yields the following:

$$E[(I - L) \otimes (I - L)](\mathbf{1} \otimes \mathbf{1}) = (\mathbf{1} \otimes \mathbf{1});$$

$$E[(I - L) \otimes S](\mathbf{1} \otimes v_2) = (\mathbf{1} \otimes v_2);$$

$$(\mathbf{1}^T \otimes v_1^T)E[S \otimes (I - L)] = (\mathbf{1}^T \otimes v_1^T);$$

$$(\mathbf{1}^T \otimes \mathbf{1}^T)E[S \otimes S] = (\mathbf{1}^T \otimes \mathbf{1}^T).$$

Here v_1 is positive such that $v_1^T E[I - L] = v_1^T$ and $v_1^T \mathbf{1} = 1$, and v_2 is positive such that $E[S]\, v_2 = v_2$ and $\mathbf{1}^T v_2 = 1$. Thus each matrix has an eigenvalue 1, and the corresponding right or left eigenvector is positive. In what follows, it will be shown that all the four matrices are irreducible. Then the conclusion will follow from Lemmas 2.4 and 2.5.

We first prove that $E\left[(I-L)\otimes(I-L)\right]$ is irreducible, which is equivalent to that the digraph $\hat{\mathcal{G}}=(\hat{\mathcal{V}},\hat{\mathcal{E}})$ corresponding to this matrix is strongly connected, where $\hat{\mathcal{V}}:=\mathcal{V}\times\mathcal{V}=\{(i,i'):i,i'\in\mathcal{V}\}$. Arrange the nodes in $\hat{\mathcal{V}}$ so that $\hat{\mathcal{V}}=\mathcal{V}_1\cup\cdots\cup\mathcal{V}_n$, where $\mathcal{V}_p=\{(p,1),\ldots,(p,n)\}$ for every $p\in[1,n]$. Now since $E\left[(I-L)\otimes(I-L)\right]=\sum_{(j,i)\in\overline{\mathcal{E}}}p_{ij}(I-L_{ji})\otimes(I-L_{ji})$, the digraph $\hat{\mathcal{G}}$ is the union of the digraphs corresponding to $p_{ij}(I-L_{ji})\otimes(I-L_{ji})$. Note that each $p_{ij}(I-L_{ji})\otimes(I-L_{ji})$ gives rise to (i) an edge from (p,j) to (p,i) in $\mathcal{V}_p$ for every $p\in[1,n]$, and (ii) edges from some nodes in $\mathcal{V}_j$ to some nodes in $\mathcal{V}_i$. Owing to that $\overline{\mathcal{G}}$ is strongly connected, the union of the above edges yields, for every $i,j\in[1,n]$, (i) a directed path from (p,i) to (p,j) in $\mathcal{V}_p$ for every $p\in[1,n]$, and (ii) directed paths from some nodes in $\mathcal{V}_i$ to some nodes in $\mathcal{V}_j$. This implies that there is a directed path from (p,i) to (q,j) for every $p,q,i,j\in[1,n]$, i.e., $\hat{\mathcal{G}}$ is strongly connected, and hence $E\left[(I-L)\otimes(I-L)\right]$ is irreducible by Lemma 2.3.

By a similar argument, we derive that the digraphs corresponding to $E\left[(I-L)\otimes S\right]$, $E\left[S\otimes(I-L)\right]$, and $E\left[S\otimes S\right]$ are all strongly connected. Therefore they are also irreducible. ■

We are now ready to provide the proof of Theorem 4.1. The necessity argument is the same as Theorem 3.1. Below is the sufficiency part.

Sufficiency proof of Theorem 4.1. By Proposition 4.1 it suffices to show that the matrix $E\left[M\otimes M\right]$ has a simple eigenvalue 1, and all other eigenvalues with moduli smaller than one. Let $M_0(k):=\begin{bmatrix}I-L(k) & 0\\ L(k) & S(k)\end{bmatrix}$ and $F(k):=\begin{bmatrix}0 & D(k)\\ 0 & -D(k)\end{bmatrix}$; from (4.3) we have $M(k)=M_0(k)+\epsilon F(k)$. Then write

$$
\begin{aligned}
E\left[M\otimes M\right] &= E\left[(M_0+\epsilon F)\otimes(M_0+\epsilon F)\right]\\
&= E\left[M_0\otimes M_0\right]+\epsilon E\left[M_0\otimes F+F\otimes M_0+F\otimes\epsilon F\right]\\
&= E\left\{\begin{bmatrix}I-L & 0\\ L & S\end{bmatrix}\otimes\begin{bmatrix}I-L & 0\\ L & S\end{bmatrix}\right\}+\epsilon E\left\{\begin{bmatrix}I-L & 0\\ L & S\end{bmatrix}\otimes\begin{bmatrix}0 & D\\ 0 & -D\end{bmatrix}+\right.\\
&\qquad\left.\begin{bmatrix}0 & D\\ 0 & -D\end{bmatrix}\otimes\begin{bmatrix}I-L & 0\\ L & S\end{bmatrix}+\begin{bmatrix}0 & D\\ 0 & -D\end{bmatrix}\otimes\epsilon\begin{bmatrix}0 & D\\ 0 & -D\end{bmatrix}\right\}.
\end{aligned}
$$

Let $p\in[1,4n]$, and $p\mathbf{n}:=\{(p-1)n+1,\ldots,pn\}$. Consider the following permutation

of the columns of $E\,[M \otimes M]$:

$$\{\mathbf{n}, 3\mathbf{n}, \ldots, (2n-1)\mathbf{n};\ 2\mathbf{n}, 4\mathbf{n}, \ldots, 2n\mathbf{n};$$
$$(2n+1)\mathbf{n}, (2n+3)\mathbf{n}, \ldots, (4n-1)\mathbf{n};\ (2n+2)\mathbf{n}, (2n+4)\mathbf{n}, \ldots, 4n\mathbf{n}\}.$$

Denoting by P the corresponding permutation matrix (which is orthogonal), one derives that

$$\begin{aligned} P^T E\,[M \otimes M]\, P &= P^T E\,[M_0 \otimes M_0]\, P + \epsilon P^T E\,[M_0 \otimes F + F \otimes M_0 + F \otimes \epsilon F]\, P \\ &=: \hat{M}_0 + \epsilon \hat{F}, \end{aligned}$$

where

$$\hat{M}_0 := E \begin{bmatrix} (I-L)\otimes(I-L) & 0 & 0 & 0 \\ (I-L)\otimes L & (I-L)\otimes S & 0 & 0 \\ L\otimes(I-L) & 0 & S\otimes(I-L) & 0 \\ L\otimes L & L\otimes S & S\otimes L & S\otimes S \end{bmatrix},$$

$$\hat{F} := E \begin{bmatrix} 0 & (I-L)\otimes D & D\otimes(I-L) & D\otimes \epsilon D \\ 0 & -(I-L)\otimes D & D\otimes L & D\otimes(S-\epsilon D) \\ 0 & L\otimes D & -D\otimes(I-L) & (S-\epsilon D)\otimes D \\ 0 & -L\otimes D & -D\otimes L & D\otimes(\epsilon D - S) - S\otimes D \end{bmatrix}.$$

Based on the above similarity transformation, we henceforth analyze the spectral properties of the matrix $\hat{M}_0 + \epsilon\hat{F}$. For this, we resort again to a perturbation argument, which proceeds similarly to the one for Theorem 3.1. First, it follows from Lemma 4.1 that the eigenvalues of the matrix $\hat{M}_0$ satisfy

$$1 = \hat{\lambda}_1 = \hat{\lambda}_2 = \hat{\lambda}_3 = \hat{\lambda}_4 > |\hat{\lambda}_5| \geq \cdots \geq |\hat{\lambda}_{4n^2}|.$$

For the eigenvalue 1, the geometric multiplicity equals four; this can be shown by verifying $\operatorname{rank}(\hat{M}_0 - I) = 4n^2 - 4$. Thus the matrix $\hat{M}_0$ can be written in the following

Jordan canonical form:

$$\hat{M}_0 = VJV^{-1} = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & \cdots & y_{4n^2} \end{bmatrix} \left[\begin{array}{cccc|c} \hat{\lambda}_1 & 0 & 0 & 0 & \\ 0 & \hat{\lambda}_2 & 0 & 0 & 0 \\ 0 & 0 & \hat{\lambda}_3 & 0 & \\ 0 & 0 & 0 & \hat{\lambda}_4 & \\ \hline & & & & \\ & 0 & & & J' \\ & & & & \end{array}\right] \begin{bmatrix} z_1^T \\ z_2^T \\ z_3^T \\ z_4^T \\ \vdots \\ z_{4n^2}^T \end{bmatrix},$$

where $y_i,\ z_i \in \mathbb{C}^{4n^2}$, $i \in [1, 4n^2]$, are respectively the (generalized) right and left eigenvectors of $\hat{M}_0$; and J' contains the Jordan block matrices corresponding to $\hat{\lambda}_5, \ldots, \hat{\lambda}_{4n^2}$. Also write $J_0 := \mathrm{diag}(\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4)$. Now let v_1 be a positive vector such that $v_1^T E[I - L] = v_1^T$ and $v_1^T \mathbf{1} = 1$, and v_2 a positive vector such that $E[S]\, v_2 = v_2$ and $\mathbf{1}^T v_2 = 1$. Then choose

$$Y := \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \mathbf{1} \otimes \mathbf{1} \\ 0 & 0 & \mathbf{1} \otimes nv_2 & -\mathbf{1} \otimes nv_2 \\ 0 & nv_2 \otimes \mathbf{1} & 0 & -nv_2 \otimes \mathbf{1} \\ nv_2 \otimes nv_2 & -nv_2 \otimes nv_2 & -nv_2 \otimes nv_2 & nv_2 \otimes nv_2 \end{bmatrix},$$

$$Z := \begin{bmatrix} z_1^T \\ z_2^T \\ z_3^T \\ z_4^T \end{bmatrix} = \begin{bmatrix} \frac{1}{n}\mathbf{1}^T \otimes \frac{1}{n}\mathbf{1}^T & \frac{1}{n}\mathbf{1}^T \otimes \frac{1}{n}\mathbf{1}^T & \frac{1}{n}\mathbf{1}^T \otimes \frac{1}{n}\mathbf{1}^T & \frac{1}{n}\mathbf{1}^T \otimes \frac{1}{n}\mathbf{1}^T \\ \frac{1}{n}\mathbf{1}^T \otimes v_1^T & 0 & \frac{1}{n}\mathbf{1}^T \otimes v_1^T & 0 \\ v_1^T \otimes \frac{1}{n}\mathbf{1}^T & v_1^T \otimes \frac{1}{n}\mathbf{1}^T & 0 & 0 \\ v_1^T \otimes v_1^T & 0 & 0 & 0 \end{bmatrix}.$$

With this choice, it is readily checked that $ZY = I$.

Next, we qualify the changes of the four eigenvalues $\hat{\lambda}_1 = \hat{\lambda}_2 = \hat{\lambda}_3 = \hat{\lambda}_4 = 1$ of $\hat{M}_0$, under a small perturbation $\epsilon\hat{F}$. Since eigenvalues are continuous functions of matrix entries, for sufficiently small $\epsilon > 0$ there must exist exactly four eigenvalues $\hat{\lambda}_1(\epsilon)$, $\hat{\lambda}_2(\epsilon)$, $\hat{\lambda}_3(\epsilon)$, and $\hat{\lambda}_4(\epsilon)$ of $\hat{M}_0 + \epsilon\hat{F}$ corresponding respectively to $\hat{\lambda}_1$, $\hat{\lambda}_2$, $\hat{\lambda}_3$, and $\hat{\lambda}_4$ of $\hat{M}_0$. Write $J_0(\epsilon) := \mathrm{diag}(\hat{\lambda}_1(\epsilon), \hat{\lambda}_2(\epsilon), \hat{\lambda}_3(\epsilon), \hat{\lambda}_4(\epsilon))$, and denote the associated right

eigenvectors by

$$Y(\epsilon) = \begin{bmatrix} y_{11}(\epsilon) & y_{12}(\epsilon) & y_{13}(\epsilon) & y_{14}(\epsilon) \\ y_{21}(\epsilon) & y_{22}(\epsilon) & y_{23}(\epsilon) & y_{24}(\epsilon) \\ y_{31}(\epsilon) & y_{32}(\epsilon) & y_{33}(\epsilon) & y_{34}(\epsilon) \\ y_{41}(\epsilon) & y_{42}(\epsilon) & y_{43}(\epsilon) & y_{44}(\epsilon) \end{bmatrix}.$$

It can be verified that $J_0(\epsilon) \to J_0$ and $Y(\epsilon) \to Y$ as $\epsilon \to 0$ [69, Section 2.10]. Then

$$\begin{aligned}
& (\hat{M}_0 + \epsilon\hat{F})Y(\epsilon) = Y(\epsilon)J_0(\epsilon) \\
\Rightarrow\ & \hat{M}_0 Y(\epsilon) + \epsilon\hat{F}Y(\epsilon) = Y(\epsilon)J_0(\epsilon) \\
\Rightarrow\ & Z\hat{M}_0 Y(\epsilon) + \epsilon Z\hat{F}Y(\epsilon) = ZY(\epsilon)J_0(\epsilon) && \text{(left multiplying } Z \text{ on both sides)} \\
\Rightarrow\ & ZY(\epsilon)J_0(\epsilon) - J_0 ZY(\epsilon) = \epsilon Z\hat{F}Y(\epsilon). && \text{(by } Z\hat{M}_0 = J_0 Z)
\end{aligned}$$

Equating the four diagonal entries on both sides of the last equation above, one derives that

$$\begin{aligned}
&\left(\frac{1}{n}\mathbf{1}^T \otimes \frac{1}{n}\mathbf{1}^T\right)(y_{11}(\epsilon) + y_{21}(\epsilon) + y_{31}(\epsilon) + y_{41}(\epsilon))\frac{\hat{\lambda}_1(\epsilon) - \hat{\lambda}_1}{\epsilon} = 0, \\
&\left(v_1^T \otimes \frac{1}{n}\mathbf{1}^T\right)(y_{12}(\epsilon) + y_{32}(\epsilon))\frac{\hat{\lambda}_2(\epsilon) - \hat{\lambda}_2}{\epsilon} = \left(\frac{1}{n}\mathbf{1}^T \otimes v_1^T E\,[D]\right)(y_{22}(\epsilon) + y_{42}(\epsilon)), \\
&\left(\frac{1}{n}\mathbf{1}^T \otimes v_1^T\right)(y_{13}(\epsilon) + y_{23}(\epsilon))\frac{\hat{\lambda}_3(\epsilon) - \hat{\lambda}_3}{\epsilon} = \left(v_1^T E\,[D] \otimes \frac{1}{n}\mathbf{1}^T\right)(y_{33}(\epsilon) + y_{43}(\epsilon)), \\
&\left(v_1^T \otimes v_1^T\right) y_{14}(\epsilon)\frac{\hat{\lambda}_4(\epsilon) - \hat{\lambda}_4}{\epsilon} = \left(v_1^T \otimes v_1^T\right)\big(E\,[(I - L) \otimes D]\, y_{24}(\epsilon) \\
&\qquad\qquad + E\,[D \otimes (I - L)]\, y_{34}(\epsilon) + E\,[D \otimes \epsilon D]\, y_{44}(\epsilon)\big).
\end{aligned}$$

Let $\epsilon \to 0$. Then the left hand sides are

$$\begin{aligned}
&\left(\frac{1}{n}\mathbf{1}^T \otimes \frac{1}{n}\mathbf{1}^T\right)(y_{11}(\epsilon) + y_{21}(\epsilon) + y_{31}(\epsilon) + y_{41}(\epsilon))\frac{\hat{\lambda}_1(\epsilon) - \hat{\lambda}_1}{\epsilon} \\
&\qquad\qquad \to \left(\frac{1}{n}\mathbf{1}^T \otimes \frac{1}{n}\mathbf{1}^T\right)(nv_2 \otimes nv_2)\,\dot{\hat{\lambda}}_1(0) = \dot{\hat{\lambda}}_1(0), \\
&\left(v_1^T \otimes \frac{1}{n}\mathbf{1}^T\right)(y_{12}(\epsilon) + y_{32}(\epsilon))\frac{\hat{\lambda}_2(\epsilon) - \hat{\lambda}_2}{\epsilon} \to \left(v_1^T \otimes \frac{1}{n}\mathbf{1}^T\right)(nv_2 \otimes \mathbf{1})\,\dot{\hat{\lambda}}_2(0) = \dot{\hat{\lambda}}_2(0), \\
&\left(\frac{1}{n}\mathbf{1}^T \otimes v_1^T\right)(y_{13}(\epsilon) + y_{23}(\epsilon))\frac{\hat{\lambda}_3(\epsilon) - \hat{\lambda}_3}{\epsilon} \to \left(\frac{1}{n}\mathbf{1}^T \otimes v_1^T\right)(\mathbf{1} \otimes nv_2)\,\dot{\hat{\lambda}}_3(0) = \dot{\hat{\lambda}}_3(0), \\
&\left(v_1^T \otimes v_1^T\right) y_{14}(\epsilon)\frac{\hat{\lambda}_4(\epsilon) - \hat{\lambda}_4}{\epsilon} \to \left(v_1^T \otimes v_1^T\right)(\mathbf{1} \otimes \mathbf{1})\,\dot{\hat{\lambda}}_4(0) = \dot{\hat{\lambda}}_4(0),
\end{aligned}$$

where $\dot{\hat{\lambda}}_i(0) := \lim_{\epsilon\to 0}(\hat{\lambda}_i(\epsilon) - \hat{\lambda}_i)/\epsilon$, $i = 1, 2, 3, 4$; and the right hand sides are

$$
\begin{aligned}
&0,\\
&\left(\frac{1}{n}\mathbf{1}^T \otimes v_1^T E\left[D\right]\right)\left(y_{22}(\epsilon) + y_{42}(\epsilon)\right) \to \left(\frac{1}{n}\mathbf{1}^T \otimes v_1^T E\left[D\right]\right)\left(-nv_2 \otimes nv_2\right) = -nv_1^T E\left[D\right] v_2 < 0,\\
&\left(v_1^T E\left[D\right] \otimes \frac{1}{n}\mathbf{1}^T\right)\left(y_{33}(\epsilon) + y_{43}(\epsilon)\right) \to \left(v_1^T E\left[D\right] \otimes \frac{1}{n}\mathbf{1}^T\right)\left(-nv_2 \otimes nv_2\right) = -nv_1^T E\left[D\right] v_2 < 0,\\
&\left(v_1^T \otimes v_1^T\right)\left(E\left[(I-L)\otimes D\right] y_{24}(\epsilon) + E\left[D \otimes (I-L)\right] y_{34}(\epsilon) + E\left[D \otimes \epsilon D\right] y_{44}(\epsilon)\right)\\
&\qquad \to \left(v_1^T \otimes v_1^T\right)\left(E\left[(I-L)\otimes D\right]\left(-\mathbf{1} \otimes nv_2\right) + E\left[D \otimes (I-L)\right]\left(-nv_2 \otimes \mathbf{1}\right)\right)\\
&\qquad = \left(v_1^T \otimes v_1^T\right)\left(-\mathbf{1} \otimes E\left[D\right] nv_2 - E\left[D\right] nv_2 \otimes \mathbf{1}\right) = -2nv_1^T E\left[D\right] v_2 < 0.
\end{aligned}
$$

So one obtains that $\dot{\hat{\lambda}}_1(0) = 0$ and $\dot{\hat{\lambda}}_2(0)$, $\dot{\hat{\lambda}}_3(0)$, $\dot{\hat{\lambda}}_4(0) < 0$. This implies that when ϵ is small, $\hat{\lambda}_1(\epsilon)$ stays put, and $\hat{\lambda}_2(\epsilon)$, $\hat{\lambda}_3(\epsilon)$, $\hat{\lambda}_4(\epsilon)$ move to the left along the real axis. Hence by continuity, there exists a positive δ_1 such that $\lambda_1(\delta_1) = 1$ and $\lambda_2(\delta_1)$, $\lambda_3(\delta_1)$, $\lambda_4(\delta_1) < 1$. On the other hand, by the eigenvalue continuity there exists a positive δ_2 such that $|\lambda_i(\delta_2)| < 1$ for all $i \in [5, 4n^2]$. Therefore for any sufficiently small $\epsilon \in (0, \min\{\delta_1, \delta_2\})$, the matrix $\hat{M}_0 + \epsilon\hat{F}$ has a simple eigenvalue 1 and all other eigenvalues with moduli smaller than one. ■

Remark 4.1. (Convergence Speed) Assuming that the gossip algorithm (4.3) converges to the initial average in mean square, the speed of its convergence is determined by the second largest (in modulus) eigenvalue of the matrix $E\left[M \otimes M\right]$. We denote this particular eigenvalue by $\lambda_2^{(g)}$, and refer to it as the *convergence factor* of algorithm (4.3). Note that $\lambda_2^{(g)} < 1$ is equivalent to mean-square average consensus (by Proposition 4.1), and $\lambda_2^{(g)}$ depends not only on the graph topology $\overline{\mathcal{G}}$ but also on the algorithm parameter ϵ.

Finally, we treat almost sure average consensus. Note that the gossip algorithm (4.3) can be viewed as a *jump linear system*, with i.i.d. system matrices $M(k)$, $k \in \mathbb{Z}_+$. For such systems, it is known (e.g., [26, Corollary 3.46]) that almost sure convergence can be implied from mean-square convergence. Therefore the convergence result on almost sure state averaging is an immediate consequence of Theorem 4.1.

Corollary 4.1. Using the gossip algorithm (4.3) with the parameter $\epsilon > 0$ sufficiently small, the agents achieve almost sure average consensus if and only if the expected digraph $\overline{\mathcal{G}}$ is strongly connected.

4.4 Convergence Results on Deterministic Dynamic Digraphs

We turn now to dynamic digraphs $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ that vary in some deterministic way. First, we present a surplus-based dynamic algorithm, which is an extension of (3.8). Then we prove that it achieves average consensus when $\mathcal{G}(k)$ is jointly strongly connected, thereby generalizing the result of [61].

4.4.1 Dynamic Algorithm

In the algorithm, there are three operations that every node i performs at time $k \in \mathbb{Z}_+$. First (sending stage), node i sends its state information $x_i(k)$ and weighted surplus $b_{ih}(k)s_i(k)$ to each out-neighbor $h \in \mathcal{N}_i^-(k)$. Second (receiving stage), node i receives state information $x_j(k)$ and weighted surplus $b_{ji}(k)s_j(k)$ from each in-neighbor $j \in \mathcal{N}_i^+(k)$. Third (updating stage), node i updates its own state $x_i(k)$ and surplus $s_i(k)$ as follows:

$$x_i(k+1) = x_i(k) + c_i(k) \sum_{j \in \mathcal{N}_i^+(k)} a_{ij}(k)(x_j(k) - x_i(k)) + \epsilon_i(k)s_i(k), \tag{4.5}$$

$$s_i(k+1) = \Big((1 - \sum_{h \in \mathcal{N}_i^-(k)} b_{ih}(k))s_i(k) + \sum_{j \in \mathcal{N}_i^+(k)} b_{ji}(k)s_j(k)\Big) - \Big(x_i(k+1) - x_i(k)\Big). \tag{4.6}$$

The parameters used in Eqs. (4.5) and (4.6) are required to satisfy the following items, for every $i, j, h \in \mathcal{V}$ and every $k \in \mathbb{Z}_+$:

(P1) amount of surplus $\epsilon_i(k) \in (0, 1)$ used for state update.

(P2) updating weights $a_{ij}(k) \in (0, 1)$ if $j \in \mathcal{N}_i^+(k)$, $a_{ij}(k) = 0$ otherwise, and $\sum_{j \in \mathcal{N}_i^+(k)} a_{ij}(k) < 1$.

(P3) sending weights $b_{ih}(k) \in (0,1)$ if $h \in \mathcal{N}_i^-(k)$, $b_{ih}(k) = 0$ otherwise, and $\sum_{h\in\mathcal{N}_i^-(k)} b_{ih}(k) < 1-\epsilon_i(k)$. The last inequality says that the amount of surplus sent to out-neighbors should be strictly less than the total minus that used for state update.

(P4) switching parameters $c_i(k) = 1$ if $\sum_{j\in\mathcal{N}_i^+(k)} a_{ij}(k)(x_j(k) - x_i(k)) \leq 0$, and $c_i(k) = 0$ otherwise. The switching rule means that whenever a node determines to make a *positive* state update based on the information from in-neighbors, it may use only its surplus for that update.

(P1)-(P4) will enable desired properties of the proposed algorithm. In particular, (P3) and (P4) will be used to establish that all the surpluses are *nonnegative*; see Lemma 4.2 below. Also, one verifies that $\mathbf{1}^T(x(k)+s(k)) = \mathbf{1}^T x(0)$ for every $k \in \mathbb{Z}_+$.

Define the *adjacency matrix* $A(k)$ of the digraph $\mathcal{G}(k)$ by $A(k) := [c_i(k)a_{ij}(k)]$. Then the *Laplacian matrix* $L(k)$ is defined as $L(k) := D(k) - A(k)$, where $D(k) = \mathrm{diag}(d_1(k),\dots,d_n(k))$ with $d_i(k) = \sum_{j=1}^n c_i(k)a_{ij}(k)$. It is easy to see that $L(k)$ has nonnegative diagonal entries, nonpositive off-diagonal entries, and zero row sums. Consequently the matrix $I-L(k)$ is nonnegative (by $\sum_{j\in\mathcal{N}_i^+(k)} a_{ij}(k) < 1$ in (P2)), and every row sums up to one; namely $I-L(k)$ is *row stochastic*. Also, let $B(k) := [b_{ih}(k)]^T$ (note that the transpose in the notation is needed because $h \in \mathcal{N}_i^-(k)$ for $b_{ih}(k)$). Define the matrix $S(k) := (I - \tilde{D}(k)) + B(k)$, where $\tilde{D}(k) = \mathrm{diag}(\tilde{d}_1(k),\dots,\tilde{d}_n(k))$ with $\tilde{d}_i(k) = \sum_{h=1}^n b_{ih}(k)$. Then $S(k)$ is nonnegative (by $\sum_{h\in\mathcal{N}_i^-(k)} b_{ih} < 1-\epsilon_i(k)$ in (P3) and $\epsilon_i(k) \in (0,1)$ in (P1)), and every column sums up to one; that is, $S(k)$ is *column stochastic*. As can be observed from Eq. (4.6), $S(k)$ captures the part of update induced by sending and receiving surplus. Finally, let $E(k) := \mathrm{diag}(\epsilon_1(k),\dots,\epsilon_n(k))$.

With the above matrices defined, the iteration of states Eq. (4.5) and surpluses Eq. (4.6) can be written in a matrix form as

$$\begin{bmatrix} x(k+1) \\ s(k+1) \end{bmatrix} = M(k) \begin{bmatrix} x(k) \\ s(k) \end{bmatrix}, \quad \text{where } M(k) := \begin{bmatrix} I - L(k) & E(k) \\ L(k) & S(k) - E(k) \end{bmatrix} \in \mathbb{R}^{2n\times 2n}. \tag{4.7}$$

Notice that the column sums of $M(k)$ are equal to one (here $S(k)$ being column stochastic is crucial), which implies that the quantity $x(k) + s(k)$ is a constant for every $k \in \mathbb{Z}_+$. Henceforth we refer to (4.7) as the dynamic algorithm, and show its convergence in the sequel. Some other useful implications derived from this algorithm (4.7) are collected in the following lemma. Define the minimum and maximum states, $\underline{m}(x)$ and $\overline{m}(x)$, by

$$\underline{m}(x) := \min_{i\in\mathcal{V}} x_i, \quad \overline{m}(x) := \max_{i\in\mathcal{V}} x_i. \tag{4.8}$$

Lemma 4.2. In the dynamic algorithm (4.7), the following properties hold:

(i) Surplus $s_i(k) \geq 0$, for every $i \in \mathcal{V}$ and every $k \in \mathbb{Z}_+$.

(ii) Minimum state $\underline{m}(x)$ is non-decreasing, i.e., $\underline{m}(x(k_1)) \leq \underline{m}(x(k_2))$ if $k_1 \leq k_2$.

(iii) Minimum state $\underline{m}(x(k)) \leq x_a$ for every $k \in \mathbb{Z}_+$; and $\underline{m}(x(k)) = x_a$ implies $(\forall i \in \mathcal{V})\ x_i(k) = x_a$ and $s_i(k) = 0$, i.e., average consensus.

Proof. **(i)** We show this property by induction on the time index k. For the base case $k = 0$, we have $s_i(0) = 0$ for all i. Now suppose that $s_i(k) \geq 0$, $k > 0$, for all i. According to Eqs. (4.5) and (4.6) we derive

$$\begin{aligned} s_i(k+1) = &\Big((1 - \sum_{h\in\mathcal{N}_i^-(k)} b_{ih}(k))s_i(k) + \sum_{j\in\mathcal{N}_i^+(k)} b_{ji}(k)s_j(k)\Big) \\ &\qquad - \Big(\sum_{j\in\mathcal{N}_i^+(k)} c_i(k)a_{ij}(k)(x_j(k) - x_i(k)) + \epsilon_i(k)s_i(k)\Big) \\ = &\Big(1 - \sum_{h\in\mathcal{N}_i^-(k)} b_{ih}(k) - \epsilon_i(k)\Big)s_i(k) + \sum_{j\in\mathcal{N}_i^+(k)} b_{ji}(k)s_j(k) \\ &\qquad - \sum_{j\in\mathcal{N}_i^+(k)} c_i(k)a_{ij}(k)(x_j(k) - x_i(k)). \end{aligned}$$

It then follows from **(P3)**, **(P4)**, and the induction hypothesis that $s_i(k+1) \geq 0$ for all i. This completes the induction.

(ii) Let $k \in \mathbb{Z}_+$ be arbitrary, and $i \in \mathcal{V}$ be such that $x_i(k) = \underline{m}(x(k))$. Then it must hold that $\sum_{j\in\mathcal{N}_i^+(k)} a_{ij}(k)(x_j(k) - x_i(k)) \geq 0$. Thus $c_i(k) = 0$ by **(P4)**. Therefore

$x_i(k+1) = x_i(k) + \epsilon_i(k)s_i(k) \geq x_i(k)$. This proves that the minimum state cannot decrease.

(iii) Suppose on the contrary that $\underline{m}(x(k)) > x_a$ for some k. This implies that $\mathbf{1}^T x(k) + \mathbf{1}^T s(k) > nx_a + \mathbf{1}^T s(k)$. But since $\mathbf{1}^T x(k) + \mathbf{1}^T s(k) = \mathbf{1}^T x(0) = nx_a$, one obtains $\mathbf{1}^T s(k) < 0$, a contradiction to the property **(i)**. Hence we conclude that $\underline{m}(x(k)) \leq x_a$ for all k. And when $\underline{m}(x(k)) = x_a$, we must also have $\overline{m}(x(k)) = x_a$ owing again to **(i)**. Therefore $x_i(k) = x_a$ and $s_i(k) = 0$ for all i. ■

4.4.2 Convergence Results

Recall from Chapter 2 that $\mathcal{G}(k)$ is *jointly strongly connected* if there exists a finite $\mathcal{K} \in \mathbb{Z}_+$ such that for every $k_0 \in \mathbb{Z}_+$, the union digraph $\mathcal{G}([k_0, k_0 + \mathcal{K}])$ is strongly connected. Our main result of this section is this:

Theorem 4.2. Using the dynamic algorithm (4.7), a network of agents achieves average consensus if and only if the dynamic digraph $\mathcal{G}(k)$ is jointly strongly connected.

Comparing our derived graphical condition with the one in [61], we drop the balanced requirement at every moment on one hand, and need strongly connected property only in a joint sense on the other. We also point out that the time-varying parameter ϵ is not required to be sufficiently small, as contrasted with Theorems 3.1 and 4.1.

We proceed to the proof of Theorem 4.2, for which we rely on the following Lyapunov result (cf. [56, Theorem 4 and Remark 5]). For any given x_a, let

$$\mathcal{X}(x_a) := \{(x, s) : \mathbf{1}^T(x + s)/n = x_a,\ s \geq 0\}. \tag{4.9}$$

Lemma 4.3. Consider the dynamic algorithm (4.7). Suppose that continuous functions $V : \mathcal{X}(x_a) \to \mathbb{R}_+$ and $\delta : \mathcal{X}(x_a) \to \mathbb{R}_+$ satisfy the following conditions:
(i) V is bounded on bounded subsets of $\mathcal{X}(x_a)$, and positive definite with respect to the average consensus point $(x_a\mathbf{1}, 0)$ (i.e., $V(x_a\mathbf{1}, 0) = 0$ and $V(x, s) > 0$ if $(x, s) \neq (x_a\mathbf{1}, 0)$);

(ii) δ is also positive definite with respect to the average consensus point $(x_a\mathbf{1}, 0)$ (i.e., $\delta(x_a\mathbf{1}, 0) = 0$ and $\delta(x, s) > 0$ if $(x, s) \neq (x_a\mathbf{1}, 0)$);
(iii) there exists a finite $\kappa \in \mathbb{Z}_+$ such that for every $(x(k), s(k)) \in \mathcal{X}(x_a)$,

$$V(x(k+\kappa), s(k+\kappa)) - V(x(k), s(k)) \leq -\delta(x(k), s(k)), \quad k \in \mathbb{Z}_+.$$

Then for every initial condition $(x(0), 0) \in \mathcal{X}(x_a)$, it holds that $(x(k), s(k)) \to (x_a\mathbf{1}, 0)$ as $k \to \infty$.

For the proof of Lemma 4.3, refer to [56]. In the sequel, we will construct two functions that satisfy the conditions in Lemma 4.3. First consider the function $V(x, s)$, $(x, s) \in \mathcal{X}(x_a)$ in (4.9), defined by

$$V(x, s) := \frac{\mathbf{1}^T(x + s)}{n} - \underline{m}(x). \tag{4.10}$$

Clearly V depends continuously on (x, s). Since $\mathbf{1}^T(x(k) + s(k))/n = \mathbf{1}^T x(0)/n = x_a$ for all k, we obtain by (ii), (iii) of Lemma 4.2 that $V(x, s)$ is non-increasing (i.e., $V(x(k_1), s(k_1)) \geq V(x(k_2), s(k_2))$ if $k_1 \leq k_2$) thus bounded, and positive definite with respect to the average consensus point $(x_a\mathbf{1}, 0)$ (i.e., $V(x_a\mathbf{1}, 0) = 0$ and $V(x, s) > 0$ if $(x, s) \neq (x_a\mathbf{1}, 0)$.

Second, for a given $\kappa \in \mathbb{Z}_+$ define the function $\delta_\kappa(x, s)$, $(x, s) \in \mathcal{X}(x_a)$ in (4.9), by

$$\delta_\kappa(x(k), s(k)) := \inf_{(x(k),s(k)),\ldots,(x(k+\kappa),s(k+\kappa))} V(x(k), s(k)) - V(x(k+\kappa), s(k+\kappa)), \tag{4.11}$$

where the infimum is taken over all sequences of the solutions of (4.7) over the time interval $[k, k + \kappa]$.

Lemma 4.4. The function $\delta_\kappa(x, s)$ defined in (4.11) is continuous.

Proof. For given k, $\kappa \in \mathbb{Z}_+$, consider an arbitrary sequence $(x(k), s(k)), (x(k+1), s(k+1)), \ldots, (x(k+\kappa), s(k+\kappa))$ satisfying

$$\begin{bmatrix} x(k+1) \\ s(k+1) \end{bmatrix} = M(k) \begin{bmatrix} x(k) \\ s(k) \end{bmatrix}, \ \ldots, \ \begin{bmatrix} x(k+\kappa) \\ s(k+\kappa) \end{bmatrix} = M(k+\kappa-1) \begin{bmatrix} x(k+\kappa-1) \\ s(k+\kappa-1) \end{bmatrix}.$$

First, we show that $M(l)$, $l = k, \ldots, k+\kappa$, is a continuous function of (x, s). According to Eqs. (4.5) and (4.6), it suffices to show that every function $x_i : \mathbb{R}^{2n} \to \mathbb{R}$, $i \in \mathcal{V}$, is continuous at an arbitrary point (x^*, s^*) satisfying $\sum_{j\in\mathcal{N}_i^+} a_{ij}(x_j^* - x_i^*) = 0$ (the boundary of switching). Let (x, s) be any point sufficiently close to (x^*, s^*) and satisfy $\sum_{j\in\mathcal{N}_i^+} a_{ij}(x_j - x_i) > 0$ (i.e., $c_i = 0$). Then

$$\lim_{(x,s)\to(x^*,s^*)} x_i(x, s) = \lim_{(x,s)\to(x^*,s^*)} x_i + \epsilon_i s_i = x_i^* + \epsilon_i s_i^*.$$

On the other hand, let (x, s) be any point again sufficiently close to (x^*, s^*) but satisfy $\sum_{j\in\mathcal{N}_i^+} a_{ij}(x_j - x_i) < 0$ (i.e., $c_i = 1$). Then

$$\begin{aligned}\lim_{(x,s)\to(x^*,s^*)} x_i(x, s) &= \lim_{(x,s)\to(x^*,s^*)} x_i + \sum_{j\in\mathcal{N}_i^+} a_{ij}(x_j - x_i) + \epsilon_i s_i \\ &= x_i^* + \sum_{j\in\mathcal{N}_i^+} a_{ij}(x_j^* - x_i^*) + \epsilon_i s_i^* = x_i^* + \epsilon_i s_i^*.\end{aligned}$$

The above derivations prove that $\lim_{(x,s)\to(x^*,s^*)} x_i(x, s)$ exists, and is equal to $x_i(x^*, s^*)$. Hence the function $x_i(\cdot, \cdot)$ is continuous.

Second, the sequence $(x(k), s(k)), (x(k + 1), s(k + 1)), \ldots, (x(k + \kappa), s(k + \kappa))$ depends continuously on $(x(k), s(k))$. This is because the function $M(l)$, $l = k, \ldots, k+\kappa$, is continuous, and there is only a finite number of possible switching sequences of $\kappa-1$ digraphs. Thus, it follows from (4.10) that the expression $V(x(k), s(k))-V(x(k+\kappa), s(k+\kappa))$ depends continuously on $(x(k), s(k))$. Finally, by the definition of (4.11), we conclude that the function $\delta_\kappa(x, s)$ is continuous. ■

Now from (4.11), one may easily see that the function $\delta_\kappa(x, s) = 0$ if $V(x, s) = 0$; so $\delta_\kappa(x_a\mathbf{1}, 0) = 0$. The following result will be vital, which asserts that there always exists a finite $\kappa \in \mathbb{Z}_+$ such that the function $\delta_\kappa(x, s)$ is positive definite with respect to the average consensus point $(x_a\mathbf{1}, 0)$, provided that the network topology is jointly strongly connected.

Lemma 4.5. Suppose that the dynamic digraph $\mathcal{G}(k)$ is jointly strongly connected. There exists a finite $\kappa \in \mathbb{Z}_+$ such that if $V(x, s)$ is strictly positive, then $\delta_\kappa(x, s)$ is also strictly positive.

Lemma 4.5 indicates that the function $\delta_\kappa(x,s) > 0$ for $(x,s) \neq (x_a\mathbf{1}, 0)$. We postpone the proof of Lemma 4.5, and provide now the proof of Theorem 4.2.

Proof of Theorem 4.2. (Sufficiency) Suppose that $\mathcal{G}(k)$ is jointly strongly connected. Then it follows from Lemmas 4.4 and 4.5 that the function δ_κ defined in (4.11) and the function V defined in (4.10) satisfy the conditions in Lemma 4.3. Therefore average consensus is achieved.

(Necessity) Suppose that $\mathcal{G}(k)$ is not jointly strongly connected. Namely for every $\mathcal{K}$ there exists k_0 such that the union digraph $\mathcal{G}([k_0, k_0+\mathcal{K}])$ is not strongly connected. Thus during this interval $[k_0, k_0+\mathcal{K}]$, there are some nodes not globally reachable; denote the number by $r \in [1,n]$. First, if $r = n$, then $\mathcal{G}([k_0, k_0+\mathcal{K}])$ has at least two distinct closed strong components, say $\mathcal{V}_1$ and $\mathcal{V}_2$ (by Lemma 2.1). Consider a state-surplus pair $(x(k_0), 0)$ such that the nodes in $\mathcal{V}_1$ have states a, those in $\mathcal{V}_2$ have states b, and $a \neq b$. Then no update will occur, and therefore average consensus cannot be achieved. Second, when $r < n$, we denote by $\mathcal{V}_g$ the set of all globally reachable nodes. Then $\mathcal{V}_g$ is the unique closed strong component in $\mathcal{G}([k_0, k_0+\mathcal{K}])$ (again by Lemma 2.1). Consider a state-surplus pair $(x(k_0), 0)$ such that the nodes in $\mathcal{V}_g$ have states a, those in $\mathcal{V} - \mathcal{V}_g$ have states b, and $a \neq b$. Then no update will occur for the states in $\mathcal{V}_g$, and therefore average consensus cannot be achieved. ■

Finally we prove Lemma 4.5.

Proof of Lemma 4.5. Fix an arbitrary time $k_0 \in \mathbb{Z}_+$, and denote by $\mu := \underline{m}(x(k_0))$ the minimum state at this time. Assume $\mu < x_a$ (i.e., average consensus is not yet reached); thus $V(x(k_0))$ is strictly positive. It must be shown that $\delta_\kappa(x(k_0))$ is also strictly positive, for some finite $\kappa \in \mathbb{Z}_+$. Define a set $a(k)$, $k \geq k_0$, by

$$a(k) := \{i \in \mathcal{V} : x_i(k) = \mu\}. \tag{4.12}$$

Then $a(k)$ is the set of nodes whose states are equal to μ at time $k \geq k_0$. First, owing to the state update Eq. (4.5), together with (P2) and (P4), any $x_j(k) > \mu$ cannot decrease to μ in finite time. This implies $a(k+1) \subseteq a(k)$, $k \geq k_0$. Next, we will establish that when the topology $\mathcal{G}(k)$ is jointly strongly connected of period $\mathcal{K}$, there exists $\tilde{\kappa}(\mathcal{K}) \in \mathbb{Z}_+$ such that $a(k + \tilde{\kappa}(\mathcal{K}))$ is strictly contained in $a(k)$, $k \geq k_0$

(that is, $a(k+\tilde{\kappa}(\mathcal{K}))$ has strictly less nodes than $a(k)$). For this, the following claim is crucial, which asserts that positive surpluses can diffuse across the network under jointly strongly connected topology.

Claim. Suppose that at time $k \geq k_0$ there are $r \in [1, n-1]$ surpluses strictly positive, say $s_1(k), \ldots, s_r(k) > 0$, and $s_{r+1}(k) = \cdots = s_n(k) = 0$. Then $s_i(k + (n-r)\mathcal{K}) > 0$, for every $i \in \mathcal{V}$.

To prove the claim, we introduce another set $b(k)$, $k \geq k_0$, given by

$$b(k) := \{i \in \mathcal{V} : s_i(k) > 0\}. \tag{4.13}$$

By the assumption, $b(k)$ is a proper subset of $\mathcal{V}$ (namely, $b(k) \neq \emptyset, \mathcal{V}$). First, owing to the surplus update Eq. (4.6), together with (P1) and (P3), any strictly positive surplus cannot decay to zero in finite time. This indicates $b(k) \subseteq b(k+1)$, $k \geq k_0$. Next, since $\mathcal{G}(k)$ is jointly strongly connected, there is an instant $\bar{k}$ in the interval $[k, k+\mathcal{K}]$ such that a directed edge (h, j) exists, for some $h \in b(k)$ and some $j \in \mathcal{V} - b(k)$. Then node j receives surplus of the amount $b_{ij}(\bar{k})s_i(\bar{k}) > 0$, and hence $b(k)$ is strictly contained in $b(k+\mathcal{K})$. Repeating this argument leads us to the conclusion that $b(k+(n-r)\mathcal{K}) = \mathcal{V}$, which shows the claim.

Now for $k \geq k_0$ we distinguish three cases. (1) $b(k) = \mathcal{V}$. Under jointly strongly connected topology, there is a directed edge (h, j), $h \in \mathcal{V} - a(k)$ and $j \in a(k)$, for some time $\bar{k} \in [k, k+\mathcal{K}]$. Then by Eq. (4.5) and (P4) we have $x_j(\bar{k}+1) = x_j(\bar{k}) + \epsilon_j(\bar{k})s_j(\bar{k}) > x_j(\bar{k}) \geq x_j(k)$. So $a(k+\mathcal{K})$ is strictly contained in $a(k)$. (2) $b(k)$ is a proper subset of $\mathcal{V}$. It follows from the above claim that $b(k + (n-r)\mathcal{K}) = \mathcal{V}$. Then by the same argument as in case (1) we obtain that $a(k + (n-r+1)\mathcal{K})$ is strictly contained in $a(k)$. (3) $b(k) = \emptyset$. Owing again to jointly strongly connected topology, there is a directed edge (h, j), with $x_h(\bar{k}) < \overline{m}(k)$ and $x_j(\bar{k}) = \overline{m}(k)$, for some time $\bar{k} \in [k, k+\mathcal{K}]$. Then by Eqs. (4.5), (4.6), and (P4) we have $s_j(\bar{k}+1) = -(x_j(\bar{k}+1) - x_j(\bar{k})) = -a_{jh}(\bar{k})(x_h(\bar{k}) - x_j(\bar{k})) > 0$, and thereby $b(k+\mathcal{K}) = \{j\}$. Now applying the derivation in case (2) leads us to that $a(k + (n+1)\mathcal{K})$ is strictly contained in $a(k)$. Summarizing the above three cases, and letting $\tilde{\kappa} = (n+1)\mathcal{K}$, we obtain that $a(k+\tilde{\kappa})$ is strictly contained in $a(k)$.

Table 4.1: Convergence factor $\lambda_2^{(g)}$ with respect to different values of parameter ϵ.

	$\epsilon = 0.2$	$\epsilon = 0.7$	$\epsilon = 2.15$
$\mathcal{G}_a$	0.9963	1.0003	1.0020
$\mathcal{G}_b$	0.9951	0.9969	1.0000
$\mathcal{G}_c$	0.9883	0.9930	0.9993

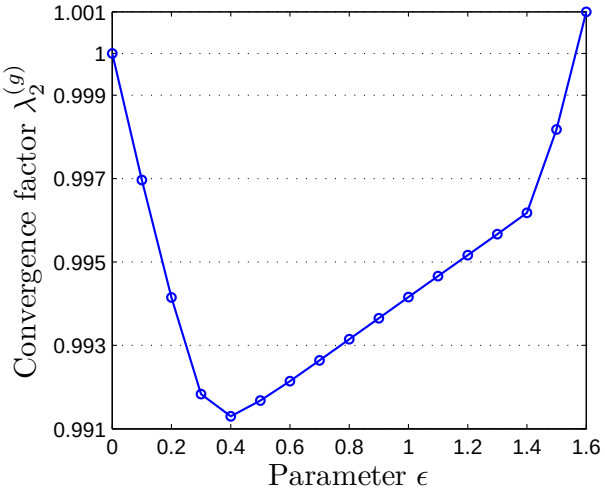


Figure 4.1: Convergence factor $\lambda_2^{(g)}$ of gossip algorithm (4.3) versus parameter ϵ.

Finally, since there are at most $n-1$ nodes in $a(k_0)$, for $\kappa := (n-1)\tilde{\kappa}$ we have $a(k_0+\kappa) = \emptyset$. This implies $\mu < \underline{m}(x(k_0+\kappa))$, and thus $V(x(k_0)) > V(x(k_0+\kappa))$ by the definition of $V(\cdot)$ in (4.10). Therefore by (4.11), $\delta_\kappa(x(k_0))$ is strictly positive with $\kappa = (n-1)(n+1)\mathcal{K}$. ■

4.5 Numerical Examples

We provide simulations to illustrate the convergence results of both gossip and dynamic algorithms.

Example 4.2. Consider again the three unbalanced digraphs in Fig. 3.5 We apply the gossip algorithm (4.3), with uniform weight $w = 1/2$ and probability $p = 1/\text{card}(\mathcal{E})$. The convergence factor $\lambda_2^{(g)}$ (see Remark 4.1) for three different values of the parameter

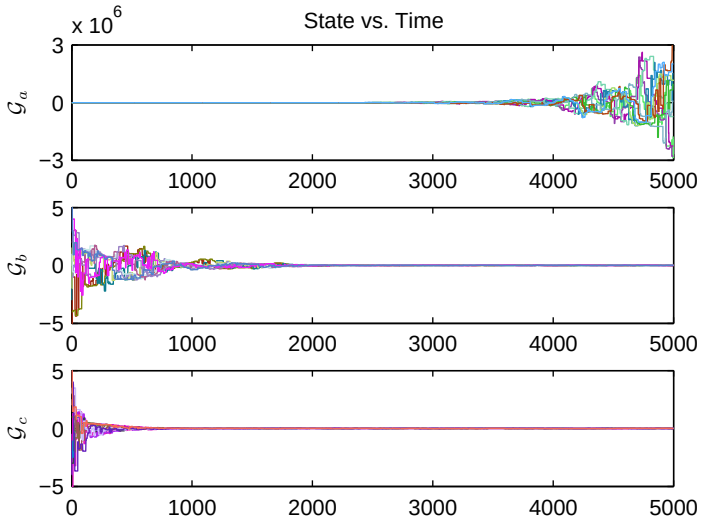


Figure 4.2: Sample paths of states: Obtained by applying gossip algorithm (4.3) with parameter $\epsilon = 0.7$ on digraphs $\mathcal{G}_a$, $\mathcal{G}_b$, and $\mathcal{G}_c$.

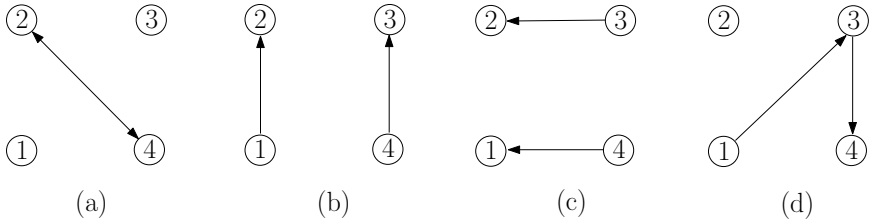


Figure 4.3: Periodically time-varying topology: a b c d a b c d $\cdots$.

ϵ are summarized in Table 4.1. We see again that small ϵ ensures convergence of the gossip algorithm (4.3) ($\lambda_2^{(g)} < 1$), whereas large ϵ can lead to instability. Also as in Example 3.4, we plot the trend of $\lambda_2^{(g)}$ as ϵ value increases in Fig. 4.1, which reflects the influence of ϵ on convergence speed. Observe that the trend is very similar to the one in Fig. 3.7 for the surplus-based algorithm (3.8).

For a random initial state $x(0)$ with the average $x_a = 0$ and the initial surplus $s(0) = 0$, we display in Fig. 4.2 the state trajectories when the gossip algorithm (4.3) is applied on digraph $\mathcal{G}_a$, $\mathcal{G}_b$, and $\mathcal{G}_c$ with parameter $\epsilon = 0.7$. While convergence fails on $\mathcal{G}_a$ ($\lambda_2^{(g)} = 1.0003$), asymptotic state averaging is achieved on $\mathcal{G}_b$ and $\mathcal{G}_c$.

Example 4.3. We turn now to illustrating the convergence of the dynamic algorithm (4.7). Consider the periodically time-varying digraph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ displayed in Fig. 4.3. No single digraph is strongly connected or balanced, but one quickly verifies

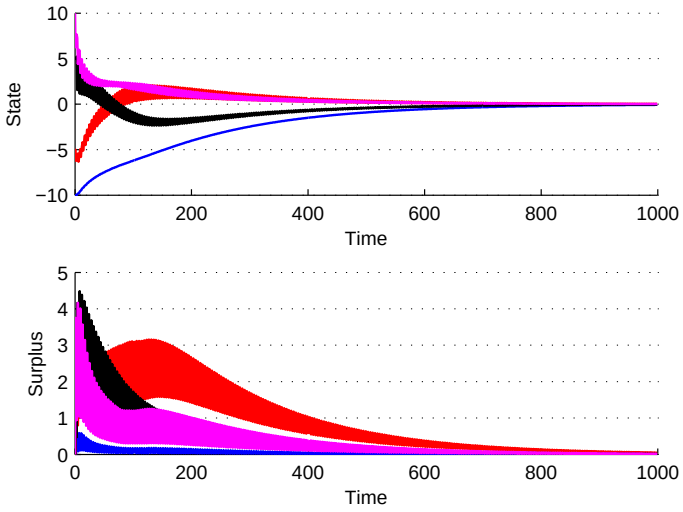


Figure 4.4: Periodically time-varying topology: Convergence trajectories of states and surpluses.

that $\mathcal{G}(k)$ is jointly strongly connected. We apply the algorithm (4.7) by choosing $\epsilon = a_{ij} = b_{ij} = 1/4$ for all edges (j, i). This choice satisfies the requirements **(P1)**-**(P3)**. For the initial state $x(0) = [-10\ 5\ 5\ 10]^T$, the state and surplus trajectories are displayed in Fig. 4.4. Observe that every state converges to the desired average 0, and every surplus is always nonnegative and vanishes eventually.

Chapter 5

Quantized Consensus and Averaging on Random Digraphs

5.1 Introduction

In this and next chapters, we embark on a new, and more realistic, setup (in contrast, in particular, with the basic one investigated in Chapter 3) to study multi-agent consensus and averaging problems. In this setup, agents' individual actions are *asynchronous*: there does not exist a global clock to which all agents may refer when executing their local commands; and their states are *quantized*: this is necessary especially when communication channels between agents are digital and of limited date rate, e.g., wireless networks. The setup change, as we will see, makes many previously employed tools not applicable, including nonnegative matrix theory and matrix spectrum analysis. Thus new methods must be sought, and we find *finite Markov chain theory* is instrumental for analysis in the current setting.

To describe asynchronous behavior of a network of agents, as in Chapter 4 we employ again the gossip randomization model [12, 22, 29, 45, 48]. This model specifies that, in the case of digraphs, at each time exactly *one* agent updates its state based on the information transmitted from only *one* of its neighbors. On the other hand, to model quantization effect, following [45] we assume at the outset that the states

are *integer-valued*, an abstraction that subsumes a class of quantization effect (e.g., uniform quantization). We note that most work dealing with quantization has concentrated on the scenario where agents have real-valued states but can transmit only quantized values through limited rate channels (see, e.g., [22, 23, 48]). By contrast, our assumption is suited to the case where the states are stored in physical memories that are also of finite capacity, as in [45, 57].

The central investigation in this chapter is to derive connectivity conditions on digraphs that ensure both general consensus (where the final consensus value is not specified a priori) and average consensus. First, for general consensus we present a class of algorithms, by which we derive a necessary and sufficient condition on digraph connectivity that guarantees convergence to some common state. This result extends those in [11, 51, 74] from real-valued to quantized states. Second, for averaging we propose a novel class of algorithms based again on surplus, and derive that an arbitrary strongly connected digraph (as in the preceding chapters) is necessary and sufficient to ensure convergence to the quantized average value. This result extends the one in [45] from undirected graphs to digraphs; the extension is challenging because with digraphs of gossip type, the state sum, and hence the average, need not be invariant at each iteration. Also, the graphical condition we find is weaker than those in [23, 74], since we do not require maintaining symmetric or balanced topologies in random time-varying networks. As a tradeoff, however, the convergence rate of the proposed algorithm may not be fast, for which we shall discuss in detail in Chapter 6. Lastly, our result is *scalable* compared to [22, 33, 57] in the sense that the true (quantized) average is always achieved regardless of the number of agents. These points of improvement come with a cost in communication, which can be, nevertheless, relaxed to two bits in addition to the integer state in the transmission at each time.

The rest of this chapter is organized as follows. First, we formulate both general and average consensus problems in Section 5.2, and then present their solutions in Sections 5.3 and 5.4, respectively. Further expositions of our solution to average consensus are given in Section 5.5, where we discuss two featured elements; and a modified

averaging algorithm is proposed in Section 5.6 with the purpose of reducing computation burden. Finally, illustrative numerical examples are provided in Section 5.7. The material in this chapter has appeared in [17, 18].

5.2 Problem Formulation

Consider a network of n (> 1) agents modeled by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Owing to quantization in information flow, we assume that at time $k \in \mathbb{Z}_+$, each agent has an integer-valued state $x_i(k) \in \mathbb{Z}$, $i \in \mathcal{V}$; the aggregate state is denoted by $x(k) = [x_1(k) \cdots x_n(k)]^T \in \mathbb{Z}^n$. We will design algorithms with which every agent updates its state such that all $x_i(k)$ eventually converge to a common value.

An important feature of distributed networks is that agents acting locally need not be precisely synchronized by a common, global clock. To address this asynchronism we model the communication graph in such a way that the agents "gossip" with one another at random. Specifically, at each time instant k exactly one edge, say (j, i), is activated independently from all earlier instants and with a time-invariant, strictly positive probability $p_{ji} \in (0, 1)$ such that $\sum_{(j,i)\in\mathcal{E}} p_{ji} = 1$. Along this activated edge, node j sends information to i, while i receives the information and makes an update accordingly.

In the first part of this chapter, we consider the general consensus problem as described below. Let the subset $\mathscr{C}$ of $\mathbb{Z}^n$ be the set of consensus states:

$$\mathscr{C} := \{x : x_1 = \cdots = x_n\}. \tag{5.1}$$

Definition 5.1. The network of agents achieves *quantized consensus almost surely* if for every initial condition $x(0)$, there exist K and $x^* \in \mathscr{C}$ such that $x(k) = x^*$ for all $k \geq K$ with probability one.

Problem 5.1. Design a distributed algorithm and find suitable connectivity such that agents achieve quantized consensus almost surely.

For this problem, in Section 5.3 we will propose a class of algorithms, under which

we derive a necessary and sufficient graphical condition that guarantees almost sure quantized consensus.

In the second part, we extend the above problem to average consensus by further requiring that the consensus value be the average of the initial state sum. Formally, let $S := x(0)^T\mathbf{1}$. Hence the average of the initial states is S/n, a number that need not be an integer in general. We can, however, always write $S = nL + R$, where L and R are both integers with $0 \leq R < n$. Thus, either L or $L+1$ (the latter if $R > 0$) may be viewed as an integer approximation of the average S/n. Henceforth we refer to $x_{\text{ave}} := L\mathbf{1}$ or $(L+1)\mathbf{1}$ as the *true (quantized) average*.

To ensure convergence to the average, the algorithms reported in the literature (e.g., [45, 61]) rely on a key property that the state sum $x^T\mathbf{1}$ remains invariant at each iteration. Unfortunately, this property in general fails in our gossip digraph setup where only one agent is allowed to update its state at each time. To overcome this difficulty, we again propose associating to each agent a surplus variable to record the changes in individual states; then the agents communicate these records to their neighbors such that this important information can be utilized for state updates. The rules of how to use these surpluses mark the distinctive feature of the averaging algorithm for integer-valued states; the concrete description is deferred to Section 5.4.

Formally, let the surplus of agent $i \in \mathcal{V}$ at time k be $s_i(k) \in \mathbb{Z}$; thus the aggregate surplus is $s(k) = [s_1(k) \cdots s_n(k)]^T \in \mathbb{Z}^n$, the initial value of which is set to be $s(0) = [0 \cdots 0]^T$. As described, the surplus is introduced to make the quantity $(x + s)^T\mathbf{1}$ invariant during iterations, i.e., for each $k \geq 0$,

$$(x(k) + s(k))^T\mathbf{1} = (x(0) + s(0))^T\mathbf{1} = nL + R. \tag{5.2}$$

Consequently, $s^T\mathbf{1} = R$ (≥ 0) if $x = L\mathbf{1}$, and $R - n$ (< 0) if $x = (L+1)\mathbf{1}$. Now define the set $\mathscr{A}$ of the average consensus states, which is a subset of $\mathbb{Z}^n \times \mathbb{Z}^n$, by

$$\mathscr{A} := \begin{cases} \mathscr{A}_L, & \text{if } R = 0; \\ \mathscr{A}_L \cup \mathscr{A}_{L+1}, & \text{if } 0 < R < n, \end{cases} \tag{5.3}$$

where

$$\mathscr{A}_L := \{(x,s) : x_i = L \ \&\ s_i \geq 0,\ i = 1,...,n\},$$
$$\mathscr{A}_{L+1} := \{(x,s) : x_i = L+1 \ \&\ s_i \leq 0,\ i = 1,...,n\}.$$

Definition 5.2. The network of agents is said to achieve *quantized averaging almost surely* if for every initial condition $(x(0), 0)$, there exist K and $(x^*, s^*) \in \mathscr{A}$ such that $(x(k), s(k)) = (x^*, s^*)$ for all $k \geq K$ with probability one.

It is worth noting that our definition of average consensus differs from that in [45]: We require that all agents' states converge to an identical integer (either L or $L+1$), a property that cannot be achieved in general with the proposed algorithm in [45] due to the "swap" operation.

Problem 5.2. Design a distributed algorithm and find graphical connectivity such that the agents achieve quantized averaging almost surely.

To solve this problem, in Section 5.4 we will propose a novel class of algorithms, under which we derive a necessary and sufficient graphical condition that guarantees almost sure quantized average.

5.3 Quantized Consensus by Gossip Algorithm

In this section we first solve Problem 5.1, the almost sure quantized consensus. We start by presenting a class of algorithms, which we call *quantized consensus* (**QC**) algorithm. Then we prove convergence to quantized consensus under a certain graphical condition.

5.3.1 QC Algorithm

Here we present **QC** algorithm. Suppose that every edge of the digraph $\mathcal{G}$ has a time-invariant, strictly positive probability of being activated. Say edge $(j,i) \in \mathcal{E}$ is activated at time k. Along the edge node j sends to i its state information, $x_j(k)$,

but does not perform any update, i.e., $x_j(k+1) = x_j(k)$. On the other hand, node i receives j's state $x_j(k)$ and updates its own as follows:

(R1) If $x_i(k) = x_j(k)$, then $x_i(k+1) = x_i(k)$;

(R2) if $x_i(k) < x_j(k)$, then $x_i(k+1) \in (x_i(k), x_j(k)]$;

(R3) if $x_i(k) > x_j(k)$, then $x_i(k+1) \in [x_j(k), x_i(k))$.

In words, node i stays put if its own state is the same as the received one; otherwise, it updates the state in the direction of diminishing the difference.

5.3.2 Convergence Result

It is convenient to define the minimum and maximum states of $x(k)$, $k \geq 0$, for the set $\mathcal{V}$ by

$$m(k) := \min_{i \in \mathcal{V}} x_i(k), \;\; M(k) := \max_{i \in \mathcal{V}} x_i(k). \tag{5.4}$$

Let $\mathcal{V}_g \subseteq \mathcal{V}$ denote the subset of all globally reachable nodes, and similarly to (5.4), define $m_g(k)$, $M_g(k)$ for $\mathcal{V}_g$.

We present the main result of this section.

Theorem 5.1. Using **QC** algorithm, a network of agents achieves quantized consensus almost surely if and only if the digraph $\mathcal{G}$ has a globally reachable node. Moreover, the consensus value lies between $m_g(k)$ and $M_g(k)$, for every $k \geq 0$.

It has been known (e.g., [51, 56, 65, 74]) that the existence of a globally reachable node is a necessary and sufficient graphical condition which ensures consensus in the case of real-valued states. In this respect, Theorem 5.1 extends the result to the setting where both stored and communicated states are quantized. For the consensus value, however, the left-eigenvector characterization for real states (e.g., [65, 74]) is no longer valid in the quantized state case. Instead, it turns out that the consensus value lies in the smallest interval containing all the states of globally reachable nodes.

Our analysis technique, provided below, is a blend of graph-theoretic and probabilistic arguments. Specifically, for the probabilistic portion we borrow the proof structure from [45], and extend the argument from undirected to directed graphs. We will see that this extension requires some insight into digraph structure. For the graph-theoretic part, we utilize a fact that relates digraph connectivity to its structure. This approach differs from the typical one (e.g., [42, 61]) that exploits the spectral properties of the Laplacian matrix associated to the graph structure. Indeed, owning to our integer state setup, the overall system does not enjoy a linear representation, and consequently the matrix approach cannot be applied.

Lastly, notice that the rules **(R2)** and **(R3)** of **QC** algorithm can be chosen so that the algorithm is similar to those for the real-valued case. Hence, we conjecture that the convergence rate of **QC** algorithm may be close to that of real-valued algorithms [29]. This conjecture is supported by the numerical example studied in Section 5.7.

Before providing the proof of Theorem 5.1, we introduce some preliminary results.

Lemma 5.1. The agents achieve quantized consensus almost surely if the following conditions hold:

(C1) The evolution of $x(k)$, $k \geq 0$, is a Markov chain with a finite state space;

(C2) if $x(k) = x^* \in \mathscr{C}$ in (5.1), then $x(k') = x^*$ for all $k' > k$;

(C3) for every $k \geq 0$ there is a finite time $K_{qc} \geq k$ such that $\Pr\big[x(K_{qc}) \in \mathscr{C} \mid x(k)\big] > 0$.

See [45] for the proof. Similar results may also be found in Markov chain theory (e.g., [58]).

The next result ensures that in the special case where the communication digraph is strongly connected, the condition **(C3)** in Lemma 5.1 holds. Further, the consensus value lies between $m(k)$ and $M(k)$.

Lemma 5.2. Consider **QC** algorithm. If the digraph $\mathcal{G}$ is strongly connected, then for each $k \geq 0$ and $j \in \mathcal{V}$ there is a finite time $K_{qc} \geq k$ such that $\Pr\big[x(K_{qc}) = x_j(k)\mathbf{1} \mid x(k)\big] > 0$.

Proof. Fix $j \in \mathcal{V}$; then $x_j(k) \in [m(k), M(k)]$. We consider the following three cases.

Case 1: $x_j(k) = M(k)$. Define $\mathcal{I}_m(k) := \{i : x_i(k) = m(k)\}$, and its cardinality $n_m(k) := |\mathcal{I}_m(k)|$; also let $\mathcal{I}_m^c(k) := \{i : x_i(k) \geq m(k) + 1\}$. Since $\mathcal{G}$ is strongly connected, there is an edge from $\mathcal{I}_m^c(k)$ to $\mathcal{I}_m(k)$. Activate this edge with a positive probability; then **(R2)** of **QC** algorithm applies, causing $n_m(k)$ to decrease by 1. Repeatedly, $n_m(k)$ can decrease to zero with a positive probability, which implies that there is $k_1 > k$ such that $\Pr\big[m(k_1) > m(k) \mid x(k)\big] > 0$. We repeat the above argument to derive that there is $K_{qc} > k$ such that $\Pr\big[m(K_{qc}) = M(K_{qc}) \mid x(k)\big] > 0$.

Case 2: $x_j(k) = m(k)$. The argument is symmetric to that of Case 1. We point out that, in the present case, **(R3)** of **QC** algorithm is repeatedly applied (as **(R2)** in the previous case).

Case 3: $x_j(k) \in (m(k), M(k))$. The conclusion follows by suitably combing the two cases above. ■

Now we are ready to prove Theorem 5.1.

Proof of Theorem 5.1. (Necessity) Suppose that $\mathcal{G}$ does not have a globally reachable node. By Lemma 2.1, $\mathcal{G}$ has at least two distinct closed strong components, say $\mathcal{V}_1$ and $\mathcal{V}_2$. Consider some initial condition $x(0)$ such that all nodes in $\mathcal{V}_1$ have the same state $a \in \mathbb{Z}$ and all nodes in $\mathcal{V}_2$ have $b \in \mathbb{Z}$, but $a \neq b$. Then the quantized consensus is achieved *almost never* (with probability 0), for both $\mathcal{V}_1$ and $\mathcal{V}_2$ are closed.

(Sufficiency) In light of Lemma 5.1, it suffices to establish the three conditions **(C1)**–**(C3)**. For **(C1)** and **(C2)**, one may readily verify that they hold under **QC** algorithm without any connectivity assumption. Thus, it remains to show that **(C3)** holds when $\mathcal{G}$ has a globally reachable node.

If $\mathcal{V}_g = \mathcal{V}$, then $\mathcal{G}$ is strongly connected, and hence **(C3)** holds by Lemma 5.2. Otherwise, let $\mathcal{G}_g$ be the induced subdigraph by $\mathcal{V}_g$. It then follows from Lemma 2.1 that $\mathcal{G}_g$ is the unique closed strong component of $\mathcal{G}$. We apply Lemma 5.2 for $\mathcal{G}_g$ and derive that there exist a positive probability and a finite time $k_0 \geq k$ such that $x_{v_g}(k_0) = x_{qc}$ for all nodes $v_g \in \mathcal{V}_g$; evidently the integer x_{qc} is in $[m_g(k), M_g(k)]$.

Now define $\mathcal{I}(k_0) := \{v \in \mathcal{V} - \mathcal{V}_g : x_v(k_0) \neq x_{qc}\}$, and its cardinality $n(k_0) :=$

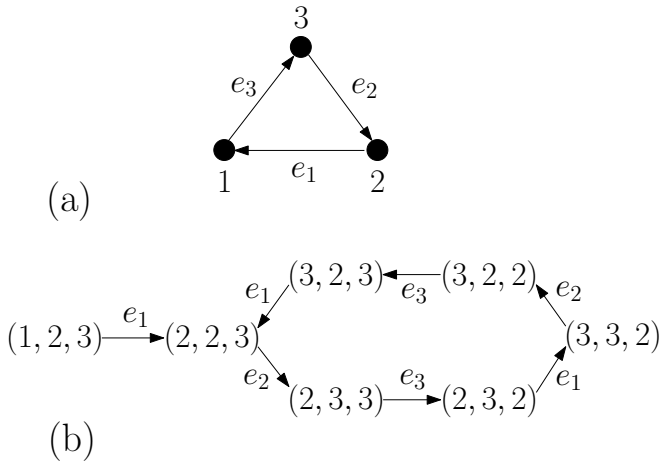


Figure 5.1: Without randomization the agents may fail to achieve quantized consensus.

$|\mathcal{I}(k_0)|$; also let $\bar{\mathcal{V}}_g(k_0) := \mathcal{V} - \mathcal{I}(k_0)$. Since the nodes in $\mathcal{V}_g$ are globally reachable, there is an edge from $\bar{\mathcal{V}}_g(k_0)$ to $\mathcal{I}(k_0)$, say (q, p) with $q \in \bar{\mathcal{V}}_g(k_0)$ and $p \in \mathcal{I}(k_0)$. Activate this edge with a positive probability, and **(R2)** of **QC** algorithm applies if $x_p(k_0) < x_q(k_0)$, or otherwise (i.e., $x_p(k_0) > x_q(k_0)$) **(R3)** applies; either update causes p's state to approach x_{qc}. Repeatedly, there is $k_1 > k_0$ such that $x_p(k_1) = x_{qc}$; so $\Pr\big[n(k_1) = n(k_0) - 1 \mid x(k)\big] > 0$. We repeat the above argument to derive that there is $K_{qc} > k$ such that $\Pr\big[n(K_{qc}) = 0 \mid x(k)\big] > 0$, which implies $\Pr\big[x(K_{qc}) = x_{qc}\mathbf{1} \in \mathscr{C} \mid x(k)\big] > 0$. Therefore **(C3)** follows, and the consensus value is x_{qc}. ■

5.3.3 Role of Randomization

We provide an example which shows that the gossip randomization, in addition to modeling asynchronous behavior, can be crucial to ensure quantized consensus under **QC** algorithm. A similar example, but for the case of undirected graphs, was reported in [45].

Example 5.1. Consider three agents in *cyclic pursuit* (see Fig. 5.1(a)), with the initial condition $x(0) = [1, 2, 3]^T$. Suppose that **QC** algorithm is used, but that the network is non-randomized and the edges are activated periodically as follows:

The corresponding state evolution is displayed in Fig. 5.1(b). We see that the

time k	0	1	2	3	4	5	$\cdots$
edge	e_1	e_2	e_3	e_1	e_2	e_3	$\cdots$

evolution is, deterministically, trapped in a loop containing no consensus state. By contrast, randomizing edge selection ensures that the evolution can break the loop with a positive probability, thereby leading to almost sure consensus.

This example thus marks a fundamental distinction between the integer- and the real-state settings. With real-valued states, it is well known [50,56,65] that consensus is guaranteed if $\mathcal{G}(k)$ has a globally reachable node *uniformly*: That is, there exists an integer $T > 0$ such that for every k_0 the union $\bigcup_{k_0}^{k_0+T} \mathcal{G}(k)$ has a globally reachable node. This condition clearly holds in this example for every $T \geq 2$; quantized consensus, however, fails.

5.4 Quantized Averaging by Gossip Algorithm

We move on to solve Problem 5.2, the quantized averaging, by appropriately extending **QC** algorithm studied in the previous section. A direct application of **QC** algorithm in general fails to ensure convergence to the quantized average, because the state sum need not be invariant at each iteration, hence causing the shift of the average. To handle this average shift, we again propose associating to each agent a surplus variable. These surpluses are used to keep track of the state changes of individual agents, so that the information of the amount of average shift is not lost but kept *locally* in these variables. Then agents communicate the surpluses to their neighbors for state updates in such a way that the average of the initial states may be recovered. Further, to assist the use of surpluses, two more auxiliaries are needed, which we call threshold and local extrema. We use these three augmented elements to make the extension of **QC** algorithm.

In the sequel, we first present the extended algorithm, which we call *quantized averaging* (**QA**) algorithm. Then we prove convergence to quantized average under general strongly connected digraphs.

5.4.1 QA Algorithm

First, we introduce the three augmented elements.

1. *Surplus.* Every agent is associated with a surplus variable to record its state changes. Recall from Section 5.2 that the surplus of agent $i \in \mathcal{V}$ is denoted by s_i. Thus the aggregate surplus is $s = [s_1 \cdots s_n]^T$, whose initial value is set to be $s(0) = [0 \cdots 0]^T$. The rules of specifying how these surpluses are updated locally and communicated over the network form the core of **QA** algorithm.

2. *Threshold.* All agents have a common threshold, denoted by $\delta \in \mathbb{Z}_+$. This (constant) number is involved in deciding whether or not to update a state using available surpluses. A proper value for the threshold will be found crucial to ensure that the set $\mathscr{A}$ defined in (5.3) is the unique invariant set where all trajectories converge. We shall determine the range of such threshold values in Section 5.5. To keep the presentation clear, in this section we fix $\delta = n$, the total number of agents in the network. Thus, every agent is required to know this information.

3. *Local extrema.* Each agent i is further assigned two variables, m_i and $M_i \in \mathbb{Z}$, to record respectively the minimal and maximal states among itself and its neighbors. These *local* extrema will be used to prevent a state, when updated by available surpluses, from exceeding the interval of all initial states (i.e., $[m(0), M(0)]$). For the initial values of local extrema we set $m_i(0) = M_i(0) = x_i(0)$, for every $i \in \mathcal{V}$. We will design updating rules for m_i and M_i as part of **QA** algorithm. The necessity of using local extrema in the algorithm will be exhibited in Section 5.5.

Thus, we have augmented the state of each agent i from a single x_i to a tuple of four elements (x_i, s_i, m_i, M_i). In addition, a common threshold δ needs to be stored. Also note that only x_i and s_i will be involved in communication.

We are now ready to present **QA** algorithm. Suppose that every edge of the communication digraph $\mathcal{G}$ has a (time-invariant) strictly positive probability of being activated. Say edge $(j, i) \in \mathcal{E}$ is activated at time k. Along the edge, node j sends to i its state information, $x_j(k)$, as well as its surplus, $s_j(k)$. While it does not perform any update on its state (nor on its local minimum and maximum), node j does always set

its surplus to be 0 after transmission, meaning that the surpluses, if any, are entirely passed to its neighbor i; that is,

$$m_j(k+1) = m_j(k), \quad M_j(k+1) = M_j(k),$$
$$x_j(k+1) = x_j(k), \quad s_j(k+1) = 0.$$

On the other hand, node i receives the information sent from j, namely $x_j(k)$ and $s_j(k)$, and performs the following updates.

1. For local minimum and maximum,

$$m_i(k+1) = \min\{m_i(k), x_j(k)\},$$
$$M_i(k+1) = \max\{M_i(k), x_j(k)\}.$$

2. State and surplus are updated as follows:

(R1) If $x_i(k) = x_j(k)$, then there are three cases:

(i) If $s_i(k) + s_j(k) \geq \delta$ and $x_i(k) \neq M_i(k+1)$, then

$$x_i(k+1) = x_i(k) + 1, \quad s_i(k+1) = s_i(k) + s_j(k) - 1.$$

(ii) If $s_i(k) + s_j(k) \leq -\delta$ and $x_i(k) \neq m_i(k+1)$, then

$$x_i(k+1) = x_i(k) - 1, \quad s_i(k+1) = s_i(k) + s_j(k) + 1.$$

(iii) Otherwise (i.e., $|s_i(k) + s_j(k)| < \delta$ or $s_i(k) + s_j(k) \geq \delta$ & $x_i(k) = M_i(k)$ or $s_i(k) + s_j(k) \leq -\delta$ & $x_i(k) = m_i(k)$),

$$x_i(k+1) = x_i(k), \quad s_i(k+1) = s_i(k) + s_j(k).$$

(R2) If $x_i(k) < x_j(k)$, then

$$x_i(k+1) \in (x_i(k), x_j(k)], \quad s_i(k+1) = s_i(k) + s_j(k) - \big(x_i(k+1) - x_i(k)\big).$$

(R3) If $x_i(k) > x_j(k)$, then

$$x_i(k+1) \in [x_j(k), x_i(k)), \quad s_i(k+1) = s_i(k) + s_j(k) - \big(x_i(k+1) - x_i(k)\big).$$

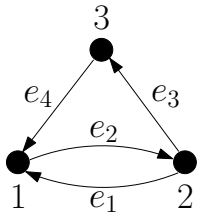


Figure 5.2: Illustration of features of **QA** algorithm

In the algorithm, first observe that the surplus is updated such that for every $k \geq 0$, $(x(k+1) + s(k+1))^T \mathbf{1} = (x(k) + s(k))^T \mathbf{1} = x(0)^T \mathbf{1}$. That is, the quantity $(x+s)^T \mathbf{1}$ stays invariant at each iteration, and thus equals the initial state sum. Also, notice that the updates of state x_i in **(R2)** and **(R3)** are exactly the same as those in **QC** algorithm. The difference, however, lies in **(R1)**: Even when the state x_i coincides with x_j, it is still updated if the sum of surpluses, $s_i + s_j$, exceeds the interval $(-\delta, \delta)$; here this interval is $(-n, n)$. This is because, when the surpluses are more than n (resp., less than $-n$), the true average must be at least $x_i + 1$ (resp., $x_i - 1$). Indeed, these surpluses should be distributed over the network such that every agent's state increases by at least 1 (resp., decreases by 1). An exception, however, is when x_i equals its local maximum (resp., local minimum), since in that case, x_i could undesirably exceed $[m(0), M(0)]$. We illustrate these features of **QA** algorithm in the following example.

Example 5.2. Consider three agents with communication network displayed in Fig. 5.2. Let the initial condition be as follows:

agent i	$x_i(0)$	$s_i(0)$	$m_i(0)$	$M_i(0)$
1	0	0	0	0
2	3	0	3	3
3	3	0	3	3

Hence the true average is $x_{\text{ave}} = 2\mathbf{1}$. Suppose that at $k = 0$, edge e_1 is activated with a positive probability; then **(R2)** of **QA** algorithm applies since $x_1(0) < x_2(0)$. For the possible update values $(x_1(0), x_2(0)]$ we let $x_1(1) = x_2(0)$; the corresponding state

change, $x_1(1) - x_1(0)$, is recorded in the surplus $s_1(1)$. Thus we obtain that

agent i	$x_i(1)$	$s_i(1)$	$m_i(1)$	$M_i(1)$
1	3	-3	0	3
2	3	0	3	3
3	3	0	3	3

Now the agents reach consensus at value 3. If **QC** algorithm is used, then no further update will take place, and consequently the true average cannot be achieved. However, that agent 1 has surplus -3 $(= -n)$ indicates that this amount should be distributed among the three agents, thereby each decreasing its state by 1. One way to distribute the surplus is to select the edges e_4, e_2, and e_3 sequentially; the probability of this selection is positive. It can then be readily verified that **(R1)(ii)**, **(R3)**, and again **(R3)** of **QA** algorithm will sequentially apply, and that at $k = 4$ we have

agent i	$x_i(4)$	$s_i(4)$	$m_i(4)$	$M_i(4)$
1	2	0	0	3
2	2	0	2	3
3	2	0	2	3

Therefore, the true average x_{ave} is achieved, and there is no further update because only **(R1)(iii)** will apply.

5.4.2 Convergence Result

We present the main result of this section.

Theorem 5.2. Using **QA** algorithm, a network of agents achieves quantized averaging almost surely if and only if the digraph $\mathcal{G}$ is strongly connected.

The necessity and sufficiency proofs of Theorem 5.2 will be provided in the next subsection. Presently we draw some remarks on this result, in comparison with those related in the literature. First, Theorem 5.2 can be seen as an extension of the main

result in [45] from undirected to directed graphs. The problem of achieving quantized average with directed graphs is, however, more difficult in that the state sum need not be invariant at each iteration. Our proposed **QA** algorithm handles this difficulty, by an essential augment of surplus variables. In addition, we note that in some quantized consensus algorithms (e.g., [22, 33, 57]), the agents converge to the average with an error which could undesirably get large as the number of agents increases. To address this *unscalable* situation, several approaches are proposed using special graph topologies [33], finer quantizers [57], and probabilistic quantizers [22]. In contrast, our result ensures, for a general (strongly connected) graph and a fixed (deterministic) quantizer, that the quantized average is always achieved regardless of the number of agents.

The foregoing merits, however, come with some costs which are twofold: For one, the convergence rate of **QA** algorithm is in general slower than that of **QC** algorithm due to averaging (see a demonstration in Section 5.7). This requires additional processing based on surpluses even after the agents achieve consensus (not at the average). For the other, as to local memories each agent needs to update, in addition to its state, three more variables — surplus, local minimum, and local maximum — and needs to store a constant threshold. The corresponding updating computations are, however, purely local and fairly simple. Moreover, each agent has to transmit surpluses, along with its state, through communication channels.

5.4.3 Proof of Theorem 5.2

The necessity argument is the same as Theorem 3.1. Before proceeding to the sufficiency part, we need to establish two key lemmas. For their proofs, see Section 5.8. Henceforth in this subsection, we assume that **QA** algorithm is used and the digraph $\mathcal{G}$ is strongly connected.

For an arbitrary pair of state and surplus $(x(k), s(k)) \in \mathbb{Z}^n \times \mathbb{Z}^n$, $k \geq 0$, define $m(k), M(k)$ as in (5.4). In the case where all nodes have the same state (i.e., $m(k) = M(k)$), our first result asserts that there is a positive probability such that, in finite

time, all surpluses in the system can pile up at a single node.

Lemma 5.3. Suppose that at time $k \geq 0$, the pair $(x(k), s(k))$ is such that $m(k) = M(k)$. Fix an arbitrary node $i \in \mathcal{V}$. Then there exists a finite time $K_s > k$ such that

$$\Pr\big[x(K_s) = x(k),\ s_i(K_s) = s_1(k) + \cdots + s_n(k),$$
$$(\forall j \neq i)\ s_j(K_s) = 0 \mid (x(k), s(k))\big] > 0.$$

Next, recall from (5.2) that $(x(0) + s(0))^T \mathbf{1} = nL + R$, where $R \in [0, n-1]$. As the quantity $(x+s)^T \mathbf{1}$ is invariant, if all states are identical to $L - \alpha$ for some $\alpha \geq 1$, then the total surplus in the system is $s^T \mathbf{1} = R + \alpha n$. Now suppose that one node i increases its state to $L - \alpha + 1$ and has all the surpluses $R + \alpha n - 1$. In order to approach the set $\mathscr{A}$ defined in (5.3), it is desired that other nodes follow i to the state $L - \alpha + 1$, thereby decreasing the total surplus to $R + (\alpha - 1)n$. Our second result asserts that this can be done in finite time with a positive probability.

Lemma 5.4. Suppose that at time $k \geq 0$, the pair $(x(k), s(k))$ is such that for one node i

$$x_i(k) = L - \alpha + 1, \quad s_i(k) = R + \alpha n - 1,$$

and for other nodes $j \neq i$

$$x_j(k) = L - \alpha, \quad s_j(k) = 0.$$

Then there exists a finite time $K_u > k$ such that

$$\Pr\big[m(K_u) = M(K_u) = L - \alpha + 1,\ s_i(K_u) = R + (\alpha - 1)n,$$
$$(\forall j \neq i)\ s_j(K_u) = 0 \mid (x(k), s(k))\big] > 0.$$

Proof of sufficiency. Similar to Lemma 5.1, it suffices to establish the following three conditions:

(C1) The evolution of $(x(k), s(k))$, $k \geq 0$, is a Markov chain with a finite state space;

(C2) if $(x(k), s(k)) \in \mathscr{A}_L$ (resp., $\mathscr{A}_{L+1}$) in (5.3), then $(x(k'), s(k')) \in \mathscr{A}_L$ (resp., $\mathscr{A}_{L+1}$) for all $k' > k$;

(C3) for every $k \geq 0$ there is a finite time $K_{qa} \geq k$ such that $\Pr\big[(x(K_{qa}), s(K_{qa})) \in \mathscr{A} \mid (x(k), s(k))\big] > 0$.

For **(C1)**: Letting $k \geq 0$, we must show that

$$\begin{aligned}&\Pr\big[(x(k+1), s(k+1)) \mid (x(k), s(k)), ..., (x(0), s(0))\big]\\ =&\Pr\big[(x(k+1), s(k+1)) \mid (x(k), s(k))\big].\end{aligned}$$

This follows directly from the gossip setup where at time k one edge is activated at random and independently from all earlier instants. Next, for finiteness we will show first for the state $x(k)$, and then for the surplus $s(k)$.

1) For $x(k)$ it will be shown, by induction, that for all $k \geq 0$ it holds $(\forall i \in \mathcal{V})\ x_i(k), m_i(k), M_i(k) \in \big[m(0), M(0)\big]$. This is clearly true for $k = 0$. Suppose that $(\forall i \in \mathcal{V})\ x_i(k-1), m_i(k-1), M_i(k-1) \in \big[m(0), M(0)\big]$. It then follows from the updating rules of local extrema in **QA** algorithm that $(\forall i \in \mathcal{V})\ m_i(k), M_i(k) \in \big[m(0), M(0)\big]$. Now for state, assume on the contrary that there exists some node i such that $x_i(k) \notin \big[m(0), M(0)\big]$. Consider the case $x_i(k) > M(0)$; this can occur only when **(R1)(i)** of **QA** algorithm applies to the following situation: At time $k-1$, for some node j the edge (j, i) is activated, and the following conditions are met:

$$\begin{aligned}&x_i(k-1) = x_j(k-1) = M(0),\\ &s_i(k-1) + s_j(k-1) \geq \delta,\\ &x_i(k-1) \neq M_i(k-1).\end{aligned}$$

But the first and third conditions together imply that $M_i(k-1) > M(0)$, which contradicts the hypothesis. The argument for the other case $x_i(k) < m(0)$ is just symmetric; a contradiction arises between the conditions that satisfy **(R1)(ii)** of **QA** algorithm and the hypothesis. Therefore $(\forall i \in \mathcal{V})\ x_i(k) \in \big[m(0), M(0)\big]$, and hence a trivial upper bound for the set of states $x(k)$ is $(M(0) - m(0) + 1)^n$.

2) For $s(k)$, it follows from $(\forall i \in \mathcal{V})\ x_i(k) \in \big[m(0), M(0)\big]$ that the minimal and maximal values that the surpluses can take are respectively $m(0) - M(0)$ and $M(0) - m(0)$; namely, $(\forall i \in \mathcal{V})\ s_i(k) \in \big[m(0) - M(0), M(0) - m(0)\big]$. Hence the set of surpluses $s(k)$ is finite, a trivial upper bound on its cardinality being $\big(2(M(0) - m(0)) + 1\big)^n$.

For **(C2)**: First consider the case $(x(k), s(k)) \in \mathscr{A}_L$, i.e.,

$$(\forall i \in \mathcal{V})\ x_i(k) = L,$$
$$s_i(k) \geq 0,\ \sum_{i=1}^{n} s_i(k) = R.$$

Then for an arbitrary edge $(h, j) \in \mathcal{E}$ activated,

$$x_h(k) = x_j(k),$$
$$s_h(k) + s_j(k) \leq \sum_{i=1}^{n} s_i(k) = R < n.$$

Recall that the threshold is $\delta = n$. Thus **(R1)(iii)** of **QA** algorithm applies, and the subsequent states and surpluses satisfy $(x(k'), s(k')) \in \mathscr{A}_L$ for all $k' > k$. Next, consider the other case $(x(k), s(k)) \in \mathscr{A}_{L+1}$ (when $R > 0$), i.e.,

$$(\forall i \in \mathcal{V})\ x_i(k) = L + 1,$$
$$s_i(k) \leq 0,\ \sum_{i=1}^{n} s_i(k) = R - n.$$

Similarly, for an arbitrary edge $(h, j) \in \mathcal{E}$ activated,

$$x_h(k) = x_j(k),$$
$$s_h(k) + s_j(k) \geq \sum_{i=1}^{n} s_i(k) = R - n > -n.$$

Again **(R1)(iii)** of **QA** algorithm applies, and hence $(x(k'), s(k')) \in \mathscr{A}_{L+1}$ for all $k' > k$.

For **(C3)**: Let $(x(k), s(k))$, $k \geq 0$, be arbitrary. If $(x(k), s(k)) \in \mathscr{A}$, then it is obtained by letting $K_{qa} = k$ that $\Pr\big[(x(K_{qa}), s(K_{qa})) \in \mathscr{A} \mid (x(k), s(k))\big] = 1$. Otherwise (i.e., $(x(k), s(k)) \notin \mathscr{A}$), we consider respectively the two cases $m(k) = M(k)$ and $m(k) \neq M(k)$ as follows.

1) $m(k) = M(k)$. We have shown that $(\forall i \in \mathcal{V})\ x_i(k) \in \big[m(0), M(0)\big]$; so $m(k) = M(k) \in \big[m(0), M(0)\big]$. First consider the case $m(k) = M(k) \in \big[m(0), L\big]$. Choose a node i such that $x_i(0) = M(0)$; namely, node i has the maximal initial state. Then, by Lemma 5.3 we derive that there exists a finite time $K_0 > k$ such that

$$\Pr\big[x(K_0) = x(k),\ s_i(K_0) = s_1(k) + \cdots + s_n(k),$$
$$(\forall j \neq i)\ s_j(K_0) = 0 \mid (x(k), s(k))\big] > 0.$$

If $m(k) = M(k) = L$, then $m(K_0) = M(K_0) = L$ and thus $s(K_0)^T\mathbf{1} = R$. But $(\forall j \neq i)\ s_j(K_0) = 0$; hence $s_i(K_0) = R$, and consequently $(x(K_0), s(K_0)) \in \mathscr{A}$. Letting $K_{qa} = K_0$ we obtain the conclusion. Otherwise ($m(k) = M(k) = L - \alpha$ for some $\alpha \in [1, L - m(0)]$), we have $m(K_0) = M(K_0) = L - \alpha$ and $s_i(K_0) = R + \alpha n$. As $\mathcal{G}$ is strongly connected, there must exist another node $j \neq i$ with an edge $(j, i) \in \mathcal{E}$. Along this edge the following conditions hold:

$$\begin{aligned}
&x_i(K_0) = x_j(K_0) = L - \alpha,\\
&x_i(K_0) = L - \alpha < M(0) = x_i(0) = M_i(0) = M_i(K_0),\\
&s_i(K_0) + s_j(K_0) = R + \alpha n \geq n(= \delta).
\end{aligned}$$

When this edge is activated, **(R1)(i)** of **QA** algorithm applies:

$$\begin{aligned}
&x_i(K_0 + 1) = x_i(K_0) + 1 = L - \alpha + 1,\\
&s_i(K_0 + 1) = s_i(K_0) + s_j(K_0) - 1 = R + \alpha n - 1.
\end{aligned}$$

Now the conditions of Lemma 5.4 are met; we hence obtain that there exists a finite time $K_1 > K_0 + 1$ such that

$$\begin{aligned}
\Pr\big[&m(K_1) = M(K_1) = L - \alpha + 1,\ s_i(K_1) = R + (\alpha - 1)n,\\
&(\forall j \neq i)\ s_j(K_1) = 0 \mid (x(k), s(k))\big] > 0.
\end{aligned}$$

Repeating the above process, we derive a sequence of times $K_1 < K_2 < \cdots < K_\alpha$, and at the last time K_α,

$$\begin{aligned}
\Pr\big[&m(K_\alpha) = M(K_\alpha) = L,\ s_i(K_\alpha) = R,\\
&(\forall j \neq i)\ s_j(K_1) = 0 \mid (x(k), s(k))\big] > 0.
\end{aligned}$$

Set $K_{qa} = K_\alpha$ and **(C3)** holds. In the other case $m(k) = M(k) \in \big[L + 1, M(0)\big]$, **(C3)** similarly holds by a symmetric argument.

2) $m(k) \neq M(k)$. Write $x(k) = [x_1(k) \cdots x_n(k)]^T$ and fix a node $j \in \mathcal{V}$. Recall from Lemma 5.2 that under **QC** algorithm for general consensus, if the digraph $\mathcal{G}$ is strongly connected, then there exists a finite time $\bar{k} > k$ such that $\Pr\big[x(\bar{k}) =$

$x_j(k)\mathbf{1} \mid x(k)\big] > 0$. It is important to note that only **(R2)** and **(R3)** of **QC** algorithm are used in proving Lemma 5.2. But these two rules for the state updates are exactly the same in **QA** algorithm. Thus under **QA** algorithm, we derive that

$$\Pr\big[(x(\bar{k}), s(\bar{k}) = (x_j(k)\mathbf{1}, s(\bar{k})) \mid (x(k), s(k))\big] > 0.$$

Hence, $m(\bar{k}) = M(\bar{k}) = x_j(k) \in \big[m(0), M(0)\big]$ and the situation is that in 1), for which **(C3)** is established. ■

The key idea of the foregoing proof is to collect all the surpluses in the system at some agent. Then this agent can determine whether or not the overall surplus exceeds the threshold; if it does, indicating that the true average is not yet reached, this agent should proceed to update its state so that the extra surpluses may be distributed over the network. This process is repeated until the overall surplus falls below the threshold. This is, indeed, the primary reason which slows down the convergence rate of **QA** algorithm.

It is also worth pointing out that both the necessity and sufficiency proofs hold even if the surpluses, if any, are transmitted one unit at a time; namely, the transmitted surpluses may take values only from the set $\{-1, 0, 1\}$. In that case, when there is more than one-unit surplus to be passed from node j to i, we may consecutively select edge (j, i) for communication until all surpluses are transmitted. Such a selection, by our gossip setup, is with a positive probability. As a result, the transmission of surpluses requires merely two bits increase in communication.

Lastly, notice that the conditions **(C1)** and **(C2)** are established without any connectivity property of the digraph. Also, it follows from **(C2)** and **(C3)** that $\mathscr{A}$ is, indeed, the *unique* invariant set to which all trajectories converge.

5.5 Threshold and Local Extrema

In the present section we provide further analyses on the threshold and local extrema in **QA** algorithm. First, we find the range of threshold values which permits the agents to converge to the invariant set $\mathscr{A}$. Second, we demonstrate that for **QA**

algorithm the local extrema are necessary in order to keep the state set bounded.

5.5.1 Threshold Range

As we have seen in Section 5.4, the threshold value in **QA** algorithm serves as a bound such that whenever the surpluses exceed this bound, they should be distributed over the network. So far, we have assumed the threshold δ to be the total number n of agents in the network, and proved that all pairs of states and surpluses converge to the invariant set $\mathscr{A}$. Now we proceed to investigate the systemic behavior when $\delta \neq n$. In particular, we aim at finding the range of threshold values necessary and sufficient to ensure that $\mathscr{A}$ is the unique invariant set to which all trajectories converge. This investigation is important because if the threshold δ has to be exactly n in order to guarantee average consensus, then **QA** algorithm may not be *robust* in applications where some agents could fail and/or new agents could join.

We present the main result of this subsection: The range of suitable threshold values turns out to be $\big[\lfloor n/2 \rfloor + 1, n\big]$, which may be fairly large in practice.

Theorem 5.3. Suppose that the communication digraph $\mathcal{G}$ is strongly connected and **QA** algorithm is used. Then $\mathscr{A}$ is the unique invariant set to which all trajectories converge if and only if the threshold satisfies $\delta \in \big[\lfloor n/2 \rfloor + 1, n\big]$.

To prove Theorem 5.3 we need the following lemma. For a fixed $R \in [0, n-1]$, define $\mathcal{X}_R := \{x(0) : (\exists L)\ x(0)^T \mathbf{1} = nL + R\}$; thus $\mathcal{X}_R$ is the family of initial states whose sums, when divided by n, have remainder R for some quotient L. Clearly $\mathcal{X}_0, \ldots, \mathcal{X}_{n-1}$ form a partition of the set of all initial states.

Lemma 5.5. Under **QA** algorithm, fix $R \in [0, n-1]$.

(i) If the threshold satisfies $\delta \geq R + 1$, then $\mathscr{A}_L$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$;

(ii) if $\delta \geq n - R + 1$, then $\mathscr{A}_{L+1}$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$.

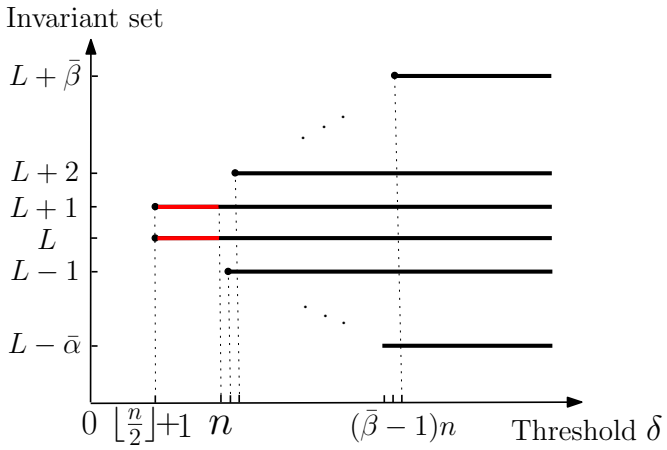


Figure 5.3: The relationship between threshold values and the invariant set ($\bar{\beta} > \bar{\alpha}$).

The proof is similar to that for **(C2)** in Theorem 5.2.

More generally, let $\bar{\alpha} := L - m(0)$ and $\bar{\beta} := M(0) - L$, where $m(0), M(0)$ are as in (5.4). For $\alpha \in [1, \bar{\alpha}]$, $\beta \in [2, \bar{\beta}]$, define the following subsets of $\mathbb{Z}^n \times \mathbb{Z}^n$:

$$\mathscr{A}_{L-\alpha} := \{(x, s) : x_i = L - \alpha \ \& \ s_i \geq 0, \ i = 1, ..., n\},$$
$$\mathscr{A}_{L+\beta} := \{(x, s) : x_i = L + \beta \ \& \ s_i \leq 0, \ i = 1, ..., n\}.$$

Similar to Lemma 5.5, we obtain for a fixed $R \in [0, n-1]$ that (i) if the threshold $\delta \geq \alpha n + R + 1$, then $\mathscr{A}_{L-\alpha}$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$; (ii) if $\delta \geq \beta n - R + 1$, then $\mathscr{A}_{L+\beta}$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$.

Remark 5.1. It is straightforward from the above derivation that the following hold:

(i) If the threshold $\delta \geq \alpha n + 1$, then $\mathscr{A}_{L-\alpha}$ is an invariant set for *some* pairs $(x(k), s(k))$;

(ii) if $\delta \geq (\beta - 1)n + 2$, then $\mathscr{A}_{L+\beta}$ is an invariant set for *some* pairs $(x(k), s(k))$.

Now we are ready to prove Theorem 5.3.

Proof of Theorem 5.3. (Necessity) Assume the threshold $\delta \notin \lfloor n/2 \rfloor + 1, n]$. First consider the case $\delta \leq \lfloor n/2 \rfloor$. By Lemma 5.5, neither $\mathscr{A}_L$ nor $\mathscr{A}_{L+1}$ is an invariant set at least for those pairs $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$ with $R = \lfloor n/2 \rfloor$. Namely, $\mathscr{A}$ is not an invariant set for all pairs $(x(k), s(k))$. For the other case $\delta \geq n + 1$, it follows

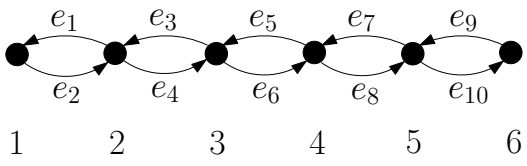


Figure 5.4: Without local extrema the states can grow arbitrarily large.

from Remark 5.1 (i) that at least $\mathscr{A}_{L-1}$ is an invariant set for some pairs $(x(k), s(k))$. Hence, $\mathscr{A}$ is not the unique one for all pairs $(x(k), s(k))$.

(Sufficiency) Let the threshold $\delta \in \big[\lfloor n/2 \rfloor + 1, n\big]$. Then, we derive by Lemma 5.5 that (i) $\mathscr{A}_L$ is an invariant set at least for those pairs $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0),\ R = 0, 1, ..., \lfloor n/2 \rfloor$; (ii) $\mathscr{A}_{L+1}$ is an invariant set at least for those pairs $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0),\ R = n-1, ..., n-\lfloor n/2 \rfloor$. But $\lfloor n/2 \rfloor = n - \lfloor n/2 \rfloor$ if n is even, or otherwise $\lfloor n/2 \rfloor + 1 = n - \lfloor n/2 \rfloor$. Consequently, $\mathscr{A}$ is an invariant set for all pairs $(x(k), s(k))$. In addition, similar to **(C3)** in the proof of Theorem 5.2 we can show that with a positive probability, every pair $(x(k), s(k)) \notin \mathscr{A}$ will enter $\mathscr{A}$ in finite time. Hence, there is no other invariant set, and $\mathscr{A}$ is the unique one to which all trajectories converge. ■

Summarizing the results in Theorem 5.3 and Remark 5.1, we conclude that for all pairs $(x(k), s(k))$, (i) when the threshold satisfies $\delta \in \big[0, \lfloor n/2 \rfloor\big]$, there is no invariant set; (ii) when $\delta \in \big[\lfloor n/2 \rfloor + 1, n\big]$, $\mathscr{A}$ is the unique invariant set; (iii) when $\delta \in \big[n+1, \infty\big)$, the invariant set expands as δ increases, but lower bounded by $L - \bar{\alpha}$ and upper bounded by $L + \bar{\beta}$. This relationship between threshold values and the invariant set is displayed in Fig. 5.3.

5.5.2 Role of Local Extrema

In **QA** algorithm, the local extrema m_i, M_i $(i \in \mathcal{V})$ are used to ensure that all the states $x_i(k)$, $k \geq 0$, remain within the interval of the initial states (i.e., $[m(0), M(0)]$). In this subsection, we provide an example which exhibits that without local extrema the states can grow arbitrarily large, thereby showing the necessity of using these variables in the algorithm.

Example 5.3. Consider six agents with the communication network in Fig. 5.4. Let the initial condition be as follows:

agent i	1	2	3	4	5	6
$x_i(0)$	10	10	10	0	0	0
$s_i(0)$	0	0	0	0	0	0

Suppose that **QA** algorithm is used, but without the conditions involving local extrema in **(R1)**. Also specify that $x_i(k+1) = \lceil (x_i(k) + x_j(k))/2 \rceil$ in **(R2)**, $x_i(k+1) = \lfloor (x_i(k) + x_j(k))/2 \rfloor$ in **(R3)**, and the threshold $\delta = 6$. Now consider the string of edges, $e_5^4\ e_3\ e_1\ (e_2\ e_1)^8$, being activated sequentially, and denote by T_1 $(= 22)$ the time after these activations. Then one may verify that

$x_i(T_1)$	13	13	0	0	0	0
$s_i(T_1)$	4	0	0	0	0	0

Thus the upper bound of the initial states, $M(0) = 10$, is exceeded by 3. Next, consider the string, $e_4^4\ e_6^4\ e_8\ e_{10}\ (e_9\ e_{10})^{17}$, and denote by T_2 $(= 66)$ the time after sequentially activating these edges. We then derive that

$x_i(T_2)$	13	13	13	13	-11	-11
$s_i(T_2)$	4	0	0	0	0	-4

Thus the initial lower upper bound, $m(0) = 0$, is exceeded by 11. As such, one may go on constructing similar strings of edges, and the states will grow arbitrarily large with a positive probability.

We have thus seen that in general the local extrema are necessary in order to keep the state set bounded. Only in a special case where the threshold δ equals exactly n, however, we find it is possible to avoid using local extrema by suitably modifying **QA** algorithm. We discuss this modification in the next section.

5.6 Modified Averaging Algorithm for Reduced Computation

We present the *modified quantized averaging* (**MQA**) algorithm, discuss its properties, and show that it works without local extrema but only when the threshold δ equals n. In particular, we shall emphasize that **MQA** algorithm features in reducing both computation and communication burden of individual agents.

Now suppose that an edge (j, i) is randomly activated at time k. For the sender j, the updating rules are as before: $x_j(k+1) = x_j(k),\ s_j(k+1) = 0$. For the receiver i, on the other hand, the rules are modified as follows:

(R1) If $x_i(k) = x_j(k)$, then there are two cases:

(i) If $s_i(k) + s_j(k) \geq \delta$, then

$$x_i(k+1) = x_i(k) + 1, \quad s_i(k+1) = s_i(k) + s_j(k) - 1.$$

(ii) Otherwise (i.e., $s_i(k) + s_j(k) < \delta$),

$$x_i(k+1) = x_i(k), \quad s_i(k+1) = s_i(k) + s_j(k).$$

(R2) If $x_i(k) < x_j(k)$, there are also two cases:

(i) If $s_i(k) + s_j(k) > 0$, then

$$x_i(k+1) = x_i(k) + \Delta(k), \text{ where } \Delta(k) \in [1, \min\{s_i(k) + s_j(k), x_j(k)\}],$$
$$s_i(k+1) = s_i(k) + s_j(k) - \Delta(k).$$

(ii) Otherwise (i.e., $s_i(k) + s_j(k) = 0$),

$$x_i(k+1) = x_i(k), \quad s_i(k+1) = s_i(k) + s_j(k).$$

(R3) If $x_i(k) > x_j(k)$, then

$$x_i(k+1) \in [x_j(k), x_i(k)), \quad s_i(k+1) = s_i(k) + s_j(k) - \big(x_i(k+1) - x_i(k)\big).$$

First, some immediate observations are in sequel. (i) Surplus variables are updated such that the quantity $(x+s)^T\mathbf{1}$ stays invariant at each iteration, as in **QA**. (ii) Surpluses are decreased in **(R1)** and **(R2)**, and increased in **(R3)**; it is thus easy to

see that all surplus variables are nonnegative. By contrast, in **QA**, surplus variables can be either positive or negative. As a result, **(R1)** here has only two cases dealing with nonnegative surpluses. (iii) Updated by **MQA** the states indeed cannot reach consensus at value $L+1$; since otherwise $x=(L+1)\mathbf{1}$, there holds $s^T\mathbf{1}=R-n<0$, which contradicts that all surpluses are nonnegative. So the set of average consensus states in this case is only $\mathscr{A}_L$.

Second, we emphasize the distinctions between **MQA** and **QA**, which may imply tradeoffs when employing one or the other in practice. (i) In **QA** each node is assigned two local extrema variables, which prevent the states from exceeding the interval of all initial states, thereby ensuring finite state set. By contrast, it will be shown for **MQA** that the state set is guaranteed to be finite without the local extrema; consequently, computation effort for a total of $2n$ variables is saved. (ii) For **QA** we discussed that converging to the average is not affected even if the surpluses are transmitted one unit at a time, thereby requiring two bits for each transmission. The same is true for **MQA**; however, since there is no negative surplus, each transmission reduces to one bit for only $\{0,1\}$. (iii) A drawback of **MQA** lies in that the threshold value δ has to be exactly n in order to guarantee that $\mathscr{A}_L$ is the unique equilibria set for all initial conditions (cf. Lemma 5.5 **(i)**). For **QA**, on the other hand, the threshold can take values in a fairly large range.

The following is the convergence result of **MQA** algorithm.

Theorem 5.4. Under **MQA** algorithm, a network of agents achieves quantized averaging at $\mathscr{A}_L$ almost surely if and only if the digraph $\mathcal{G}$ is strongly connected.

Proof. The proof is analogous to that of Theorem 5.2, except the justification for the finite state set with no local extrema involved. To this end, for an arbitrary state $x(k)$ define $m(k), M(k)$ as in (5.4), and let $S=x(0)^T\mathbf{1}$. Since no negative surplus can be generated in **MQA** algorithm, the minimum $m(k)$ is non-decreasing. Hence, an upper bound for $M(k)$ is $S-(n-1)m(0)$, when there is one agent having this value, all other $n-1$ agents having $m(0)$, and all surpluses are 0. It then follows that a trivial upper bound for the state set is $(S-nm(0)+1)^n$. In addition, for surpluses

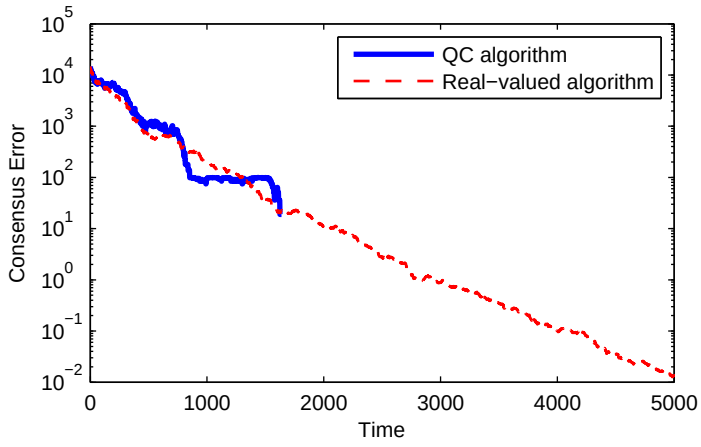


Figure 5.5: Decay of consensus error in **QC** and real-valued consensus algorithms

we derive that for every $i \in [1, n]$ and $k \geq 0$, $s_i(k) \in \big[0, S - nm(0)\big]$. Thus, a trivial upper bound for the surplus set is also $(S - nm(0) + 1)^n$. ■

5.7 Numerical Examples

In this section we provide a set of numerical examples to illustrate our theoretic developments, with special emphasis on algorithm convergence time.

5.7.1 QC and Real-Valued Consensus Algorithms

We compare the convergence rate of **QC** algorithm with that of real-valued consensus algorithms [29]. For this we consider a cyclic digraph of 20 agents (cf. Fig. 5.1(a)), whose initial (integer) states are chosen uniformly at random from the interval $[-10, 10]$. For **QC** algorithm we specify that $x_i(k+1) = \lceil (x_i(k) + x_j(k))/2 \rceil$ in **(R2)** and $x_i(k+1) = \lfloor (x_i(k) + x_j(k))/2 \rfloor$ in **(R3)**; for the real-valued algorithm let $x_i(k+1) = (x_i(k) + x_j(k))/2$ in all cases. Now define the consensus error $e := \sum_{i,j \in \mathcal{V}, i<j} (x_i - x_j)^2$; we compare the decay rates of this error in both algorithms. Two curves showing the decay trajectories are displayed in Fig. 5.5, which are the average of 100 runs of the respective algorithms. Observe that while real-valued algorithm converges asymptotically, **QC** algorithm converges in finite time. Prior to the finite convergence, the two error decay rates are indeed analogous; this observation supports our conjecture on

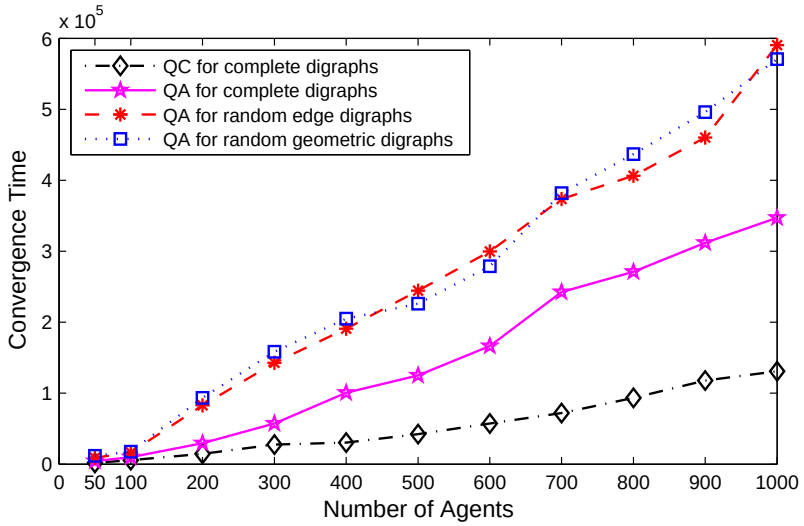


Figure 5.6: Convergence time versus number of agents

the convergence time of **QC** algorithm in Section 5.3.

5.7.2 Convergence Time versus Number of Agents

We turn next to the study of convergence time with respect to the number of agents in the network. The states of the agents are randomly initialized from a uniform distribution on the interval $[-5, 5]$.

First, we deal with the increasing rates of convergence time as the number of agents increases for both **QC** and **QA** algorithms on complete digraphs (i.e., every agent is reachable from every other agent via a directed edge). The results are respectively the dash-dot and solid curves in Fig. 5.6, each plotted value being the average convergence time of 100 runs of the corresponding algorithms. It is seen that the convergence time of **QA** algorithm is longer than that of **QC** algorithm, which supports our assertion in Section 5.4 that the additional averaging process required in **QA** algorithm slows down its convergence.

Second, we do an analogous investigation for **QA** algorithm on two types of random digraphs. One type, referred to as random edge digraphs, is defined as follows (e.g., [39]): The existence of a directed edge between every pair of agents is determined randomly, independent of other edges, with a (possibly non-uniform) positive

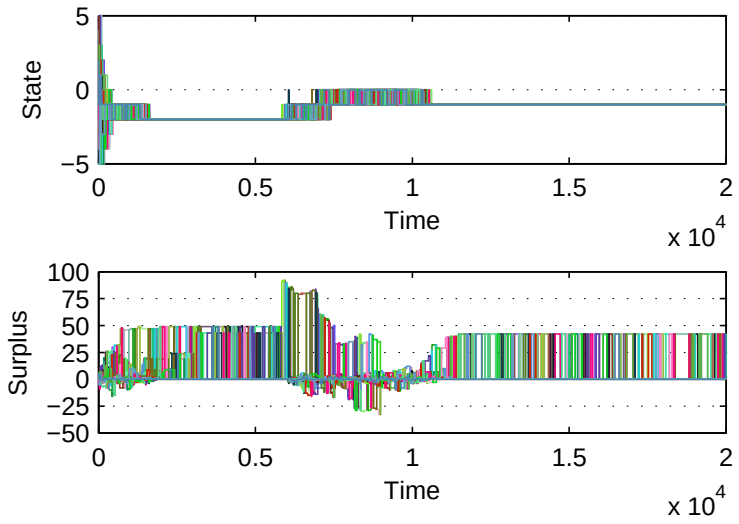


Figure 5.7: Convergence sample path of 50 agents on random edge digraphs

probability. Hence in expectation, we obtain complete digraphs. Here for simplicity, we assume that every edge exists with the same probability p. The other type is the random geometric digraphs (e.g., [38]), which have been widely used for modeling *ad hoc* wireless sensor networks. In two dimensions, a random geometric digraph $\mathcal{G}(n, r)$ denotes a network of n agents whose transmission radius is within r. It is obtained by placing n agents uniformly at random in a unit square, and connecting every pair of agents to each other that are within distance r.

In Fig. 5.6, the dashed and dotted curves show the average convergence time of 100 runs of **QA** algorithm on random edge digraphs with $p = 0.6$ and random geometric digraphs with $r = 0.5$, respectively. We see that as the network expands, the increasing rates of convergence time in these two cases are roughly of the same polynomial order; this indicates that the graph connectivity resulted from the chosen parameters might be similar.

In addition, the convergence time of **QA** algorithm is longer on random digraphs than on complete digraphs. This is due evidently to the parameter choices, for complete digraphs can be viewed as special random digraphs by setting $p = 1$ or $r = \sqrt{2}$. To further illustrate this point, we display the convergence sample paths of random edge and complete digraphs for 50 agents, corresponding to the first plotted value in

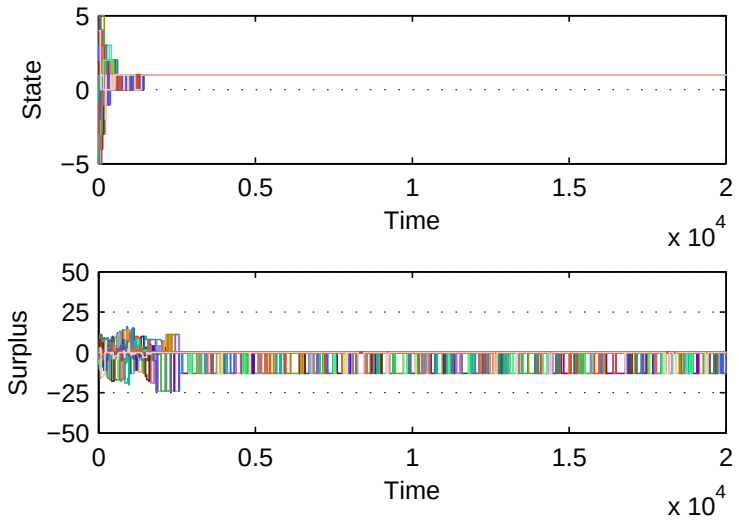


Figure 5.8: Convergence sample path of 50 agents on complete digraphs

Fig. 5.6. For random edge digraphs, we exhibit in Fig. 5.7 the case where the initial state sum is $\sum_{i=1}^{50} x_i(0) = -8$, hence the true average being either -1 or 0. The trajectories show that the states converge to -1, and the corresponding total surplus settles at 42. Note that the convergence time of this sample path is 1.1×10^4; for 100 runs of **QA** algorithm we obtain the average convergence time 9.0×10^3.

For complete digraphs, Fig. 5.8 displays the example where the initial state sum is $\sum_{i=1}^{50} x_i(0) = 37$; the true average is thus either 0 or 1. We see that all states converge to 1, with the steady state surplus being -13. This convergence takes only 2.6×10^3 time steps; also the average of 100 runs of **QA** algorithm is merely 4.2×10^3. Thus larger value of the parameter p gives rise to higher graph connectivity, and therefore accelerates the convergence speed.

5.7.3 Convergence Time versus Threshold Value

In Section 5.5, we have justified that the threshold in **QA** algorithm can take values in the range $\lceil \lfloor n/2 \rfloor + 1, n \rceil$ so as to guarantee convergence to the average consensus set $\mathscr{A}$. Here we provide an example to show the impact of different threshold values (in the valid range) on the convergence time of **QA** algorithm. Consider a complete digraph of 50 agents, with random initial states in $[-5, 5]$. In Fig. 5.9 we plot the

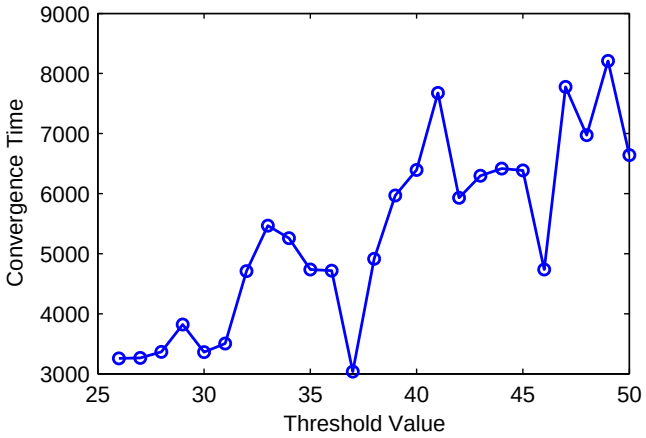


Figure 5.9: Impact of threshold values on convergence time of **QA** algorithm

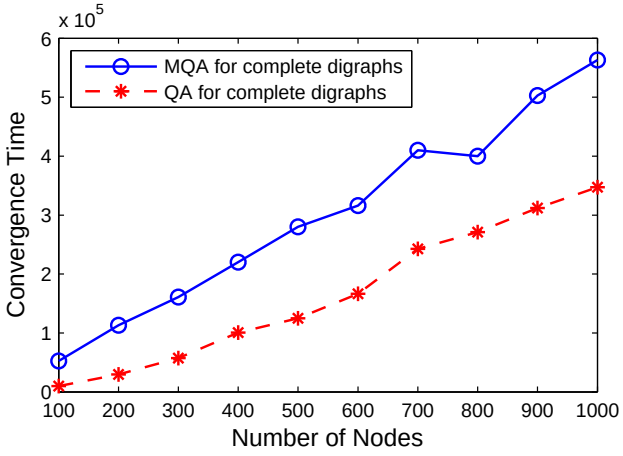


Figure 5.10: Convergence time comparison between **QA** and **MQA** algorithms on complete digraphs.

average convergence time over 100 runs of **QA** algorithm, for each valid threshold value ranging from 26 to 50. We can observe an increasing trend of convergence time as the threshold value increases. This is mainly because with a smaller threshold, the decision on distributing surpluses over the network can be made potentially faster, hence accelerating the averaging process.

5.7.4 QA versus MQA

Finally, we compare the increasing rates of convergence time as the number of nodes increases between **QA** and **MQA** on complete digraphs. Initial states are drawn from

the interval $[-5, 5]$ uniformly at random. The results are respectively the solid and dashed curves in Fig. 5.10, each plotted value being the average convergence time of 100 runs of the corresponding algorithms. Observe that **MQA** converges more slowly than **MQA**; this indicates the following tradeoff: The benefit of **MQA** in reducing computation and communication effort is at the cost of increasing convergence time.

5.8 Proofs

Proof of Lemma 5.3. Fix a node in $\mathcal{V}$ and denote it by i_0. As $\mathcal{G}$ is strongly connected, for each $i \neq i_0$ there is a directed path from i to i_0. The *length* of a path is defined to be the number of its edges. Now let l_{i,i_0} be the *minimal* length of all the paths from i to i_0. Partition the set $\mathcal{V}$ of nodes into $\{\mathcal{V}_0, \mathcal{V}_1, ..., \mathcal{V}_r\}$, for some $r \in [1, n-1]$, with

$$\mathcal{V}_0 = \{i_0\}, \quad \mathcal{V}_h = \{i \in \mathcal{V} : l_{i,i_0} = h\},\ h = 1, ..., r.$$

It is evident that there always exists r such that $\mathcal{V}_0, ..., \mathcal{V}_r$ are nonempty, disjoint, and $\mathcal{V}_0 \cup \cdots \cup \mathcal{V}_r = \mathcal{V}$. In the following we describe the sequence of activating edges which causes all surpluses in the system to pile up at i_0 in finite time, the idea being visualized in Fig. 5.11. Owing to that each edge in $\mathcal{E}$ has a positive probability to be activated, this sequence of activation also enjoys a positive probability. We now proceed by induction.

First, take an arbitrary node $i_1 \in \mathcal{V}_1$ and activate edge (i_1, i_0). By assumption $x_{i_0}(k) = x_{i_1}(k)$; thus only **(R1)** of **QA** algorithm applies. If it is the case **(R1)(iii)**, then

$$x(k+1) = x(k),\ s_{i_1}(k+1) = 0,\ s_{i_0}(k+1) = s_{i_0}(k) + s_{i_1}(k). \tag{5.5}$$

Otherwise (i.e., the case **(R1)(i)/(ii)**),

$$x_{i_0}(k+1) = x_{i_0}(k) \pm 1,\ s_{i_0}(k+1) = s_{i_0}(k) + s_{i_1}(k) \mp 1,$$
$$x_{i_1}(k+1) = x_{i_1}(k),\ s_{i_1}(k+1) = 0;$$

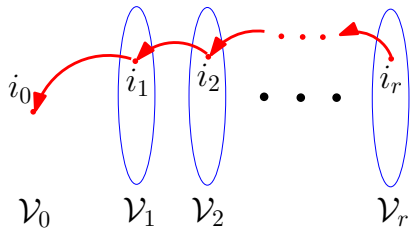


Figure 5.11: The idea of the proof for Lemma 5.3

in either case, activate edge (i_1, i_0) again. This time **(R3)/(R2)** of **QA** algorithm applies, yielding

$$x_{i_0}(k+2) = x_{i_0}(k+1) \mp 1,\ s_{i_0}(k+2) = s_{i_0}(k+1) + s_{i_1}(k+1) \pm 1,$$
$$x_{i_1}(k+2) = x_{i_1}(k+1),\ s_{i_1}(k+2) = 0.$$

Hence

$$x(k+2) = x(k),\ s_{i_1}(k+2) = 0,\ s_{i_0}(k+2) = s_{i_0}(k) + s_{i_1}(k). \tag{5.6}$$

We see in (5.5) and (5.6) that the state is the same and the surplus of i_1 comes to i_0. Repeating the foregoing process for every other node in $\mathcal{V}_1$, we derive that there must exist a finite time $k_1 > k$ such that

$$x(k_1) = x(k),\ (\forall i_1 \in \mathcal{V}_1)\ s_{i_1}(k_1) = 0,\ s_{i_0}(k_1) = s_{i_0}(k) + \sum_{i_1 \in \mathcal{V}_1} s_{i_1}(k).$$

Now suppose that there is a finite time $k_{j-1} > \cdots > k_1$ (for some $j \in [2, r]$) such that $x(k_{j-1}) = x(k)$,

$$(\forall h \in [1, j-1])(\forall i_h \in \mathcal{V}_h)\ s_{i_h}(k_{j-1}) = 0,\ s_{i_0}(k_{j-1}) = s_{i_0}(k) + \sum_{h=1}^{j-1} \sum_{i_h \in \mathcal{V}_h} s_{i_h}(k).$$

Let $i_j \in \mathcal{V}_j$. Then there must exist a directed path from i_j to i_0: $(i_j, i_{j-1}) \cdots (i_2, i_1)(i_1, i_0)$ for some $i_h \in \mathcal{V}_h$ $(h = 1, ..., j-1)$. First activate edge (i_j, i_{j-1}). By hypothesis $x_{i_{j-1}}(k_{j-1}) = x_{i_j}(k_{j-1})$; thus only **(R1)** of **QA** algorithm applies. The present situation is the same as that in the base case – if it is **(R1)(iii)**, no further activation takes place; otherwise, activate edge (i_j, i_{j-1}) once more. As in (5.5) and (5.6) we obtain that there is $\tau_1 > k_{j-1}$ such that

$$x(\tau_1) = x(k_{j-1}),\ s_{i_j}(\tau_1) = 0,\ s_{i_{j-1}}(\tau_1) = s_{i_{j-1}}(k_{j-1}) + s_{i_j}(k_{j-1}).$$

Now sequentially for the edges $(i_{j-1}, i_{j-2}) \cdots (i_1, i_0)$, there is a sequence of times $\tau_1 < \tau_2 < \cdots < \tau_j$ such that

$$\begin{aligned} &x(\tau_2) = x(\tau_1),\ s_{i_{j-1}}(\tau_2) = 0,\ s_{i_{j-2}}(\tau_2) = s_{i_{j-2}}(\tau_1) + s_{i_{j-1}}(\tau_1); \\ &\qquad \vdots \\ &x(\tau_j) = x(\tau_{j-1}),\ s_{i_1}(\tau_j) = 0,\ s_{i_0}(\tau_j) = s_{i_0}(\tau_{j-1}) + s_{i_1}(\tau_{j-1}). \end{aligned}$$

From these derivations and the hypothesis, it follows that

$$\begin{aligned} &x(\tau_j) = x(k),\ s_{i_j}(\tau_j) = 0,\ (\forall h \in [1, j-1])(\forall i_h \in \mathcal{V}_h)\ s_{i_h}(\tau_j) = 0, \\ &s_{i_0}(\tau_j) = s_{i_0}(k) + \sum_{h=1}^{j-1} \sum_{i_h \in \mathcal{V}_h} s_{i_h}(k) + s_{i_j}(k). \end{aligned}$$

Hence, at time τ_j, the state is the same and the surplus of i_j comes to i_0. Repeating the same process for every other node in $\mathcal{V}_j$, we derive that there must exist a finite time $k_j > k_{j-1}$ such that $x(k_j) = x(k)$,

$$(\forall h \in [1, j])(\forall i_h \in \mathcal{V}_h)\ s_{i_h}(k_j) = 0,\ s_{i_0}(k_j) = s_{i_0}(k) + \sum_{h=1}^{j} \sum_{i_h \in \mathcal{V}_h} s_{i_h}(k).$$

This completes the induction step. The conclusion follows by letting $j = r$. ■

Proof of Lemma 5.4. First, for $\tilde{k} \geq k$ define two subsets of nodes with states $L - \alpha$ and $L - \alpha + 1$, respectively, by $\mathcal{V}_1(\tilde{k}) := \{i \in \mathcal{V} : x_i(\tilde{k}) = L - \alpha\}$ and $\mathcal{V}_2(\tilde{k}) := \{i \in \mathcal{V} : x_i(\tilde{k}) = L - \alpha + 1\}$. Let their cardinalities be $n_1(\tilde{k}) := |\mathcal{V}_1(\tilde{k})|$ and $n_2(\tilde{k}) := |\mathcal{V}_2(\tilde{k})|$. Denote by i_0 the node that has state $L - \alpha + 1$ and surplus $R + \alpha n - 1$ at time k. By assumption,

$$\begin{aligned} &\mathcal{V}_2(k) = \{i_0\},\ \mathcal{V}_1(k) = \mathcal{V} - \mathcal{V}_2(k);\ n_2(k) = 1,\ n_1(k) = n - 1; \\ &s_{i_0}(k) = R + \alpha n - 1,\ (\forall i \neq i_0)\ s_i(k) = 0. \end{aligned} \tag{5.7}$$

In the following we show that there is a positive probability such that all nodes in $\mathcal{V}_1$ will enter $\mathcal{V}_2$ one by one in finite time; we proceed by induction.

Consider the base case in (5.7). Since $\mathcal{G}$ is strongly connected, there must exist a directed edge (i_0, i_1) for some $i_1 \in \mathcal{V}_1(k)$. If this edge is activated, **(R2)** of **QA**

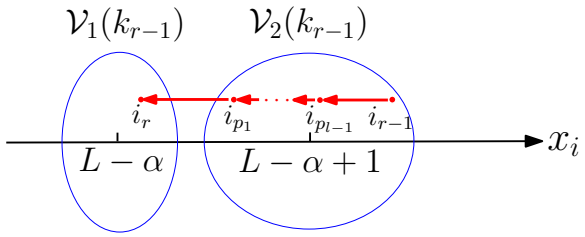


Figure 5.12: The idea of the proof for Lemma 5.4

algorithm will apply because $x_{i_1}(k) < x_{i_0}(k)$. In that case,

$$x_{i_1}(k+1) = L-\alpha+1,\ s_{i_1}(k+1) = s_{i_0}(k) + s_{i_1}(k) - 1 = R+\alpha n-2,$$
$$x_{i_0}(k+1) = x_{i_0}(k),\ s_{i_0}(k+1) = 0.$$

Hence, the following hold at time $k_1 := k+1$ with a positive probability:

$$n_1(k_1) = n-2,\ n_2(k_1) = 2,\ s_{i_1}(k_1) = R+\alpha n-2,\ (\forall i \neq i_1)\ s_i(k_1) = 0,$$

where $i_1 \in \mathcal{V}_2(k_1) \cap \mathcal{V}_1(k)$. That is, node i_1 enters $\mathcal{V}_2$, and holds all the surpluses.

Now suppose that there is a positive probability such that, at time $k_{r-1} > \cdots > k_1$ (for some $r \in [2,n]$),

$$n_1(k_{r-1}) = n-r,\ n_2(k_{r-1}) = r,\ s_{i_{r-1}}(k_{r-1}) = R+\alpha n-r,\ (\forall i \neq i_{r-1})\ s_i(k_{r-1}) = 0,$$

where $i_{r-1} \in \mathcal{V}_2(k_{r-1}) \cap \mathcal{V}_1(k_{r-2})$. For i_{r-1} choose a node $i_r \in \mathcal{V}_1(k_{r-1})$ such that the directed path from i_{r-1} to i_r is one of the shortest from the node i_{r-1} to the set $\mathcal{V}_1(k_{r-1})$ (see Fig. 5.12). Let l be the corresponding length, and denote this path by $(i_{r-1}, i_{p_{l-1}}) \cdots (i_{p_2}, i_{p_1})(i_{p_1}, i_r)$. Notice that the nodes $i_{p_1}, i_{p_2}, ..., i_{p_{l-1}}$ are all in $\mathcal{V}_2(k_{r-1})$, because otherwise this path is not one of the shortest from i_{r-1} to $\mathcal{V}_1(k_{r-1})$. Hence, for the path $(i_{r-1}, i_{p_{l-1}}) \cdots (i_{p_2}, i_{p_1})$ Lemma 5.3 applies, by which all the states of these nodes remain the same and all the surpluses (currently held by i_{r-1}) may pile up at any chosen node. Here we choose this node to be i_{p_1}, and obtain that there is a positive probability such that at time $k'_{r-1} > k_{r-1}$,

$$x_{i_{p_1}}(k'_{r-1}) = \cdots = x_{i_{r-1}}(k'_{r-1}) = L-\alpha+1,\ s_{i_{p_2}}(k'_{r-1}) = \cdots = s_{i_{r-1}}(k'_{r-1}) = 0,$$
$$s_{i_{p_1}}(k'_{r-1}) = s_{i_{p_1}}(k_{r-1}) + \cdots + s_{i_{r-1}}(k_{r-1}) = R+\alpha n-r.$$

Subsequently we activate edge (i_{p_1}, i_r); since $x_{i_r}(k'_{r-1}) < x_{i_{p_1}}(k'_{r-1})$, **(R2)** of **QA** algorithm applies:

$$\begin{aligned}
&x_{i_r}(k'_{r-1}+1) = L-\alpha+1,\\
&s_{i_r}(k'_{r-1}+1) = s_{i_{p_1}}(k'_{r-1}) + s_{i_r}(k'_{r-1}) - 1 = R+\alpha n - r - 1,\\
&x_{i_{p_1}}(k'_{r-1}+1) = x_{i_{p_1}}(k'_{r-1}),\ s_{i_{p_1}}(k'_{r-1}+1) = 0.
\end{aligned}$$

Hence, the following hold at time $k_r := k'_{r-1}+1$ with a positive probability:

$$n_1(k_r) = n-r-1,\ n_2(k_r) = r+1;\ s_{i_r}(k_r) = R+\alpha n - r - 1,\ (\forall i \neq i_r)\ s_i(k_r) = 0,$$

where $i_r \in \mathcal{V}_2(k_r) \cap \mathcal{V}_1(k_{r-1})$. This establishes the induction. Letting $r = n$ we derive that at time k_{n-1} and with a positive probability, all nodes have state $L-\alpha+1$, i.e.,

$$m(k_{n-1}) = M(k_{n-1}) = L-\alpha+1;$$

the node i_{n-1}, which enters $\mathcal{V}_2$ lastly, holds all the surpluses, i.e.,

$$s_{i_{n-1}}(k_{n-1}) = R+(\alpha-1)n,\ (\forall i \neq i_{n-1})\ s_i(k_{n-1}) = 0.$$

Finally, we invoke again Lemma 5.3 to collect all the surpluses in the system (currently held by i_{n-1}) at node i_0, and the conclusion ensues. ■

Chapter 6

Convergence Time of Quantized Gossip Algorithms

6.1 Introduction

We have designed gossip-type algorithms which solve quantized consensus and averaging problems on digraphs with the least restrictive connectivity requirements. In the present chapter, we investigate the performance of these algorithms by providing upper bounds on their *mean convergence time*. The state transition structures resulting from these algorithms turn out to be rather complicated. Hence in our analysis on convergence time, we focus on the special case of complete graphs. The analysis is still challenging, but we will also discuss that the general approach can be useful for other graph topologies. First, for **QC** algorithm, we find that the mean convergence time is $O(n^2)$. To derive this bound, we view reaching consensus as the smallest interval containing all states shrinking its length to zero. This perspective leads us to characterizing convergence time by the *hitting* time in a certain Markov chain, which yields the polynomial bound. Second, we obtain that the mean convergence time of **QA** algorithm is $O(n^3)$. As the original algorithm in Chapter 5 is found to induce complex state transition structures, we have suitably revised it to manage the complexity. For the modified algorithm, a Lyapunov function is proposed which measures

the distance from the average value. We then bound convergence time by way of bounding the number of iterations required to decrease the Lyapunov function; the latter is again characterized by the hitting time in a special Markov chain.

Our work is related to [32,45,48,81], which deal also with the convergence time of gossip averaging algorithms with quantized states. In [45], a Lyapunov approach is adopted and polynomial bounds on convergence time are obtained for fully connected and linear networks. The work [81] generalizes these bounds to arbitrarily connected networks (fixed or switching), utilizing the results on the meeting time of two random walks on graphs. Also, bounds for arbitrarily connected networks are provided in [32,48]; these bounds are, however, in terms of graph topology rather than the number of nodes. In these cited references, a common feature is that the graphs are undirected. By contrast, our averaging algorithm is designed for arbitrary strongly connected digraphs, and we are interested in studying the corresponding convergence time.

To bound the convergence time, a frequently employed approach is to bound the decay time of some suitable Lyapunov functions [45, 57]. In particular, [57] derives tight polynomial bounds on the convergence time of synchronized averaging algorithms, with either real or quantized states. In addition, [12,79] investigate the fastest averaging algorithms by optimizing the updating weights. This optimization is shown to be a semidefinite program in case of symmetric weights. In addition, [22] investigates a variety of quantization effects on averaging algorithms, and demonstrate favorable convergence properties by simulations. Our work adopts the Lyapunov method, as in [45, 57]; the common function used in these papers turns out, however, not to be a valid Lyapunov function for our averaging algorithm. This is due again to that the state sum does not remain invariant, and the augmented surplus evolution must also be taken into account. According to these features, we establish an appropriate Lyapunov function, and prove that bounding its decay time can be reduced to finding the hitting time in a certain Markov chain.

For the convergence time analysis below, we will impose the following two assumptions on the graph topology and the probability distribution of activating edges.

Assumption 6.1. The digraph $\mathcal{G}$ is *complete* (i.e., every node is connected to every other node by a directed edge). It follows that there are $\text{card}(\mathcal{E}) = n(n-1)$ edges.

Assumption 6.2. The probability distribution on edge activation is *uniform*; namely, each edge can be activated with the same probability $p := 1/\text{card}(\mathcal{E})$.

Also, it is convenient to specify a finite interval as follows:

$$\mathcal{X} := \{x : m \leq x_i \leq M,\ i \in \mathcal{V}\}, \tag{6.1}$$

where $m, M \in \mathbb{Z}$ are some finite constants. Suppose throughout this chapter that the initial state vector $x(0)$ satisfies $x(0) \in \mathcal{X}$.

The rest of this chapter is organized as follows. First in Section 6.2, we formulate and solve the problem of convergence time analysis for **QC** algorithm. Then in Sections 6.3 and 6.4, we derive an upper bound for the convergence time of **QA** algorithm. Further, we demonstrate the derived time bounds by simulation in Section 6.5. The material in this chapter has appeared in [14].

6.2 Convergence Time Analysis of Quantized Consensus Algorithm

6.2.1 Problem Formulation and Result Statement

Recall from Section 5.3 the **QC** algorithm, and from (5.1) that $\mathscr{C}$ is the set of general consensus states. The convergence time of **QC** algorithm is the random variable T_{qc} defined by $T_{qc} := \inf\{k \geq 0 : x(k) \in \mathscr{C}\}$. The mean convergence time (with respect to the probability distribution on edge activation), starting from a state $x_0 \in \mathcal{X}$ in (6.1), is then given by

$$E_{qc}(x_0) := E\left[T_{qc} | x(0) = x_0\right]. \tag{6.2}$$

Problem 6.1. Let Assumptions 1-2 hold. Find an upper bound of the mean convergence time $E_{qc}(x_0)$ of **QC** algorithm with respect to all possible initial states $x_0 \in \mathcal{X}$.

We now present the main result of this section: an upper bound of the mean convergence time $E_{qc}(x_0)$ for all possible initial states $x_0 \in \mathcal{X}$.

Theorem 6.1. Let Assumptions 1 and 2 hold. Then

$$\max_{x_0 \in \mathcal{X}} E_{qc}(x_0) < n(n-1)(M-m) = O(n^2).$$

To derive this bound, we first provide preliminaries on the hitting time in finite Markov chains.

6.2.2 Preliminaries on Hitting Time

Let $\{X_k\}_{k\geq 0}$ be a Markov chain with a finite state space $\mathcal{S}$ and a transition probability matrix $P = (P_{ij})$ (e.g., [58]). The entry P_{ij} denotes the one-step transition probability from state i to state j. In particular, the diagonal entry P_{ii} denotes the *selfloop* transition probability. A state $i \in \mathcal{S}$ is said to be *absorbing* if $P_{ii} = 1$. For a given $\{X_k\}_{k\geq 0}$, the *hitting time* of a subset $\mathcal{T}$ of $\mathcal{S}$ is the random variable $H_{\mathcal{T}}(\{X_k\}_{k\geq 0})$ defined by

$$H_{\mathcal{T}}(\{X_k\}_{k\geq 0}) := \inf\{l \geq 0 : X_l \in \mathcal{T}\}.$$

The mean time (with respect to the probability distribution specified by P) taken for the chain, starting from a state $i \in \mathcal{S}$, to hit $\mathcal{T}$ is given by

$$E_i := E\left[H_{\mathcal{T}}(\{X_k\}_{k\geq 0}) \,|X_0 = i\right] = \sum_{l=0}^{\infty} l \cdot \Pr\left[H_{\mathcal{T}}(\{X_k\}_{k\geq 0}) = l | X_0 = i\right], \qquad (6.3)$$

where $E[\cdot|\cdot]$ and $\Pr[\cdot|\cdot]$ denote the conditional expectation and conditional probability operators, respectively. Here is an important fact on mean hitting times [58, Theorem 1.3.5].

Lemma 6.1. The vector of mean hitting times $(E_i)_{i\in\mathcal{S}}$ of a subset $\mathcal{T}$ satisfies the system of linear equations

$$\begin{cases} E_i = 0 & \text{for } i \in \mathcal{T}, \\ E_i = \sum_{j \notin \mathcal{T}} P_{ij} E_j + 1 & \text{for } i \notin \mathcal{T}. \end{cases}$$

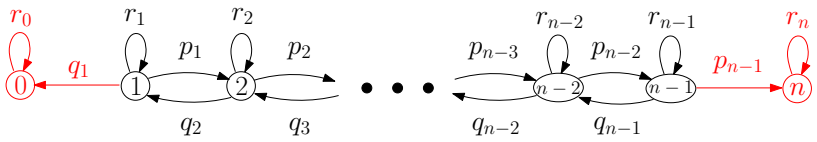


Figure 6.1: Markov chain I: states 0 and n are absorbing. Here $r_0, \ldots, r_n$ are selfloop transition probabilities.

Using Lemma 6.1, we derive a closed-form expression of the mean hitting time for a specific Markov chain; this chain will be shown to characterize the state transition structure under **QC** algorithm. For the proof of this result, see Section 6.6.

Lemma 6.2. Consider the Markov chain in Fig. 6.1 with transition probabilities

$$p_z + r_z + q_z = 1,\ p_z = q_z\ (z = 1, ..., n-1), \qquad r_0 = 1, \qquad r_n = 1.$$

Then the mean hitting time of the state 0 or n starting from state z is

$$E_z = (1 - \frac{z}{n}) \sum_{i=1}^{z-1} \frac{i}{p_i} + \frac{z}{n} \sum_{j=z}^{n-1} \frac{n-j}{p_j}\ (z = 1, ..., n-1).$$

6.2.3 Analysis of Convergence Time

We now proceed as follows. Recall from (5.4) the minimum state $m(k)$ and the maximum state $M(k)$. We view the state $x(k)$ converging to $\mathscr{C}$ as the interval $[m(k), M(k)]$ shrinking to length 0. Let the random variable T_{qc}^1 be the time when one interval shrinkage occurs; then the corresponding mean time, starting from a state x, is $E_{qc}^1(x) := E\left[T_{qc}^1 | x \in \mathcal{X}\right]$. Since one shrinkage decreases the interval length by at least 1, there can be at most $M - m$ shrinkages for $x_0 \in \mathcal{X}$. It then follows that

$$\max_{x_0 \in \mathcal{X}} E_{qc}(x_0) \leq \max_{x \in \mathcal{X}} E_{qc}^1(x) \cdot (M - m). \tag{6.4}$$

Consider a subset $\mathcal{X}_1$ of $\mathcal{X}$ defined by

$$\mathcal{X}_1 := \{x : x_1 = \cdots = x_z = 1\ \&\ x_{z+1} = \cdots = x_n = 0,\ z \in [1, n-1]\}. \tag{6.5}$$

Thus the interval has length 1 for all $x \in \mathcal{X}_1$. It is easy to see that $\max_{x_0 \in \mathcal{X}_1} E_{qc}(x_0) = \max_{x \in \mathcal{X}} E_{qc}^1(x)$. The following lemma states an upper bound of $E_{qc}(x_0)$ for $x_0 \in \mathcal{X}_1$.

Lemma 6.3. Let Assumptions 1 and 2 hold. Then

$$\max_{x_0 \in \mathcal{X}_1} E_{qc}(x_0) < n(n-1) = O(n^2).$$

Proof. By Assumptions 1 and 2, every directed edge in $\mathcal{G}$ can be activated with the uniform probability $p = 1/(n(n-1))$. Starting from an arbitrary state in the set $\mathcal{X}_1$, the transition structure under **QC** algorithm is the Markov chain displayed in Fig. 6.1; in the diagram,

$$\begin{cases} \text{state } 0: & \text{the vector } \mathbf{0} = [0 \cdots 0]^T \text{ of all zeros,} \\ \text{state } n: & \text{the vector } \mathbf{1} = [1 \cdots 1]^T \text{ of all ones,} \\ \text{state } z: & \text{the vector } [\overbrace{1 \cdots 1}^{z}\ 0 \cdots 0]^T \text{ in } \mathcal{X}_1, \end{cases} \tag{6.6}$$

and the transition probabilities are $p_z = q_z = z(n-z)p$, $z \in [1, n-1]$. To see this, consider the transition from state z to state $z+1$; this occurs when an edge (j,i) is activated, with $x_j = 1$ and $x_i = 0$, so that **(R2)** of **QC** algorithm applies. Since there are $z(n-z)$ such edges, the transition probability $p_z = z(n-z)p$. Likewise, one may derive that the transition from state z to state $z-1$ is with probability $q_z = z(n-z)p$, which occurs when **(R3)** of **QC** algorithm applies. Now observe in Fig. 6.1 that the states $0, n \in \mathscr{C}$ and $1, ..., n-1 \in \mathcal{X}_1$; hence $\max_{z \in [1,n-1]} E_z = \max_{x_0 \in \mathcal{X}_1} E_{qc}(x_0)$, where E_z is from (6.3).

It is left to invoke the formula of E_z in Lemma 6.2 for the obtained transition probabilities, which yields

$$\begin{aligned} E_z &= (1 - \frac{z}{n}) \sum_{i=1}^{z-1} \frac{1}{(n-i)p} + \frac{z}{n} \sum_{j=z}^{n-1} \frac{1}{jp} \\ &\leq (1 - \frac{z}{n}) \frac{z-1}{(n-z+1)p} + \frac{z}{n} \frac{n-z}{zp} \\ &= \frac{n-z}{n-z+1} \cdot \frac{1}{p} < \frac{1}{p} = n(n-1). \end{aligned}$$

Thus $E_z < n(n-1)$ for all $z \in [1, n-1]$. Therefore $\max_{x_0 \in \mathcal{X}_1} E_{qc}(x_0) < n(n-1) = O(n^2)$. ■

Finally, our main result (Theorem 6.1) on upper bounding $E_{qc}(x_0)$ for $x_0 \in \mathcal{X}$ follows immediately from Lemma 6.3 and (6.4).

Remark 6.1. We discuss the idea of how this result for complete graphs might be extended to handle more general topologies. We still view reaching consensus as the interval $[m(k), M(k)]$ shrinking to length 0; thereby the inequality (6.4) holds. We then again consider the subset $\mathcal{X}_1$ given in (6.5), and as long as the digraph is strongly connected (i.e., every node is connected to every other node) one can verify that the state transition structure under **QC** algorithm is still the one in Fig. 6.1. The associated transition probabilities, however, depend crucially on topologies. In order to apply again Lemma 6.2 to derive bounds, it would be important to establish the relation between transition probabilities and graph topologies; this will be targeted in our future work.

6.3 Quantized Averaging Algorithm and Its Lyapunov Function

In this and next sections, we address the convergence time analysis for **QA** algorithm, which is a modification of the one in the preceding chapter. We start by presenting the modified algorithm, and formulate the corresponding time analysis problem. We then propose a Lyapunov function, which turns out to be a suitable measure for the average consensus error. In Section 6.4, we will derive an upper bound on the mean convergence time by means of bounding the decay time of the proposed Lyapunov function.

6.3.1 Problem Formulation and Result Statement

Suppose that an edge (j, i) is randomly activated at time k. There are two stages: (I) Along the edge, node j sends to i its state $x_j(k)$ and surplus $s_j(k)$. Node j does not update its state, but sets its surplus to be 0 after transmission (see Fig. 6.2). (II) Based on the information received, node i determines either to update its state and surplus, or to send back to j the surplus $s_j(k)$ by activating the opposite edge (i, j) (see Fig. 6.3). Notice that the latter operation in (II) requires bidirectional communication

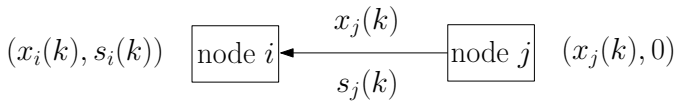


Figure 6.2: Stage (I): Node j sends to i its state and surplus through the edge (j, i).

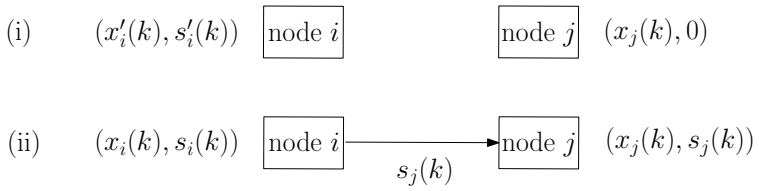


Figure 6.3: Stage (II): Either (i) node i updates its state and surplus, or (ii) it sends $s_j(k)$ back to node j through edge (i, j).

between two nodes at a single time instant; this is possible in complete digraphs (our assumption), but not in general strongly connected digraphs. Formally, **QA** algorithm is described as follows.

(R1) If $x_i(k) = x_j(k)$, then there are two cases:

(i) If $s_i(k) > 0$ & $s_j(k) > 0$, then

$$x_i(k+1) = x_i(k), \quad s_i(k+1) = s_i(k);$$
$$x_j(k+1) = x_j(k), \quad s_j(k+1) = s_j(k).$$

(ii) Otherwise (i.e., either surplus equals zero),

$$x_i(k+1) = x_i(k), \quad s_i(k+1) = s_i(k) + s_j(k) \in \{0, 1\};$$
$$x_j(k+1) = x_j(k), \quad s_j(k+1) = 0.$$

(R2) If $x_i(k) < x_j(k)$, then there are two cases:

(i) If $s_i(k) + s_j(k) > 0$, then

$$x_i(k+1) = x_i(k) + 1, \quad s_i(k+1) = s_i(k) + s_j(k) - 1 \in \{0, 1\};$$
$$x_j(k+1) = x_j(k), \qquad s_j(k+1) = 0.$$

(ii) Otherwise (i.e., $s_i(k) + s_j(k) = 0$),

$$x_i(k+1) = x_i(k), \quad s_i(k+1) = s_i(k) + s_j(k) = 0;$$
$$x_j(k+1) = x_j(k), \quad s_j(k+1) = 0.$$

(R3) If $x_i(k) > x_j(k)$, then there are two cases:

(i) If $s_i(k) + s_j(k) = 0$, then

$$x_i(k+1) = x_i(k) - 1, \quad s_i(k+1) = s_i(k) + s_j(k) + 1 = 1;$$
$$x_j(k+1) = x_j(k), \qquad s_j(k+1) = 0.$$

(ii) Otherwise (i.e., $s_i(k) + s_j(k) > 0$),

$$x_i(k+1) = x_i(k), \quad s_i(k+1) = s_i(k);$$
$$x_j(k+1) = x_j(k), \quad s_j(k+1) = s_j(k).$$

In the algorithm, observe that (1) **(R1)(i)** and **(R3)(ii)** are where node i sends $s_j(k)$ back to node j in stage (II), which requires bidirectional communication; (2) only **(R3)(i)** generates one surplus, and only **(R2)(i)** consumes one surplus; (3) the quantity $(x+s)^T\mathbf{1}$ stays invariant, i.e., for every $k \geq 0$,

$$(x(k+1) + s(k+1))^T\mathbf{1} = (x(k) + s(k))^T\mathbf{1} = x(0)^T\mathbf{1}. \tag{6.7}$$

Distinct from the algorithm in Chapter 5, this **QA** algorithm does not involve the threshold constant and the local extrema variables, thus reducing individual computation effort. Also each surplus variable is indeed binary-valued, and therefore requires merely one bit for both storage and transmission. A further difference between the two algorithms lies in the use of surplus variables: The algorithm in Chapter 5 allows surpluses to pile up, which is indeed required to achieve average consensus for arbitrary strongly connected digraphs. By contrast, our **QA** algorithm here prevents surpluses from piling up, and meanwhile simplifies the transition structure.

Now recall from (5.3) that $\mathscr{A}$ is the set of average consensus states. Here is the convergence result of **QA** algorithm for complete digraphs.

Proposition 6.1. Let Assumption 1 hold. Then, under **QA** algorithm, a network of agents achieves quantized averaging almost surely.

This convergence result may be justified by a similar argument as given in proof of Theorem 5.2; some care, however, has to be taken for the operations on surplus

variables, as pointed out above. For completeness, the proof is provided in Section 6.6. In addition, we note that the convergence can also be implied by the time analysis using Lyapunov approach in Section 6.4 below.

The convergence time of **QA** algorithm is the random variable T_{qa} defined by $T_{qa} := \inf\{k \geq 0 : (x(k), s(k)) \in \mathscr{A}\}$. The mean time taken for this convergence (according again to the probability distribution on edge activation), starting from $(x_0, 0)$ with $x_0 \in \mathcal{X}$, is then given by

$$E_{qa}(x_0) := E\left[T_{qa} | (x(0), 0) = (x_0, 0)\right]. \tag{6.8}$$

Problem 6.2. Let Assumptions 1 and 2 hold. Find an upper bound of the mean convergence time $E_{qa}(x_0)$ of **QA** algorithm with respect to all possible initial states $x_0 \in \mathcal{X}$.

Our main result is the following upper bound of $E_{qa}(x_0)$ with respect to all possible initial states $x_0 \in \mathcal{X}$.

Theorem 6.2. Let Assumptions 1 and 2 hold. Then

$$\max_{x_0 \in \mathcal{X}} E_{qa}(x_0) < n^2(n-1)\frac{3(M-m)}{2} + n(n-1)\frac{R(R-1)}{n-(R/2)} = O(n^3),$$

where $R \in [0, n-1]$ is an integer.

We note that the order of this polynomial bound is the same as that in [45] for undirected, complete graphs. To derive this bound, we will first propose a valid Lyapunov function for **QA** algorithm. Then we will upper bound the mean convergence time by way of upper bounding the mean decay time of the Lyapunov function.

6.3.2 Lyapunov Function

We start by introducing two variables, called positive surplus S_+ and negative surplus S_-; they are global variables, but are needed only for the convergence time analysis. Write the initial state sum $x(0)^T\mathbf{1} = nL + R$, where $L := \lfloor x(0)^T\mathbf{1}/n \rfloor$ is one of the possible values for average consensus, and $0 \leq R < n$. Observe that when a surplus

is generated/consumed, the corresponding state moves one-step either closer to or farther from the value L. Positive and negative surplus variables are used to identify these two directions. Concretely, when a surplus is generated, we increase S_+ (resp. S_-) if the corresponding state moves towards (resp. away from) L. On the other hand, when a surplus is consumed, we distinguish the following two situations: In one case where the state moves closer to L, we decrease S_- if it is nonzero, and S_+ otherwise; in the other case where the state moves away from L, we decrease only S_+.

We now formalize the updating rules of S_+ and S_-. Let $D(k) := \sum_{i=1}^n |x_i(k) - L|$ be the sum of average consensus errors, and suppose that the edge $(j,i) \in \mathcal{E}$ is activated at time k.

(S1) If **(R3)(i)** generates one surplus, then there are two cases:

(i) If $D(k+1) = D(k) - 1$ (i.e., $x_i(k) > L$), then

$$S_+(k+1) = S_+(k) + 1.$$

(ii) If $D(k+1) = D(k) + 1$ (i.e., $x_i(k) \leq L$), then

$$S_-(k+1) = S_-(k) + 1.$$

(S2) If **(R2)(i)** consumes one surplus, then there are also two cases:

(i) If $D(k+1) = D(k) + 1$ (i.e., $x_i(k) \geq L$), then

$$S_+(k+1) = S_+(k) - 1.$$

(ii) If $D(k+1) = D(k) - 1$ (i.e., $x_i(k) < L$), then

$$S_-(k) = 0 \Rightarrow S_+(k+1) = S_+(k) - 1;$$
$$S_-(k) > 0 \Rightarrow S_-(k+1) = S_-(k) - 1.$$

(S3) Otherwise

$$S_+(k+1) = S_+(k);$$
$$S_-(k+1) = S_-(k).$$

The case **(S3)** above includes **(R1)**, **(R2)(ii)**, and **(R3)(ii)** of **QA** algorithm; note that, in these cases, there is no state update. Since initially there is no surplus in the system (i.e., $s(0) = 0$), we set $S_+(0) = S_-(0) = 0$. Also, one may readily see that $S_+(k) + S_-(k) = s(k)^T\mathbf{1}$, which relates the global surpluses to the local ones.

We are ready to define the Lyapunov function $V(k)$, $k \geq 0$, which is given by

$$V(k) := D(k) + S_+(k) - S_-(k). \tag{6.9}$$

It is not difficult to see from **(S1)**-**(S3)** that $V(k)$ is non-increasing. Indeed, $V(k)$ stays put except for only one case – **(S2)(ii)** and negative surplus $S_-(k) = 0$ – where it decreases by 2, i.e., $V(k+1) = V(k) - 2$. Notice that after this decrement, $S_+(k+1) \geq 0$ and $S_-(k+1) = 0$.

Remark 6.2. We compare the Lyapunov function (6.9) with the one given in (4.10) for the dynamic algorithm (4.7) in Chapter 4 where the states are real valued. The Lyapunov function (4.10) was found valid to establish the convergence of the dynamic algorithm (4.7), based on the crucial property that the surpluses are nonnegative. Since **QA** algorithm also has this property, we expect that the function (4.10) may as well be valid for proving the corresponding convergence. On the other hand, for the convergence time analysis (as we will see), by considering decreasing the value of the function (6.9) we can characterize the state transition structure into a special Markov chain. Thereby we can derive the convergence time by computing the corresponding hitting time. Whereas with the function (4.10), it seems difficult to carry out the same approach.

Remark 6.3. We emphasize that the validity of $V(k)$ as a Lyapunov function is not restricted only to undirected graphs, since the updating rules **(S2)** and **(S3)** do not involve **(R1)(i)** and **(R3)(ii)** where bidirectional communication is required. Indeed, $V(k)$ is a suitable Lyapunov function for the original **QA** algorithm in [15–18], which can achieve average consensus on arbitrary strongly connected digraphs. This is one contribution of our work, which might also provide a preliminary to attack convergence time on more general topologies.

In the following lemma, we collect several useful implications from the definition of function $V(k)$.

Lemma 6.4.

(1) A lower bound of $V(k)$ is R, i.e., $V(k) \geq R$ for all k.

(2) If $V(k) = R$, then $S_-(k) = 0$, $S_+(k) \geq 0$, and $(\forall i \in [1, n])\ x_i(k) \geq L$.

(3) If $D(k) = 0$, then $S_-(k) = 0$ and $V(k) = S_+(k) = R$.

(4) Suppose $R = 0$. Then $D(k) = 0$ if and only if $V(k) = 0$, and in both cases $S_-(k) = S_+(k) = 0$.

Proof. We prove these statements in this order: (2), (1), (3), and (4).

(2) Let $V(k) = R$. Then there must exist $k_0 \leq k$ such that $V(k_0 - 1) = R + 2$ and $V(k_0) = R$. Also we have $S_+(k_0) \geq 0$ and $S_-(k_0) = 0$. Now assume $x_1(k_0) < L$. It follows from (6.7) that $x_1(k_0) + \sum_{i=2}^{n} x_i(k_0) + s(k_0)^T \mathbf{1} = nL + R$. Rearranging the terms and by $s(k_0)^T \mathbf{1} = S_+(k_0) + S_-(k_0)$, we obtain $\sum_{i=2}^{n} x_i(k_0) - (n-1)L = (L - x_1(k_0)) + R - S_+(k_0)$. Thus

$$\begin{aligned} V(k_0) &= (L - x_1(k_0)) + \sum_{i=2}^{n} x_i(k_0) + S_+(k_0) - S_-(k_0) \\ &= 2(L - x_1(k_0)) + R > R. \end{aligned}$$

This contradicts $V(k_0) = R$, and hence $x_i(k_0) \geq L$ for all i. The latter holds also for time k because the minimum states are non-decreasing by **QA** algorithm. Finally, according to the updating rules of S_+ and S_-, one may easily see that $S_-(k) = 0$ and $S_+(k) \geq 0$.

(1) When $V(k) = R$, every state $x_i(k) \geq L$ and consequently **(S3)(ii)** cannot occur. As $V(k)$ is non-increasing, it is lower bounded by R.

(3) Let $D(k) = 0$. Then $x(k)^T \mathbf{1} = nL$, and thus $S_+(k) + S_-(k) = s(k)^T \mathbf{1} = R$. It follows that $V(k) = S_+(k) - S_-(k) \leq R$. But $V(k) \geq R$, so that necessarily $V(k) = S_+(k) - S_-(k) = R$, which also implies that $S_-(k) = 0$ and $S_+(k) = R$.

(4) Assume $R = 0$. (Only if) The conclusion follows immediately from (3). (If) Let $V(k) = 0$. Then there must exist $k_0 \leq k$ such that $V(k_0 - 1) = 2$ and $V(k_0) = 0$.

Also we have $S_+(k_0) \geq 0$ and $S_-(k_0) = 0$. Hence $D(k_0) + S_+(k_0) = 0$, which results in $D(k_0) = S_+(k_0) = 0$. As average consensus is achieved at k_0, no further state or surplus update occurs. So the conclusion for time k follows. ■

Next, we find an upper bound for the function $V(k)$.

Proposition 6.2. Let $x(0) \in \mathcal{X}$ in (6.1). Then for every $k \geq 0$,

$$V(k) \leq \frac{(M-m)n}{2} + R.$$

Proof. Since the function $V(k)$ is non-increasing, it suffices to find an upper bound for $V(0) = \sum_{i=1}^{n} |x_i(0) - L|$. Consider the function $V(0) - R$; it is convex in $x(0)$, and $\mathcal{X}$ is a convex set. Hence, one of the extreme points of $\mathcal{X}$ is a maximizer. Fix $r \in [1, n]$, and let $x(0) \in \mathcal{X}$ be such that $x_1(0) = \cdots = x_r(0) = m$ and $x_{r+1}(0) = \cdots = x_n(0) = M$. Then $V(0) - R = r(L-m) + (n-r)(M-L) - R$. Also we have $L = (\mathbf{1}^T x(0) - R)/n = (rm + (n-r)M - R)/n$. Substituting this into the above equation and rearranging the terms, we derive

$$\begin{aligned}
V(0) - R &= -\frac{2(M-m)}{n} r^2 + \left(2(M-m) - 2\frac{R}{n}\right) r \\
&= \frac{2(M-m)}{n} \left[-\left(r - \frac{1}{2}(n - \frac{R}{M-m}) \right)^2 + \frac{1}{4}(n - \frac{R}{M-m})^2 \right] \\
&\leq \frac{2(M-m)}{n} \cdot \frac{1}{4}(n - \frac{R}{M-m})^2 \qquad (\text{ equality holds iff } r = \tfrac{1}{2}(n - \tfrac{R}{M-m}) \,) \\
&= \frac{1}{2} \frac{(n(M-m) - R)^2}{n(M-m)} \\
&\leq \frac{1}{2} \frac{(n(M-m))^2}{n(M-m)} = \frac{(M-m)n}{2} \qquad (\text{ equality holds iff } R = 0 \,).
\end{aligned}$$

Thus $V(k) - R$ is upper bounded by $(M-m)n/2$, which is achievable if and only if $R = 0$ and $r = n/2$. ■

6.4 Convergence Time Analysis of Quantized Averaging Algorithm

We turn now to analyzing the mean convergence time of **QA** algorithm, by way of upper bounding the mean decay time of the Lyapunov function $V(\cdot)$ in (6.9).

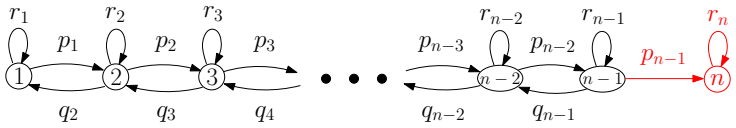


Figure 6.4: Markov chain II: state n is absorbing.

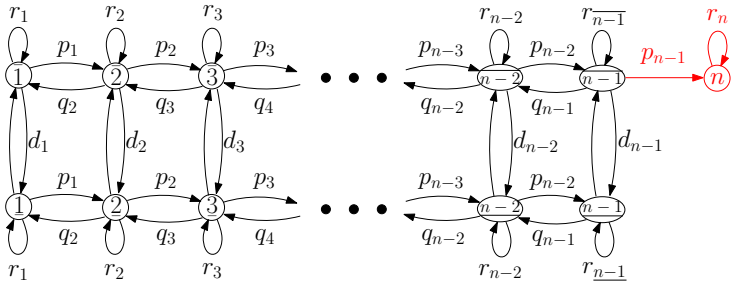


Figure 6.5: Markov chain III: state n is absorbing.

This Lyapunov approach is also adopted in [45, 57]; the common function used is $V'(k) = \sum_{i=1}^{n}(x_i(k) - x(0)^T\mathbf{1}/n)^2$. It can be verified that $V'(k)$ is, however, not a valid Lyapunov function with respect to our **QA** algorithm. This is due to that the state sum is not preserved in each iteration and the surplus evolution must also be taken into account, as in our function $V(k)$.

6.4.1 Preliminaries on Hitting Time

As in Subsection 6.2.2, we provide preliminaries on the hitting time in finite Markov chains, specific to the analysis of **QA** algorithm. For the proofs see Section 6.6.

Lemma 6.5. Consider the Markov chain in Fig. 6.4 with transition probabilities

$$p_1 + r_1 = 1, \qquad p_z + r_z + q_z = 1 \ (z = 2, ..., n-1), \qquad r_n = 1.$$

Then the mean hitting times of the state n starting from state 1 and z are respectively

$$E_1 = \sum_{l=2}^{n-1}\left[\left(\prod_{i=2}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_1} + \sum_{j=2}^{l}\left(\prod_{i=j+1}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_j}\right] + \frac{1}{p_1},$$

$$E_z = \sum_{l=z}^{n-1}\left[\left(\prod_{i=2}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_1} + \sum_{j=2}^{l}\left(\prod_{i=j+1}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_j}\right] \ (z = 2, ..., n-1).$$

Lemma 6.6. Consider the Markov chain in Fig. 6.5 with transition probabilities

$$p_1 + r_1 + d_1 = 1, \qquad p_z + r_z + q_z + d_z = 1 \ (z = 2, ..., n-2),$$
$$r_{\underline{n-1}} + q_{n-1} + d_{n-1} = 1, \qquad p_{n-1} + r_{\overline{n-1}} + q_{n-1} + d_{n-1} = 1, \quad r_n = 1.$$

Here $\underline{\cdot}$ and $\overline{\cdot}$ denote the states of the lower and upper rows, respectively. Then for states $\underline{n-1}$ and $\overline{n-1}$, their mean hitting times of the absorbing state n are

$$E_{\overline{n-1}} = \left(\prod_{i=2}^{n-1} \frac{q_i}{p_i}\right) \cdot \frac{2}{p_1} + \sum_{j=2}^{n-1} \left(\prod_{i=j+1}^{n-1} \frac{q_i}{p_i}\right) \cdot \frac{2}{p_j},$$
$$E_{\underline{n-1}} < \left(1 + \frac{p_{n-1}}{d_{n-1}}\right) E_{\overline{n-1}}.$$

In the rest of this section, the proof of Theorem 6.2 is given. We will need the following notation. Define the random variable $T_V := \inf\{k \geq 0 : V(k) = R\}$; thus T_V is the time when $V(\cdot)$ decreases to R. The mean decay time, starting from $(x_0, 0)$, is then given by

$$E_V(x_0) := E\left[T_V | (x(0), 0) = (x_0, 0)\right]. \tag{6.10}$$

According to the value of $R \in [0, n-1]$, we will proceed with two cases in this order: $R = 0$ and $R > 0$. When $R = 0$ the mean convergence time $E_{qa}(x_0)$ is found to satisfy $E_{qa}(x_0) = E_V(x_0)$, whereas when $R > 0$ we have $E_{qa}(x_0) \geq E_V(x_0)$ in general and the corresponding analysis turns out to be based on the former case.

6.4.2 Proof for the case $R = 0$

In this case, the mean convergence time $E_{qa}(x_0)$ is characterized by the mean time that the function $V(k)$ decays to 0; that is, $E_{qa}(x_0) = E_V(x_0)$ in (6.10). This is because by Lemma 6.4 (4), $V(k) = 0$ if and only if $D(k) = 0$, and the latter implies $(x(k), s(k)) \in \mathscr{A}$. As each decrement reduces $V(k)$ by 2, the initial value $V(0)$ is necessarily even, and there need in total $V(0)/2$ decrements (see Fig. 6.6).

To upper bound $E_V(x_0)$, we view the decay of $V(k)$ as the descent of *level sets* in the $(n+2)$-dimensional space of the triples $u := (x, S_+, S_-)$ (see Fig. 6.7). In this

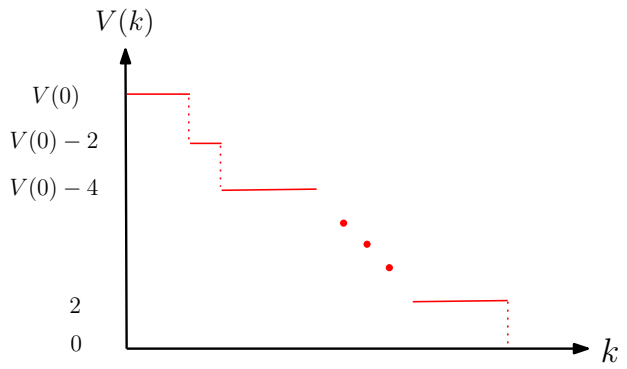


Figure 6.6: Decay of function $V(k)$ in case $R = 0$

space, the average consensus state is simply the point $(L\mathbf{1}, 0, 0)$. Define the level sets

$$\mathcal{U}_l := \{u : V = \sum_{i=1}^{n} |x_i - L| + S_+ - S_- = 2 \cdot l\}, \quad l = 1, ..., V(0)/2.$$

Thus when $u(k) \in \mathcal{U}_l$, we interpret that $(x(k), s(k))$ is l-step away from $\mathscr{A}$ (i.e., $V(k)$ requires l decrements to reach 0). Also, it is important to note that on every level set $\mathcal{U}_l$, the triple evolution may start, and may descend to the next level, only from a strict subset $\mathcal{U}_l^0$ defined by

$$\mathcal{U}_l^0 := \{u \in \mathcal{U}_l : S_- = 0 \ \& \ S_+ \geq 0\}.$$

To see this, first recall that the decrement of $V(\cdot)$ (i.e., level set descent) requires $S_- = 0$ and $S_+ > 0$. Moreover, for the outmost level $\mathcal{U}_{V(0)/2}$, the initial triple is of the form $(x_0, 0, 0)$; and for each subsequent level, the triple evolution starts right after descending from the preceding level, where we have $S_- = 0$ and $S_+ \geq 0$.

Now let the random variable T_1 be the time of *one decrement* of $V(\cdot)$. The corresponding mean time, starting from a triple $u \in \mathcal{U}_l^0$, is then given by $E_1^l(u) := E\left[T_1 | u \in \mathcal{U}_l^0\right]$, $l \in [1, V(0)/2]$. Since the initial value $V(0)$ is upper bounded by $(M-m)n/2$ (Proposition 6.2), the function $V(\cdot)$ requires at most $(M-m)n/4$ decrements to reach 0. Hence, an upper bound of its mean decay time is the following:

$$\max_{x_0 \in \mathcal{X}} E_V(x_0) \leq \max_{l \in [1, V(0)/2], u \in \mathcal{U}_l^0} E_1^l(u) \cdot \frac{(M-m)n}{4}. \tag{6.11}$$

Here is a key result.

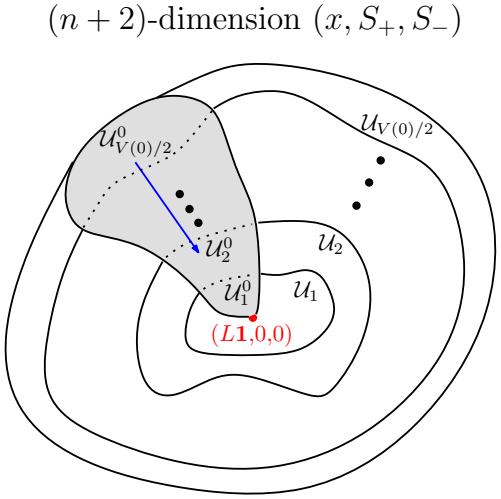


Figure 6.7: Decay of $V(k)$ viewed as level set descent in the $(n+2)$ dimensions of (x, S_+, S_-). Descending is possible only from the shaded area and through the dotted curves.

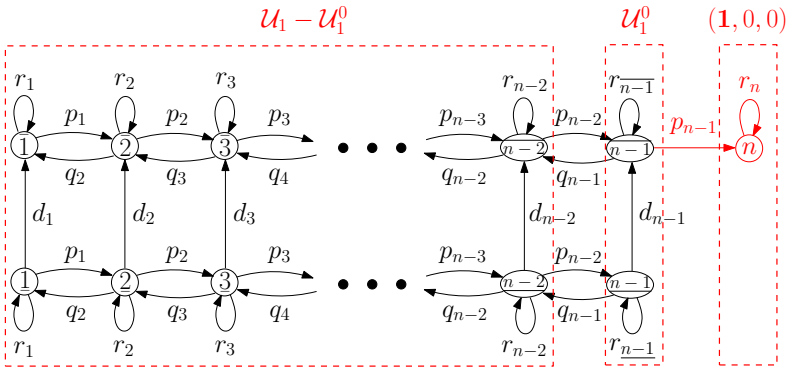


Figure 6.8: One step away: from $\mathcal{U}_1$ to $(\mathbf{1}, 0, 0)$.

Proposition 6.3. Let Assumptions 1 and 2 hold. Then

$$\max_{l\in[1,V(0)/2],u\in\mathcal{U}^0_l} E^l_1(u) < 6n(n-1) = O(n^2).$$

To prove Proposition 6.3, it suffices to establish

$$\max_{u\in\mathcal{U}^0_l} E^l_1(u) < 6n(n-1) = O(n^2), \tag{6.12}$$

for every $l \in [1, V(0)/2]$. In the sequel we will provide the proof for the case $l = 1$ (i.e., one step away from average consensus), which contains the essential idea of our argument. Specifically, we first exhaust the possible triple evolution under **QA** algorithm, second derive the evolution structure and transition probabilities, and third

calculate the corresponding mean hitting time. The analysis of the case $l \geq 2$ follows in a similar fashion but is more involved; we refer to Appendix for the proof.

Proof for the case $l = 1$: Without loss of generality let $L = 1$. We investigate the triple evolution from the level set $\mathcal{U}_1$, starting in $\mathcal{U}_1^0$, to the average consensus state $(\mathbf{1}, 0, 0)$. By Assumptions 1 and 2, every directed edge in $\mathcal{G}$ can be activated with the uniform probability $p = 1/(n(n-1))$. Consider the triple $([2 \overbrace{1 \cdots 1}^{n-2} 0]^T,\ 0,\ 0) \in \mathcal{U}_1^0$; we show that either S_- or S_+ can be generated. Case 1: an edge (j, i) is activated, with $x_j = 0$ and $x_i = 1$. In this case, **(R3)(i)** of **QA** algorithm applies, and the resulting triple is $([2 \overbrace{1 \cdots 1}^{n-3} 0\ 0]^T,\ 0,\ 1) \in \mathcal{U}_1 - \mathcal{U}_1^0$. There are $n-2$ such edges; so the probability of this transition is $(n-2)p$. In fact, such transitions can continue until all the ones become zeros, generating in total $S_- = n - 2$. Case 2: an edge (j, i) is activated, with $x_j = 0$ or 1 and $x_i = 2$. Again **(R3)(i)** of **QA** algorithm applies, the resulting triple being $(\overbrace{[1\ 1 \cdots 1}^{n-1} 0]^T,\ 1,\ 0) \in \mathcal{U}_1^0$. This transition is with probability $(n-1)p$, since there are $n-1$ such edges.

Now starting from the triple $(\overbrace{[1\ 1 \cdots 1}^{n-1} 0]^T,\ 1,\ 0)$, on one hand, we can have a similar process, as from $([2 \overbrace{1 \cdots 1}^{n-2} 0]^T,\ 0,\ 0)$ described above, generating in total $S_- = n - 2$. On the other hand, observe that there is only one edge (j, i) such that $x_j = 1$, $s_j = 1$, and $x_i = 0$, $s_i = 0$. If this edge is activated (with probability p), then **(R2)(i)** of **QA** algorithm applies, and the resulting triple is the average consensus state $(\mathbf{1}, 0, 0)$.

Based on the above descriptions, we derive that the transition structure from $\mathcal{U}_1$ to $(\mathbf{1}, 0, 0)$ under **QA** algorithm is the one displayed in Fig. 6.8.[1] In this diagram, the state n is the average consensus state $(\mathbf{1}, 0, 0)$, and the other states belong to $\mathcal{U}_1$, listed below:

[1]The transition structure in Fig. 6.8 is obtained with a minor modification from the original. For those triples in $\mathcal{U}_1 - \mathcal{U}_1^0$, we treat the following transitions from left to right as selfloops: For some node i such that $x_i = 0$ and $s_i = 0$, its state x_i increases by consuming one negative surplus (under **R2(i)** of **QA** algorithm). By treating such transitions as selfloops, only the probability of moving towards the average consensus state is reduced; so it can be verified that the mean hitting time derived from this structure is an upper bound of that from the original. We make such modifications in our analysis henceforth.

$\underline{n-1}:\ ([2\ 1\ 1\ \cdots\ 1\ 1\ 0]^T,\ 0,\ 0)$	$\overline{n-1}:\ ([1\ 1\ 1\ \cdots\ 1\ 1\ 0]^T,\ 1,\ 0)$
$\underline{n-2}:\ ([2\ 1\ 1\ \cdots\ 1\ 0\ 0]^T,\ 0,\ 1)$	$\overline{n-2}:\ ([1\ 1\ 1\ \cdots\ 1\ 0\ 0]^T,\ 1,\ 1)$
$\vdots$	$\vdots$
$\underline{2}:\ ([2\ 1\ 0\ \cdots\ 0\ 0\ 0]^T,\ 0,\ n-3)$	$\overline{2}:\ ([1\ 1\ 0\ \cdots\ 0\ 0\ 0]^T,\ 1,\ n-3)$
$\underline{1}:\ ([2\ 0\ 0\ \cdots\ 0\ 0\ 0]^T,\ 0,\ n-2)$	$\overline{1}:\ ([1\ 0\ 0\ \cdots\ 0\ 0\ 0]^T,\ 1,\ n-2)$

Note that negative surplus is zero ($S_- = 0$) only in the states $\underline{n-1}$ and $\overline{n-1}$; hence these two triples are in $\mathcal{U}_1^0$. Also, one may verify that the transition probabilities are as follows:

$$p_1 = (n-2)p, \quad d_1 = p; \quad p_{n-1} = p, \quad q_{n-1} = (n-2)p, \quad d_{n-1} = (n-1)p;$$
$$p_z = (n-1-z)zp, \quad q_z = (z-1)p, \quad d_z = zp \ \ (z = 2, ..., n-2).$$

To upper bound $E_1^1(u)$ for $u \in \mathcal{U}_1^0$, in Fig. 6.8 we add transitions from the state $\overline{z}$ to $\underline{z}$ with the probability d_z, $z \in [1, n-1]$, thereby increasing the probabilities of moving away from the average consensus state n. This modification leads us to the same structure displayed in Fig. 6.5; thus, we have $\max_{u \in \mathcal{U}_1^0} E_1^1(u) \leq E_{\underline{n-1}}$, where $E_{\underline{n-1}}$ is given in (6.3).

It is left to calculate $E_{\underline{n-1}}$ with respect to the obtained transition probabilities. For this we invoke the formulas in Lemma 6.6. First,

$$\prod_{i=2}^{n-1} \frac{q_i}{p_i} = \frac{n-2}{1} \cdot \frac{n-3}{n-2} \cdot \frac{n-4}{2(n-3)} \ \cdots \ \frac{2}{(n-4)3} \cdot \frac{1}{(n-3)2} = \frac{1}{(n-3)!}.$$

Similarly,

$$\prod_{i=3}^{n-1} \frac{q_i}{p_i} = \frac{2}{(n-4)!}, \ \prod_{i=4}^{n-1} \frac{q_i}{p_i} = \frac{3}{(n-5)!}, \ \cdots, \ \frac{q_{n-2}q_{n-1}}{p_{n-2}p_{n-1}} = n-3, \ \frac{q_{n-1}}{p_{n-1}} = n-2.$$

We then have

$$\begin{aligned} E_{\overline{n-1}} &= \left(\prod_{i=2}^{n-1} \frac{q_i}{p_i}\right) \cdot \frac{2}{p_1} + \sum_{j=2}^{n-1} \left(\prod_{i=j+1}^{n-1} \frac{q_i}{p_i}\right) \cdot \frac{2}{p_j} \\ &= \frac{1}{(n-3)!} \cdot \frac{2}{(n-2)p} + \frac{2}{(n-4)!} \cdot \frac{2}{(n-3)2p} + \cdots + (n-2) \cdot \frac{2}{(n-2)p} + \frac{2}{p} \\ &= \frac{2}{p} \cdot \left[\frac{1}{(n-2)!} + \frac{1}{(n-3)!} + \cdots + 1 + 1\right] \\ &< \frac{2}{p} \cdot 3 = 6n(n-1) = O(n^2). \end{aligned}$$

Finally, $E_{\underline{n-1}} < (1 + (p_{n-1}/d_{n-1}))\, E_{\overline{n-1}} = (1 + (p/((n-1)p))) \cdot 6n(n-1) < 6n(n-1) = O(n^2)$. ■

Therefore, it follows from Proposition 6.3 and equation (6.11) that the upper bound of $E_{qa}(x_0)$ in Theorem 6.2 holds for the case $R = 0$.

6.4.3 Proof for the case $R \in [1, n-1]$

When $R \neq 0$, we have $E_{qa}(x_0) \geq E_V(x_0)$ in general. This is because $V(k) = R$ does not generally imply $(x(k), s(k)) \in \mathscr{A}$, and even after $V(k)$ reaches its lower bound R (Lemma 6.4 (1) and (2)), the pair $(x(k), s(k))$ may require extra time to reach $\mathscr{A}$. Define the level set $\mathcal{U}_R := \{u : V = \sum_{i=1}^{n} |x_i - L| + S_+ - S_- = R\}$; then the mean convergence time starting from a triple $u \in \mathcal{U}_R$ is given by $E_{qa}(u) := E\,[T_{qa} | u \in \mathcal{U}_R]$. Also recall from (6.10) that $E_V(x_0)$, with $x_0 \in \mathcal{X}$ in (6.1), denotes the mean decay time of $V(k)$ to the lower bound R. From these we obtain the mean convergence time of **QA** algorithm

$$\max_{x_0 \in \mathcal{X}} E_{qa}(x_0) \leq \max_{x_0 \in \mathcal{X}} E_V(x_0) + \max_{u \in \mathcal{U}_R} E_{qa}(u). \tag{6.13}$$

In the sequel, we find upper bounds for $E_V(x_0)$ and $E_{qa}(u_R)$, respectively. First, as in the case $R = 0$, we have

$$\max_{x_0 \in \mathcal{X}} E_V(x_0) < n^2(n-1)\frac{3(M-m)}{2} = O(n^3). \tag{6.14}$$

This is due to the following reason. The function $V(k)$ decays from its initial value $V(0)$ to R, and each decrement reduces $V(k)$ by 2. It follows that $V(0) - R$ is necessarily even and there need in total $(V(0) - R)/2$ decrements. For $l \in [1, (V(0) - R)/2]$ recall that $E_1^l(u)$ denotes the mean time spent for one decrement of $V(k)$, starting from a triple $u \in \mathcal{U}_l^0$. Following Proposition 6.3, one may similarly derive that $\max_{l \in [1,(V(0)-R)/2], u \in \mathcal{U}_l^0} E_1^l(u) < 6n(n-1)$. Moreover, $V(0) - R \leq (M-m)n/2$ by Proposition 6.2; thus $V(k)$ requires at most $(M-m)n/4$ decrements to reach R. Therefore, $\max_{x_0 \in \mathcal{X}} E_V(x_0) \leq \max_{l \in [1,(V(0)-R)/2], u \in \mathcal{U}_l^0} E_1^l(u) \cdot (M-m)n/4 < n^2(n-1)3(M-m)/2 = O(n^3)$.

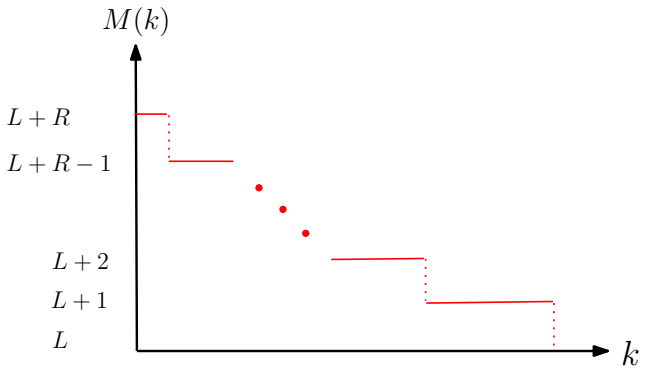


Figure 6.9: Decrement of maximum state when $u \in \mathcal{U}_R$

Next, we find an upper bound for $\max_{u\in\mathcal{U}_R} E_{qa}(u)$. By Lemma 6.4 (2) we have $(\forall i \in \mathcal{V})\ x_i \geq L$; so the maximum state $M(k)$ in (5.4) satisfies $M(k) \in [L, L+R]$. If $R = 1$, then in fact $(x(k), s(k)) \in \mathscr{A}$; thus in this case $E_{qa}(u_R) = 0$, and we have from (6.13) and (6.14) that $\max_{x_0\in\mathcal{X}} E_{qa}(x_0) = O(n^3)$. It is left to consider $R \in [2, n-1]$. Since $M(k) = L$ or $L+1$ implies $(x(k), s(k)) \in \mathscr{A}$, the mean convergence time $E_{qa}(u)$ can be characterized by the mean time that $M(k)$ decays to $L+1$. The decay of $M(k)$ is displayed in Fig. 6.9; observe that $M(k)$ requires at most $R-1$ decrements to reach $L+1$. Let $E_M(u)$ denote the mean time taken for one decrement of $M(k)$, starting from a triple $u \in \mathcal{U}_R$. Then an upper bound for $E_{qa}(u)$ is as follows:

$$\max_{u\in\mathcal{U}_R} E_{qa}(u) \leq \max_{u\in\mathcal{U}_R} E_M(u) \cdot (R-1). \tag{6.15}$$

Proposition 6.4. Let Assumptions 1 and 2 hold. Then

$$\max_{u\in\mathcal{U}_R} E_M(u) < n(n-1)\frac{R}{n-(R/2)} = O(n^2).$$

To prove Proposition 6.4, we first find the subset in which one decay of $M(k)$ takes the longest time, second derive the transition structure and probabilities under **QA** algorithm, and third compute the mean hitting time.

Proof of Proposition 6.4. We consider the following two cases when R is even and odd, respectively.

1) R is even. Let $\mathcal{U}_e$ be a subset of $\mathcal{U}_R$ given by $\mathcal{U}_e := \{u = (x, S_+, S_-) : x \in \mathcal{X}_e,\ S_+ = S_- = 0\}$, where

$$\mathcal{X}_e := \{x : x_1 = \cdots = x_{\frac{R}{2}} = L+2,\quad x_{\frac{R}{2}+1} = \cdots = x_n = L\}.$$

For a state in $\mathcal{X}_e$, one decrement of its maximum value $L+2$ occurs only when all the $R/2$ state components having that value decrease; thus it is not hard to see $\max_{u\in\mathcal{U}_R} E_M(u) = \max_{u\in\mathcal{U}_e} E_M(u)$.

Now pick an arbitrary triple u in $\mathcal{U}_e$; we investigate its evolution under **QA** algorithm. If an edge (j,i) is activated, with $x_j = L$ and $x_i = L+2$, then **(R3)(i)** of **QA** algorithm applies, and the resulting triple is $([\overbrace{L+2\ \cdots L+2}^{(R/2)-1}\ L+1\ L\ \cdots L]^T,\ 1,\ 0)$. Namely, one maximum state decreases. Also observe that there are $(R/2)\,(n-(R/2))$ such edges; so the probability of this transition is $(R/2)\,(n-(R/2))\,p$, where $p = 1/(n(n-1))$ by Assumptions 1 and 2. Indeed, this process can continue until all the $R/2$ maximum states decrease to the value $L+1$, and we derive that the corresponding transition structure under **QA** algorithm is the one displayed in Fig. 6.4 with the length $n = (R/2)+1$. In the diagram,

$$\begin{cases} \text{state } 1: & ([\overbrace{L+2\ L+2\ \cdots\ L+2\ L+2}^{R/2}\ L\ \cdots\ L]^T,\ 0,\ 0) \\ \text{state } 2: & ([L+2\ L+2\ \cdots\ L+2\ L+1\ L\ \cdots\ L]^T,\ 1,\ 0) \\ \quad\vdots & \\ \text{state } R/2: & ([L+2\ L+1\ L+1\ \cdots\ L+1\ L\ \cdots\ L]^T,\ (R/2)-1,\ 0) \\ \text{state } (R/2)+1: & ([L+1\ L+1\ L+1\ \cdots\ L+1\ L\ \cdots\ L]^T,\ R/2,\ 0) \end{cases}$$

and the transition probabilities are $p_1 = (R/2)\,(n-(R/2))\,p$, $p_z = ((R/2)-z+1)(n-(R/2))p$, $q_z = (z-1)((R/2)-z+1)p$, $z\in[2,R/2]$. Observe that the state $1\in\tilde{\mathcal{U}}_R$ and the state $(R/2)+1\in\mathcal{A}$; so $\max_{u\in\mathcal{U}_e} E_M(u) = E_1$, where E_1 is from (6.3).

It remains to invoke the formulas in Lemma 6.5 to calculate E_1. First,

$$\begin{aligned}\prod_{i=2}^{R/2}\frac{q_i}{p_i} &= \frac{(R/2)-1}{n-(R/2)}\cdot\frac{((R/2)-2)2}{2(n-(R/2))}\ \cdots\ \frac{2((R/2)-2)}{((R/2)-2)(n-(R/2))}\cdot\frac{(R/2)-1}{((R/2)-1)(n-(R/2))} \\ &= \frac{((R/2)-1)!}{(n-(R/2))^{(R/2)-1}} \le \left(\frac{(R/2)-1}{n-(R/2)}\right)^{(R/2)-1} < 1;\end{aligned}$$

the last inequality is due to $R<n$. Similarly $\prod_i^{R/2} q_i/p_i < 1$ for $i = 3,...,R/2$. Then

we obtain

$$\left(\prod_{i=2}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_1}+\sum_{j=2}^{l}\left(\prod_{i=j+1}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_j} < \frac{1}{(n-(R/2))p}\left(\frac{1}{(R/2)}+\frac{1}{(R/2)-1}+\cdots+\frac{1}{(R/2)-l+1}\right).$$

Hence,

$$\begin{aligned}E_1 &= \sum_{l=2}^{R/2}\left[\left(\prod_{i=2}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_1}+\sum_{j=2}^{l}\left(\prod_{i=j+1}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_j}\right]+\frac{1}{p_1}\\ &< \frac{1}{(n-(R/2))p}\left(\frac{1}{R/2}+\frac{1}{(R/2)-1}+\cdots+\frac{1}{2}+1\right)\\ &\quad+\frac{1}{(n-(R/2))p}\left(\frac{1}{R/2}+\frac{1}{(R/2)-1}+\cdots+\frac{1}{2}\right)\\ &\quad+\cdots+\frac{1}{(n-(R/2))p}\left(\frac{1}{R/2}+\frac{1}{(R/2)-1}\right)+\frac{1}{(n-(R/2))p}\cdot\frac{1}{R/2}\\ &= \frac{R}{(n-(R/2))p}=\frac{R}{(n-(R/2))}\cdot n(n-1).\end{aligned}$$

Therefore, $\max_{u\in\mathcal{U}_R}E_M(u)=E_1<n(n-1)R/(n-(R/2))=O(n^2)$.

2) R is odd. Let $\mathcal{U}_o$ be a subset of $\mathcal{U}_R$ given by $\mathcal{U}_o := \{u=(x,S_+,S_-) : x\in\mathcal{X}_o,\ S_+=S_-=0\}$, where

$$\mathcal{X}_o := \{x : x_1=\cdots=x_{\frac{R-1}{2}}=L+2,\quad x_{\frac{R+1}{2}}=L+1,\quad x_{\frac{R+1}{2}+1}=\cdots=x_n=L\}.$$

For the same reason in the preceding case, one can verify that $\max_{u\in\mathcal{U}_R}E_M(u)=\max_{u\in\mathcal{U}_o}E_M(u)$. Also it turns out that the transition structure, together with the associated transition probabilities, starting from $\mathcal{U}_o$ is analogous to that starting from $\mathcal{U}_e$. Thus by a similar derivation given above, we can conclude again that $\max_{u\in\mathcal{U}_R}E_M(u)<n(n-1)R/(n-(R/2))=O(n^2)$. ■

Finally, it follows from equations (6.13)-(6.15) and Proposition 6.4 that an upper bound of the mean convergence time $E_{qa}(x_0)$ of **QA** algorithm is $E_{qa}(x_0)<n^2(n-1)3(M-m)/2+n(n-1)R(R-1)/(n-(R/2))=O(n^3)$ for the case $R>0$. This completes the proof of Theorem 6.2.

Remark 6.4. We have derived an upper bound for the convergence time of **QA** algorithm on complete graphs, by proposing a suitable Lyapunov function for the

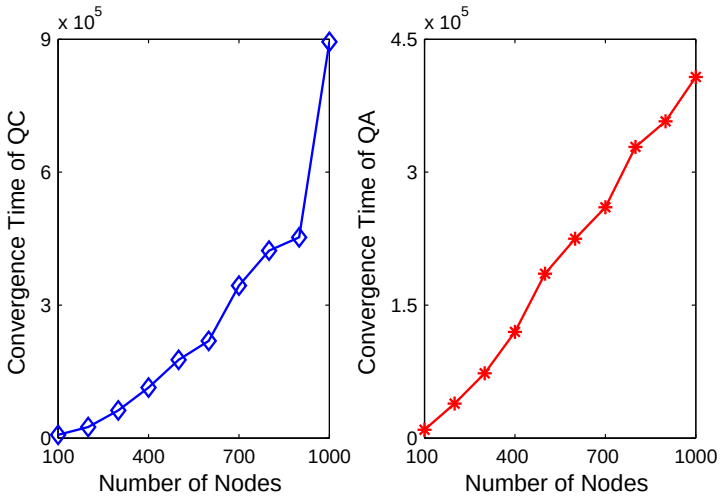


Figure 6.10: Convergence time of **QC** and **QA**

algorithm and characterizing a Markov chain for the state-surplus transition structure. To extend this result to more general topologies, the Lyapunov function is still valid (see Remark 6.3) which in turn validates inequalities (6.11) and (6.13). Thus it is crucial to establish the relation between graph topologies and the transition structure with associated probabilities, as done in the proofs of Propositions 6.3 and 6.4 for complete graphs. Establishing such a relation for general topologies currently appears to be difficult, but will be explored in our future work.

6.5 Numerical Example

We have proved polynomial upper bounds on the convergence time of **QC** and **QA** algorithms for complete digraphs. Now we compare these theoretic bounds with numerical simulations, so as to illustrate the tightness of our derived results. For this purpose, we consider the following initial states $x(0)$ which correspond to the worst case convergence time: For **QC** algorithm, we choose $x(0) = [\overbrace{1 \cdots 1}^{\lfloor n/2 \rfloor} \; 0 \cdots 0]^T$ (cf. proof of Lemma 6.3); for **QA** algorithm, we choose $x(0) = [2 \; \overbrace{1 \cdots 1}^{n-2} \; 0]^T$ (cf. proof of Proposition 6.3). The simulation results are displayed in Fig. 6.10, each plotted value being the mean convergence time of 100 runs of the corresponding algorithms.

It is observed that the convergence rate of **QC** algorithm is approximately quadrat-

ic, which demonstrates that the derived theoretic bound is relatively tight. On the other hand, the convergence rate of **QA** algorithm appears to be at most quadratic, if not linear. This indicates that the cubic theoretic bound may not be tight, though it is in the same order as the one in [45] also for complete graphs. Thus, deriving tighter bounds for the convergence time of **QA** algorithm awaits future effort.

6.6 Proofs

Proof of Lemma 6.2. The proof is a direct calculation. By Lemma 6.1 the mean hitting times of state 0 or n satisfy the following linear equations

$$E_0 = 0, \tag{6.16}$$

$$E_z = p_z E_{z+1} + r_z E_z + q_z E_{z-1} + 1, \quad z = 1, ..., n-1, \tag{6.17}$$

$$E_n = 0. \tag{6.18}$$

Since $p_z = q_z$, it follows from (6.17) that $p_z(E_{z+1} - E_z) - p_z(E_z - E_{z-1}) + 1 = 0$. Let $F_{z+1} := E_{z+1} - E_z$. Then

$$F_{z+1} = F_z - \frac{1}{p_z}.$$

This is a non-homogeneous first-order linear difference equation, whose solution is of the general form

$$F_{z+1} = F_1 - \sum_{i=1}^{z} \frac{1}{p_i}.$$

To obtain the initial condition F_1, consider

$$F_n + F_{n-1} + \cdots + F_1 = (E_n - E_{n-1}) + (E_{n-1} - E_{n-2}) + \cdots + (E_1 - E_0) = 0,$$

$$F_n + F_{n-1} + \cdots + F_1 = nF_1 - \sum_{j=1}^{n-1} \sum_{i=1}^{j} \frac{1}{p_i}.$$

From the above we have $F_1 = (1/n)\sum_{j=1}^{n-1}\sum_{i=1}^{j} 1/p_i$. Finally,

$$\begin{aligned} E_z = E_z - E_0 &= F_z + F_{z-1} + \cdots + F_2 + F_1 \\ &= zF_1 - \sum_{j=1}^{z-1}\sum_{i=1}^{j}\frac{1}{p_i} \\ &= \frac{z}{n}\sum_{j=1}^{n-1}\sum_{i=1}^{j}\frac{1}{p_i} - \sum_{j=1}^{z-1}\sum_{i=1}^{j}\frac{1}{p_i} \\ &= (1-\frac{z}{n})\sum_{i=1}^{z-1}\frac{i}{p_i} + \frac{z}{n}\sum_{j=z}^{n-1}\frac{n-j}{p_j}. \end{aligned}$$

■

Proof of Lemma 6.5. By Lemma 6.1 the mean hitting times of state n satisfy the following linear equations

$$E_n = 0, \tag{6.19}$$

$$E_1 = p_1 E_2 + r_1 E_1 + 1, \tag{6.20}$$

$$E_z = p_z E_{z+1} + r_z E_z + q_z E_{z-1} + 1, \quad z = 2, ..., n-1. \tag{6.21}$$

Rearrange the terms in (6.21) to obtain $p_z(E_{z+1} - E_z) - q_z(E_z - E_{z-1}) + 1 = 0$. Let $F_{z+1} := E_{z+1} - E_z$. Then

$$F_{z+1} = \frac{q_z}{p_z}F_z - \frac{1}{p_z},$$

whose initial condition is $F_2 = E_2 - E_1 = -1/p_1$ by (6.20). This is a non-homogeneous first-order linear difference equation with variable coefficients, whose solution is of the general form

$$F_{z+1} = \left(\prod_{i=2}^{z}\frac{q_i}{p_i}\right)\cdot(-\frac{1}{p_1}) + \sum_{j=2}^{z}\left(\prod_{i=j+1}^{z}\frac{q_i}{p_i}\right)\cdot\left(-\frac{1}{p_j}\right).$$

Since

$$\begin{aligned} F_n + F_{n-1} + \cdots + F_{z+1} &= (E_n - E_{n-1}) + (E_{n-1} - E_{n-2}) + \cdots + (E_{z+1} - E_z) \\ &= E_n - E_z = -E_z, \end{aligned}$$

we derive

$$E_z = -(F_n + F_{n-1} + \cdots + F_{z+1}) = \sum_{l=z}^{n-1}\left[\left(\prod_{i=2}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_1} + \sum_{j=2}^{l}\left(\prod_{i=j+1}^{l}\frac{q_i}{p_i}\right)\cdot\frac{1}{p_j}\right].$$

Finally,

$$E_1 = E_2 + \frac{1}{p_1} = \sum_{l=2}^{n-1} \left[\left(\prod_{i=2}^{l} \frac{q_i}{p_i} \right) \cdot \frac{1}{p_1} + \sum_{j=2}^{l} \left(\prod_{i=j+1}^{l} \frac{q_i}{p_i} \right) \cdot \frac{1}{p_j} \right] + \frac{1}{p_1}.$$

■

Proof of Lemma 6.6. It follows from Lemma 6.1 that the mean hitting times of state n satisfy the following linear equations

$$\begin{cases} E_{\overline{1}} = p_1 E_{\overline{2}} + r_1 E_{\overline{1}} + d_1 E_{\underline{1}} + 1, \\ E_{\underline{1}} = p_1 E_{\underline{2}} + r_1 E_{\underline{1}} + d_1 E_{\overline{1}} + 1; \end{cases} \tag{6.22}$$

$$\begin{cases} E_{\overline{z}} = p_z E_{\overline{z+1}} + r_1 E_{\overline{z}} + q_z E_{\overline{z-1}} + d_z E_{\underline{z}} + 1, \\ E_{\underline{z}} = p_z E_{\underline{z+1}} + r_1 E_{\underline{z}} + q_z E_{\underline{z-1}} + d_z E_{\overline{z}} + 1; \end{cases} \quad (z = 2, ..., n-2) \tag{6.23}$$

$$\begin{cases} E_{\overline{n-1}} = p_{n-1} E_n + r_{\overline{n-1}} E_{\overline{n-1}} + q_{n-1} E_{\overline{n-2}} + d_{n-1} E_{\underline{n-1}} + 1, \\ E_{\underline{n-1}} = r_{\underline{n-1}} E_{\underline{n-1}} + q_{n-1} E_{\underline{n-2}} + d_{n-1} E_{\overline{n-1}} + 1; \end{cases} \tag{6.24}$$

$$E_n = 0. \tag{6.25}$$

Rearrange the terms in (6.23) as

$$\begin{cases} p_z(E_{\overline{z+1}} - E_{\overline{z}}) - q_z(E_{\overline{z}} - E_{\overline{z-1}}) - d_z((E_{\overline{z}} - E_{\underline{z}})) + 1 = 0, \\ p_z(E_{\underline{z+1}} - E_{\underline{z}}) - q_z(E_{\underline{z}} - E_{\underline{z-1}}) + d_z((E_{\overline{z}} - E_{\underline{z}})) + 1 = 0. \end{cases}$$

Let $F_{\overline{z+1}} := E_{\overline{z+1}} - E_{\overline{z}}$, $F_{\underline{z+1}} := E_{\underline{z+1}} - E_{\underline{z}}$, and add these two equations; we obtain

$$F_{\overline{z+1}} + F_{\underline{z+1}} = \frac{q_z}{p_z}(F_{\overline{z}} + F_{\underline{z}}) - \frac{2}{p_z},$$

whose initial condition is $F_{\overline{2}} + F_{\underline{2}} = -2/p_1$ by (6.22). This is again a non-homogeneous first-order linear difference equation with variable coefficients, whose solution is

$$F_{\overline{z+1}} + F_{\underline{z+1}} = \left(\prod_{i=2}^{z} \frac{q_i}{p_i} \right) \cdot (-\frac{2}{p_1}) + \sum_{j=2}^{z} \left(\prod_{i=j+1}^{z} \frac{q_i}{p_i} \right) \cdot \left(-\frac{2}{p_j} \right).$$

Now rearrange the terms in (6.24)

$$\begin{cases} p_{n-1}(E_n - E_{\overline{n-1}}) - q_{n-1}(E_{\overline{n-1}} - E_{\overline{n-2}}) - d_{n-1}((E_{\overline{n-1}} - E_{\underline{n-1}})) + 1 = 0, \\ -q_{n-1}(E_{\underline{n-1}} - E_{\underline{n-2}}) + d_{n-1}((E_{\overline{n-1}} - E_{\underline{n-1}})) + 1 = 0. \end{cases}$$

Adding these two equations and applying (6.25), we derive

$$E_{\overline{n-1}} = -\frac{q_{n-1}}{p_{n-1}}\left(F_{\overline{n-1}} + F_{\underline{n-1}}\right) + \frac{2}{p_{n-1}} = \left(\prod_{i=2}^{n-1}\frac{q_i}{p_i}\right)\cdot\frac{2}{p_1} + \sum_{j=2}^{n-1}\left(\prod_{i=j+1}^{n-1}\frac{q_i}{p_i}\right)\cdot\frac{2}{p_j}.$$

It is left to obtain the upper bound for $E_{\underline{n-1}}$. For this we start by rearranging the terms in (6.22) as follows:

$$\begin{cases} (p_1+d_1)E_{\overline{1}} - d_1E_{\underline{1}} = p_1E_{\overline{2}} + 1, \\ (p_1+d_1)E_{\underline{1}} - d_1E_{\overline{1}} = p_1E_{\underline{2}} + 1. \end{cases}$$

Subtracting the first equation from the second, we have $(p_1+2d_1)(E_{\underline{1}} - E_{\overline{1}}) = p_1(E_{\underline{2}} - E_{\overline{2}})$. Hence

$$E_{\underline{1}} - E_{\overline{1}} = \frac{p_1}{p_1+2d_1}(E_{\underline{2}} - E_{\overline{2}}) < E_{\underline{2}} - E_{\overline{2}}.$$

Similarly, from (6.23) we obtain a chain of inequalities

$$E_{\underline{2}} - E_{\overline{2}} < E_{\underline{3}} - E_{\overline{3}} < \cdots < E_{\underline{n-2}} - E_{\overline{n-2}} < E_{\underline{n-1}} - E_{\overline{n-1}}.$$

Finally, rearrange the terms in (6.24) as

$$\begin{cases} (p_{n-1}+q_{n-1}+d_{n-1})E_{\overline{n-1}} - d_{n-1}E_{\underline{n-1}} = p_{n-1}E_n + q_{n-1}E_{\overline{n-2}} + 1, \\ (q_{n-1}+d_{n-1})E_{\underline{n-1}} - d_{n-1}E_{\overline{n-1}} = q_{n-1}E_{\underline{n-2}} + 1. \end{cases}$$

Subtracting the first equation from the second and applying (6.25), we deduce

$$(q_{n-1}+2d_{n-1})(E_{\underline{n-1}} - E_{\overline{n-1}}) - p_{n-1}E_{\overline{n-1}} = q_{n-1}(E_{\underline{n-2}} - E_{\overline{n-2}}) < q_{n-1}(E_{\underline{n-1}} - E_{\overline{n-1}}).$$

Rearranging these terms we have $E_{\underline{n-1}} < (1 + (p_{n-1}/d_{n-1}))\, E_{\overline{n-1}}$. ■

Proof of Proposition 6.1. Like the proof for Theorem 5.2, it suffices to establish the following three conditions:

(C1) The evolution of $(x(k), s(k))$, $k \geq 0$, is a Markov chain with a finite state space;

(C2) the set $\mathscr{A}$ defined in (5.3) is an invariant set under **QA** algorithm;

(C3) for every $(x(0), 0) \notin \mathscr{A}$ there is a finite time K_a such that $\Pr\big[(x(K_a), s(K_a)) \in \mathscr{A} \mid (x(0), 0)\big] > 0$.

For an arbitrary state $x(k)$, observe in **QA** algorithm that the minimum $m(k)$ is non-decreasing and the maximum $M(k)$ non-increasing, where $m(k)$, $M(k)$ are

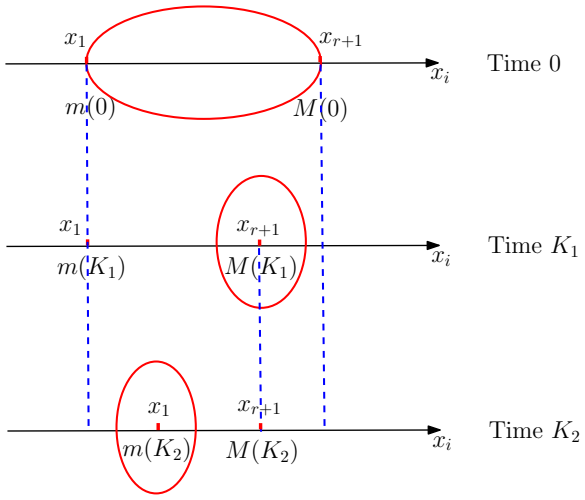


Figure 6.11: Idea of induction step

defined in (5.4). Thus the conditions **(C1)** and **(C2)** easily follow. It remains to establish **(C3)** when the digraph $\mathcal{G}$ is complete (Assumption 1), for which we proceed by induction on the number n (> 1) of nodes. Let $F(k) := M(k) - m(k)$. Assume $(x(0), 0) \notin \mathscr{A}$; then $F(0) \geq 2$.

(i) Base case: $n = 2$. Label the two nodes such that $x_1(0) = m(0)$ and $x_2(0) = M(0)$. As $\mathcal{G}$ is complete, there are two edges, $(1, 2)$ and $(2, 1)$, each of which has a positive probability to be activated. Consider the sequence of alternate activation: $(1, 2), (2, 1), (1, 2), (2, 1) \cdots$. Then in **QA** algorithm, **(R3)(i)** and **(R2)(i)** will alternately apply, thereby shrinking the interval $[m(k), M(k)]$. It is easy to see that there exist a finite time K_a and a positive probability such that $x_1(K_a) = x_2(K_a) = \lfloor (x_1(0) + x_2(0))/2 \rfloor$ (thus $(x(K_a), s(K_a)) \in \mathscr{A}$), and at most one node holds a surplus. Also in this process, $M(k)$ decreases by at least 1 and $m(k)$ increases by at least 1.

(ii) Induction step: let $r \in [2, n-1]$. Suppose that for a network of r nodes, there exist a finite time K_a and a positive probability such that $x_1(K_a) = \cdots = x_r(K_a) = \lfloor (1/r) \sum_{i=1}^{r} x_i(0) \rfloor$, and at most $r-1$ nodes each holds one surplus. Also suppose that in this process, $M(k)$ decreases by at least one and $m(k)$ increases by at least one.

Now consider the case with $r + 1$ nodes. Label them such that $m(0) = x_1(0) \leq \cdots \leq x_{r+1}(0) = M(0)$. In the sequel, we describe a sequence of activating edges, which causes the interval $[m(k), M(k)]$ to shrink, the process being displayed in Fig. 6.11.

The existence of the selected edges follows from that $\mathcal{G}$ is complete; and since each edge has a positive probability to be activated, the sequence of activation also has a positive probability.

First, consider the nodes $2, \ldots, r+1$. We distinguish three cases as follows.

Case 1: $x_{r+1}(0) - x_2(0) \geq 2$. Then applying the hypothesis, we obtain that in a finite time K_1 and with a positive probability, $x_2(K_1) = \cdots = x_{r+1}(K_1) = \lfloor (1/r) \sum_{i=2}^{r+1} x_i(0) \rfloor$.

Case 2: $x_{r+1}(0) - x_2(0) = 1$. For each node i (> 2) such that $x_i(0) - x_2(0) = 1$, activate the edge $(2, i)$; then **(R3)(i)** of **QA** algorithm applies, thereby resulting again in $x_2(K_1) = \cdots = x_{r+1}(K_1) = \lfloor (1/r) \sum_{i=2}^{r+1} x_i(0) \rfloor$.

In both cases above, the maximum state decreases as $M(K_1) < M(0)$; hence $F(K_1) < F(0)$. In addition, there are at most $r-1$ nodes each having one surplus. Activate (one at a time, in an arbitrary order) the edges connecting those nodes with a surplus to the node 1. Thus **(R2)(i)** applies, and the surpluses are consumed to increase $x_1(k)$, which in turn causes $F(k)$ to decrease. At time at most $K_1' := K_1 + r - 1$, all the surpluses in the system can be consumed.

Case 3: $x_{r+1}(0) - x_2(0) = 0$. For this special case, we go directly to the next step.

Second, consider the nodes $1, \ldots, r$. When $F(K_1') \geq 2$ (or Case 3 above), applying the hypothesis we derive that in a finite time K_2 and with a positive probability, $x_1(K_2) = \cdots = x_r(K_2) = \lfloor (1/r) \sum_{i=1}^{r} x_i(K_1') \rfloor$. Since the minimum state $m(k)$ increases by at least one, we have $F(K_2) < F(K_1')$. Also, at most $r-1$ nodes each has one surplus. Select (one at a time, in an arbitrary order) the edges connecting the node $r+1$ to those with a surplus; then **(R2)(i)** applies, and the surpluses are consumed. Note that, however, here $F(k)$ stays put. At time at most $K_2' := K_2 + r - 1$, all the surpluses in the system can be consumed. If $F(K_2') \geq 2$, we apply the hypothesis again for the nodes $2, ..., r+1$, as is done in the first step above.

Thus we can repeat these two steps, in an alternate fashion, so that $F(k)$ decreases until $F(K_a') = 1$, for some finite time K_a'. There are two possibilities: (1) $x_1(K_a') = m(K_a')$, others $m(K_a') + 1$, and at most $r-1$ nodes each has one surplus; and (2) $x_{r+1}(K_a') = M(K_a')$, others $M(K_a') - 1$, and at most $r-1$ nodes each has one surplus.

Analogous to the edge activation done above, one can show in both scenarios that there exist a finite time $K_a > K'_a$ and a positive probability such that $F(K_a) = 0$, and at most r nodes each has one surplus. Therefore necessarily, $x_1(K_a) = \cdots = x_{r+1}(K_a) = \lfloor (1/(r+1)) \sum_{i=1}^{r+1} x_i(0) \rfloor$. Finally, it is evident that in this averaging process, $M(k)$ decreases by at least one and $m(k)$ increases by at least one. This finishes the induction step. ■

Proof of Proposition 6.3. We have given in Section 6.4.2 the proof for the case $l = 1$, one step away from average consensus. It remains to establish (6.12) for every $l \in [2, V(0)/2]$. Before proceeding, we introduce the following notation for an economical representation of the transition structure in Fig. 6.8:

$$([1\ 1\ 1\ \cdots\ 1\ 1\ 0]^T,\ 1,\ 0)$$
$$\uparrow$$
$$([2\ 1\ 1\ \cdots\ 1\ 1\ 0]^T,\ 0,\ 0)$$

Here $([1\ 1\ 1\ \cdots 1\ 1\ 0]^T,\ 1,\ 0)$ represents the upper row of states $\overline{1}, \ldots, \overline{n-1}$, and $([2\ 1\ 1\ \cdots 1\ 1\ 0]^T,\ 0,\ 0)$ represents the lower row of states $\underline{1}, \ldots, \underline{n-1}$. It is well to note that the state n (i.e., the average consensus state $(\mathbf{1}, 0, 0)$) is not involved. Observe that only the triples in $\mathcal{U}_1^0$ are used, and only the triple with positive surplus $S_+ > 0$ has a transition probability to the average consensus state. We will use this notation to display the transition structures in the subsequent analysis.

(i) Two steps away: from $\mathcal{U}_2$ to $\mathcal{U}_1$. The corresponding transition structure is displayed in Fig. 6.12; there are four triples, representing four rows similar to the above. These rows can be arranged into three blocks $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ as shown. Notice that the displayed triples are all in $\mathcal{U}_2^0$, and only those triples with positive surplus $S_+ > 0$ have a transition probability to $\mathcal{U}_1$. One can readily see that starting from the triple $([3\ 1\ 1 \cdots 1\ 1\ -1]^T, 0, 0)$, the mean hitting time of $\mathcal{U}_1$ is the longest; thus we need to analyze the whole structure.

In the sequel, the structure will be simplified in two steps. First, treat the transition to $\mathcal{B}_3$ as a selfloop at the triple $([2\ 1\ 1 \cdots 1\ 1\ -1]^T, 1, 0)$ in $\mathcal{B}_2$. This modification increases the mean hitting time starting from $\mathcal{B}_1$. To see this, note that the triple in $\mathcal{B}_3$

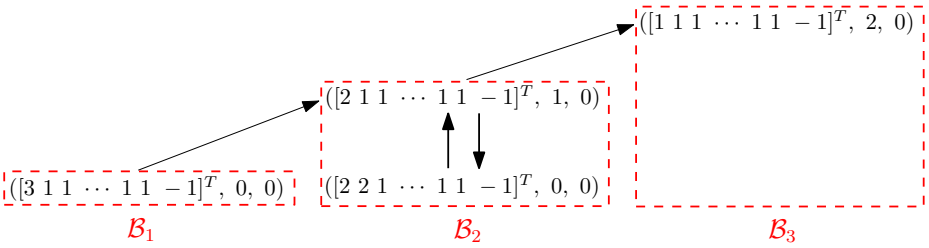


Figure 6.12: Two steps away: from $\mathcal{U}_2$ to $\mathcal{U}_1$.

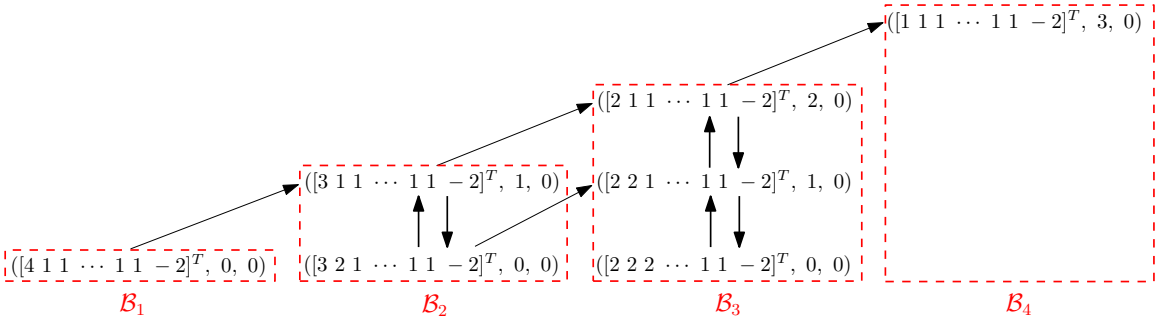


Figure 6.13: Three steps away: from $\mathcal{U}_3$ to $\mathcal{U}_2$.

has more positive surplus S_+, which results in higher probabilities of moving towards $\mathcal{U}_1$. It then follows that selflooping in $\mathcal{B}_2$ takes longer time to hit $\mathcal{U}_1$ than transiting to $\mathcal{B}_3$. Second, combine $([3\ 1\ 1\cdots 1\ 1\ -1]^T, 0, 0)$ in $\mathcal{B}_1$ and $([2\ 2\ 1\cdots 1\ 1\ -1]^T, 0, 0)$ in $\mathcal{B}_2$. This amounts to combining the corresponding two rows of triples. It can be verified that the associated transition probabilities in these two rows are the same, except for those moving to $([2\ 1\ 1\cdots 1\ 1\ -1]^T, 1, 0)$. Since the latter means moving towards $\mathcal{U}_1$, taking the smaller transition probabilities from the two rows will increases the mean hitting time.

After the above modifications, the transition structure is simplified to the one displayed in Fig. 6.5, with the following transition probabilities:

$$p_1 = (n-2)p, \quad d_1 = p; \quad p_{n-1} = p, \quad q_{n-1} = (n-2)p, \quad d_{n-1} = (n-2)p;$$

$$p_z = (n-1-z)zp, \quad q_z = (z-1)p, \quad d_z = (z-1)p \quad (z = 2, ..., n-2).$$

Hence, we have $\max_{u\in\mathcal{U}_2^0} E_2^1(u) \leq E_{\underline{n-1}}$, where $E_{\underline{n-1}}$ is given in (6.3). Invoke the formulas in Lemma 6.6, and perform an analogous calculation as before; we then obtain that $\max_{u\in\mathcal{U}_2^0} E_1^2(u) = O(n^2)$.

(ii) Three steps away: from $\mathcal{U}_3$ to $\mathcal{U}_2$. The corresponding transition structure

is displayed in Fig. 6.13; we now have four blocks. Since starting from the triple $([4\ 1\ 1\cdots 1\ 1\ -2]^T, 0, 0)$ the mean hitting time of $\mathcal{U}_2$ is the longest, we need to analyze again the whole structure.

We take three steps to simplify the structure. First, treat the transition to $\mathcal{B}_4$ as a selfloop at the triple $([2\ 1\ 1\cdots 1\ 1\ -2]^T, 2, 0)$ in $\mathcal{B}_3$. This is the same as that in (ii), and hence increases the mean hitting time starting from $\mathcal{B}_1$. Second, treat the transitions to block $\mathcal{B}_3$ as selfloops at the corresponding triples in $\mathcal{B}_2$. This modification also increases the mean hitting time. To see this, compare the structure of $\mathcal{B}_2$ and its counterpart in $\mathcal{B}_3$ (i.e., the lower two triples alone). One may verify that the former has longer rows of triples and higher probabilities of moving away from $\mathcal{U}_2$. Hence, the mean time taken to hit $\mathcal{U}_2$ in the structure of $\mathcal{B}_2$ is longer than that in its counterpart in $\mathcal{B}_3$. Further, the top triple $([2\ 1\ 1\cdots 1\ 1\ -2]^T, 2, 0)$ in $\mathcal{B}_3$, with more positive surplus S_+, makes the mean hitting time even shorter. Therefore, selflooping in $\mathcal{B}_2$ increases the mean time to hit $\mathcal{U}_2$ compared to transiting to $\mathcal{B}_3$. Lastly, combine $([4\ 1\ 1\cdots 1\ 1\ -2]^T, 0, 0)$ in $\mathcal{B}_1$ and $([3\ 2\ 1\cdots 1\ 1\ -2]^T, 0, 0)$ in $\mathcal{B}_2$, as is done in (ii).

The above simplifications lead us again to the structure displayed in Fig. 6.5, with exactly the same transition probabilities as (ii). We thus obtain $\max_{u\in\mathcal{U}_3^0} E_3^1(u) \leq E_{\underline{n-1}} = O(n^2)$.

(iii) General l (> 3) steps away: from $\mathcal{U}_l$ to $\mathcal{U}_{l-1}$. The corresponding transition structure consists of $l+1$ blocks. Apply an analogous procedure to simplify this structure; it can be found by a similar argument that transiting to further blocks will accelerate hitting $\mathcal{U}_{l-1}$. Consequently, the structure with $l+1$ blocks can also be reduced to the one in Fig. 6.5, the probabilities of which are those in (ii). Therefore, $\max_{u\in\mathcal{U}_l^0} E_l^1(u) \leq E_{\underline{n-1}} = O(n^2)$. ■

Chapter 7

Conclusions

7.1 Thesis Summary

Motivated by many natural phenomena (e.g., bird flocking and oscillator synchronization) and potential engineering applications (e.g., load balancing and sensor information fusion), we have studied two fundamental problems, consensus and averaging, in multi-agent systems. Component agents are interconnected through a network, and iteratively update their states according to a prescribed algorithm supported by the information received from neighboring peers. Our focus has been on the design of distributed algorithms. Central to solving the consensus and averaging problems is the topology of the interconnection network among agents, which determines for individual agents who their neighbors are. We have schematically represented the network topology by digraphs. Our chief goal has been to derive the most general conditions on digraphs that permit the existence of distributed algorithms solving both problems; and under those general conditions, construct provably correct solution algorithms.

First, we have dealt with the averaging problem in the basic setting where agents' states are real-valued, and networks static. We have proposed a novel distributed algorithm, whose essence is to keep local records of individual state updates, thereby ensuring consensus on the average despite that the state sum of agents is not preserved. This is achieved by augmenting an additional surplus variable for each agent. Under

this algorithm, we have derived that an arbitrary strongly connected digraph is a necessary and sufficient condition to guarantee state averaging (Theorem 3.1). This graphical condition is more general than those previously reported in the literature, in the sense that it does not require symmetric or balanced topological structures. For analysis, we have introduced certain useful tools from matrix perturbation theory to establish our results.

Second, we have tackled the averaging problem in the dynamic network model, where the interconnection topology among agents is time-varying. The time-varying mechanism can be either deterministic or random. In the random case, we have adopted that agents asynchronously gossip with one another in the sense that only a single interaction randomly occurs at a time. We have proposed distributed algorithms in both deterministic and random scenarios; these algorithms are based again on using surplus variables, and have been justified to ensure state averaging on general topology (Theorems 4.1 and 4.2). In particular, the necessary and sufficient condition, a jointly strongly connected digraph, derived for the deterministic case is worth commenting: While it is known from the literature that digraphs have to be symmetric or balanced at every moment, this condition does not require so at any moment. To derive this result, we have relied on Lyapunov-type arguments.

Third, we have studied both consensus and averaging problems in the setup where agents' states are quantized, and networks randomized in the gossip sense. To model quantization effect, we have abstracted each agent's state to be an integer. For the consensus problem, we have proposed a class of algorithms, under which it is derived that the existence of a globally reachable node in digraphs is a necessary and sufficient condition that guarantees convergence to some common value (Theorem 5.1). For the averaging problem, we have developed a counterpart quantized surplus-based algorithm; under this algorithm, we have proved again that a general strongly connected digraph is necessary and sufficient to guarantee state averaging (Theorem 5.2). For analysis, our tools have been from finite Markov chain theory. Furthermore, we have addressed the convergence time issue. Specifically, we have investigated the shrinking

Table 7.1: Summary and contrast of setups, methods, and results.

	real-valued states	quantized states
objective	$x_i(k) \to \frac{1}{n}\sum_i x_i(0)$, $s_i(k) \to 0$, as $k \to \infty$	$x_i(K) = \lfloor \frac{1}{n}\sum_i x_i(0) \rfloor$ or $\lceil \frac{1}{n}\sum_i x_i(0) \rceil$, $s_i(K)$ need not be zero, for finite K
algorithm	linear, with parameter ϵ	nonlinear, with parameter threshold
analysis	matrix eigenvalues	finite Markov chains
result	average consensus $\Longleftrightarrow$ general strongly connected topology	

time of the smallest interval that contains all states for the consensus algorithm, and the decay time of a suitable Lyapunov function for the averaging algorithm. The investigation has led us to characterizing the convergence time by the hitting time in certain special Markov chains. We have simplified the structures of state transition by considering the special case of complete networks, and derived polynomial upper bounds on convergence time (Theorems 6.1 and 6.2).

In conclusion, the thesis has mainly explored, from several different setups, a new approach to the design of distributed algorithms for achieving multi-agent average consensus: Add extra variables to keep track of state changes. The setups, methods, and results are summarized and contrasted in Table 7.1. The underlying message of the thesis is that this new approach has successfully enabled multi-agent systems to achieve average consensus on general network topology, even in the case where the state sum of agents need not stay put. This result has advanced the knowledge in the literature. Also, we are led to observe the following intuitive tradeoff. On one hand, employing extra variables one for each agent doubles the dimension of state space, thereby causing increased burden on local storage, computation, and communication of individual agents. On the other hand, employing extra variables effectively weakens the global connectivity requirement on network topology of the whole system. Therefore local burden trades off against global connectivity. In practice, it would be

up to the system designer to choose one over the other according to specific tasks at hand.

7.2 Future Research

We suggest a few topics for future research arising from this thesis.

In Chapters 3 and 4, we have designed a surplus-based algorithm for static digraphs, and a gossip algorithm for randomly time-varying digraphs. Both algorithms have been shown to ensure state averaging on general strongly connected digraphs, assuming that the parameter ϵ is sufficiently small. A general upper bound on ϵ has been provided, which is, however, conservative. Thus it is important, and of theoretic interest, to relax this assumption by deriving tight(er) bounds for the parameter. Also, in Figs. 3.7 and 4.1 we have observed similar trends of the convergence factors of both algorithms with respect to the parameter ϵ. It would be interesting to find the optimal value of ϵ which renders the speed fastest. In addition, the agent model we have adopted in these two chapters is the (discrete-time) single integrator. An extension could be to consider higher-order linear time-invariant models (e.g., higher-order integrators), or even nonlinear models (e.g., unicycles).

In Chapter 6 we have analyzed the convergence time of the quantized consensus and averaging algorithms designed in Chapter 5, and obtained polynomial upper bounds for complete graphs. One direction of future work is to extend the results to more general topologies, which would be to study potentially greater complexity of the state and surplus transition structure owing to topological constraints. An alternative approach might be to explore the relation between the convergence time bounds and the spectral properties of the Laplacian matrix associated to a given topology. More broadly speaking, seeking fast quantized consensus and averaging algorithms deserves further effort.

Finally, viewing the thesis as a whole, there are several topics potential for our future research. The first topic is time delay, which is inherent in communication

among agents. We would like to explicitly consider time delay in our model, and study its effect on convergence as well as performance. The second issue is to compute more general functions of the initial states, other than the average or linear combinations. This is motivated by that certain nonlinear functions, e.g., power mean and geometric mean, are found useful for information fusion in sensor networks. We attempt to suitably modify our algorithms to achieve that functionality. Lastly, our successful use of surplus suggests that providing augmented variables for individual agents could potentially enable the whole network to accomplish some more demanding tasks, and is therefore worth being applied to addressing other distributed control problems in multi-agent systems, such as formation and routing.

Bibliography

[1] W. Aiello, B. Awerbuch, B. Maggs, and S. Rao. Approximate load balancing on dynamic and asynchronous networks. In *Proc. 25th ACM Symposium on Theory of Computing*, pages 632–641, 1993.

[2] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robotics and Automation*, 15(5):818–828, 1999.

[3] T. C. Aysal, M. J. Coates, and M. G. Rabbat. Distributed average consensus with dithered quantization. *IEEE Trans. Signal Processing*, 56(10):4905–4918, 2008.

[4] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Verlag, 2002.

[5] B. R. Barmish. *New Tools for Robustness of Linear Systems*. Macmilllan, 1994.

[6] D. Bauso, L. Giarré, and R. Pesenti. Non-linear protocols for optimal distributed consensus in networks of dynamic agents. *Systems & Control Letters*, 55(11):918–928, 2006.

[7] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli. Weighted gossip: distributed averaging using non-doubly stochastic matrices. In *Proc. IEEE Int. Symposium on Information Theory*, pages 1753–1757, Austin, TX, 2010.

[8] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, 1994.

[9] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[10] R. Bhatia. *Matrix Analysis*. Springer Verlag, 1996.

[11] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. 44th IEEE Conf. on Decision and Control and Eur. Control Conf.*, pages 2996–3000, Seville, Spain, 2005.

[12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Inform. Theory*, 52(6):2508–2530, 2006.

[13] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Princeton University Press, 2009.

[14] K. Cai and H. Ishii. Convergence time analysis of quantized gossip algorithms on digraphs. In *Proc. 49th IEEE Conf. on Decision and Control*, pages 7669–7674, Atlanta, GA, 2010.

[15] K. Cai and H. Ishii. Further results on randomized quantized averaging: A surplus-based approach. In *Proc. IEEE Multi-Conf. on Systems and Control*, pages 1558–1563, Yokohama, Japan, 2010.

[16] K. Cai and H. Ishii. Gossip consensus and averaging algorithms with quantization. In *Proc. American Control Conf.*, pages 6306–6311, Baltimore, MD, 2010.

[17] K. Cai and H. Ishii. Quantized average consensus on gossip digraphs with reduced computation. *SICE J. Control, Measurement, and System Integration*, to appear, 2011.

[18] K. Cai and H. Ishii. Quantized consensus and averaging on gossip digraphs. *IEEE Trans. Autom. Control*, to appear, 2011.

[19] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*, 4(1):7–23, 1997.

[20] R. Carli and F. Bullo. Quantized coordination algorithms for rendezvous and deployment. *SIAM J. on Control and Optimization*, 48(3):1251–1274, 2009.

[21] R. Carli, F. Bullo, and S. Zampieri. Quantized average consensus via dynamic coding/decoding schemes. *International J. of Robust and Nonlinear Control*, 20(2):156–175, 2010.

[22] R. Carli, F. Fagnani, P. Frasca, and S. Zampieri. Gossip consensus algorithms via quantized communication. *Automatica*, 46(1):70–80, 2010.

[23] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, 2008.

[24] J. Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44(3):726–737, 2008.

[25] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. Robotics and Automation*, 20(2):243–255, 2004.

[26] O.L.V. Costa, M.D. Fragoso, and R.P. Marques. *Discrete-Time Markov Jump Linear Systems*. Springer Verlag, 2004.

[27] P. J. Davis. *Circulant Matrices*. AMS Chelsea Publishing, 1994.

[28] D. V. Dimarogonas and K. H. Johansson. Quantized agreement under time-varying communication topology. In *Proc. American Control Conf.*, pages 4376–4381, Seattle, WA, 2008.

[29] F. Fagnani and S. Zampieri. Asymmetric randomized gossip algorithms for consensus. In *Proc. 17th IFAC World Congress*, pages 9052–9056, Seoul, Korea, 2008.

[30] M. Franceschelli, A. Giua, and C. Seatzu. Consensus on the average on arbitrary strongly connected digraphs based on broadcast gossip algorithms. In *Proc. 1st IFAC Workshop on Estimation and Control of Networked Systems*, pages 66–71, Venice, Italy, 2009.

[31] M. Franceschelli, A. Giua, and C. Seatzu. Distributed averaging in sensor networks based on broadcast gossip algorithms. *IEEE Sensors J.*, 11(3):808–817, 2011.

[32] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri. Average consensus by gossip algorithms with quantized communication. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 4831–4836, Cancun, Mexico, 2008.

[33] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri. Average consensus on networks with quantized communication. *International J. of Robust and Nonlinear Control*, 19(16):1787–1816, 2009.

[34] N. M. Freris, H. Kowshik, and P. R. Kumar. Fundamentals of large sensor networks: connectivity, capacity, clocks, and computation. *Proc. IEEE*, 98(11):1828–1846, 2010.

[35] F. R. Gantmacher. *The Theory of Matrices, Vol. 2*. Chelsea, 1959.

[36] B. Ghosh and S. Muthukrishnan. Dynamic load balancing by random matchings. *J. Computer and System Sciences*, 53(3):357–370, 1996.

[37] G. Grimmett and D. Stirzaker. *Probabilitty and Random Processes*. Oxford University Press, 2001.

[38] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Trans. Inf. Theory*, 46(2):388–404, 2000.

[39] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Trans. Autom. Control*, 50(11):1867–1872, 2005.

[40] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[41] H. Ishii and R. Tempo. Distributed randomized algorithms for the pagerank computation. *IEEE Trans. Autom. Control*, 55(9):1987–2002, 2010.

[42] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents sing nearest neighbor rules. *IEEE Trans. Autom. Control*, 48(6):988–1001, 2003.

[43] E. I. Jury. A modified stability table for linear discrete systems. *Proc. IEEE*, 53(2):184–185, 1965.

[44] E. I. Jury. Modified stability table for 2-d digital filters. *IEEE Trans. Circuits and Systems*, 35(1):116–119, 1988.

[45] A. Kashyap, T. Başar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.

[46] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 482–491, Washington DC, 2003.

[47] Y. Kuramoto. *Chemical Oscillations, Waves, and Turbulence*. Springer Verlag, 1984.

[48] J. Lavaei and R. M. Murray. Quantized consensus by means of gossip algorithm. *IEEE Trans. Autom. Control*, accepted, 2011.

[49] T. Li, M. Fu, L. Xie, and J. F. Zhang. Distributed consensus with limited communication data rate. *IEEE Trans. Autom. Control*, 56(2):279–292, 2011.

[50] Z. Lin. *Distributed Control and Analysis of Coupled Cell Systems*. VDM Verlag, 2008.

[51] Z. Lin, M. Broucke, and B. A. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Trans. Autom. Control*, 49(4):622–629, 2004.

[52] Z. Lin, B. A. Francis, and M. Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Trans. Autom. Control*, 50(1):121–127, 2005.

[53] Z. Lin, B. A. Francis, and M. Maggiore. State agreement for continuous-time coupled nonlinear systems. *SIAM J. on Control and Optimization*, 46(1):288–307, 2007.

[54] J. Liu, B. D. O. Anderson, M. Cao, and A. S. Morse. Analysis of accelerated gossip algorithms. In *Proc. 48th IEEE Conf. on Decision and Control*, pages 871–876, Shanghai, China, 2009.

[55] W. Lu and T. Chen. New approach to synchronization analysis of linearly coupled ordinary differential systems. *Physica D*, 213(2):214–230, 2006.

[56] L. Moreau. Stability of multi-agent systems with time dependent communication links. *IEEE Trans. Autom. Control*, 50(2):169–182, 2005.

[57] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Trans. Autom. Control*, 54(11):2506–2517, 2009.

[58] J. R. Norris. *Markov Chains*. Cambridge University Press, 1997.

[59] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE*, 95(1):215–233, 2007.

[60] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. S. Shamma. Belief consensus and distributed hypothesis testing in sensor networks. In P. J. Antsaklis and P. Tabuada, editors, *Network Embedded Sensing and Control,* vol. 331 of *Lecture Notes in Control and Information Sciences*, pages 169–182, 2006.

[61] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49(9):1520–1533, 2004.

[62] L. M. Pecora and T. L. Carroll. Master stability functions for synchronized coupled systems. *Phys. Rev. Lett.*, 80(10):2109–2112, 1998.

[63] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer, 1994.

[64] M. Porfiri and D. J. Stilwell. Consensus seeking over random weighted directed graphs. *IEEE Trans. Autom. Control*, 52(9):1767–1773, 2007.

[65] W. Ren and R. W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control*, 50(5):655–661, 2005.

[66] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Springer Verlag, 2008.

[67] W. Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 27(2):71–82, 2007.

[68] C. W. Reynolds. Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics*, 21(4):25–34, Jul 1987.

[69] A. P. Seyranian and A. A. Mailybaev. *Multiparameter Stability Theory with Mechanical Applications*. World Scientific, 2004.

[70] J. R. Silvester. Determinants of block matrices. *The Mathematical Gazette*, 84(501):460–467, 2000.

[71] G. W. Stewart and J. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[72] S. H. Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Phys. D*, 143(1-4):1–20, 2000.

[73] K. Sugihara and I. Suzuki. Distributed motion coordination of multiple mobile robots. In *Proc. 5th IEEE Int. Symposium on Intelligent Control*, pages 138–143, Philadelphia, PA, 1990.

[74] A. Tahbaz-Salehi and A. Jadbabaie. A necessary and sufficient condition for consensus over random networks. *IEEE Trans. Autom. Control*, 53(3):791–795, 2008.

[75] R. Tempo and H. Ishii. Monte Carlo and Las Vegas randomized algorithms for systems and control: An introduction. *European J. Control*, 13:189–203, 2007.

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

[77] T. Vicsek, A. Czirok, E. Jacob, I. Cohen, and O. Schochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75(6):1226–1229, 1995.

[78] C. W. Wu. Synchronization and convergence of linear dynamics in random directed networks. *IEEE Trans. Autom. Control*, 51(7):1207–1210, 2006.

[79] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[80] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *International Conf. on Information Processing in Sensor Networks*, pages 63–70, Los Angeles, CA, 2005.

[81] M. Zhu and S. Martínez. On the convergence time of distributed quantized averaging algorithms. In *Proc. 47th IEEE Conf. on Decision and Control*, pages 3971–3976, Cancun, Mexico, 2008.