Convergence Speed in Distributed Consensus and Averaging

by

Alexander Olshevsky

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2006
[June 2006]

Author .....
Department of Electrical Engineering and Computer Science
May 25, 2006

Certified by .....
John N. Tsitsiklis
Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Convergence Speed in Distributed Consensus and Averaging

by

Alexander Olshevsky

B.S. Applied Mathematics (2004)

B.S. Electrical Engineering (2004)

Georgia Institute of Technology

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

We propose three new algorithms for the distributed averaging and consensus problems: two for the fixed-graph case, and one for the dynamic-topology case. The convergence times of our fixed-graph algorithms compare favorably with other known methods, while our algorithm for the dynamic-topology case is the first to be accompanied by a polynomial-time bound on the worst-case convergence time.

Thesis Supervisor: John N. Tsitsiklis
Title: Professor

Acknowledgments

I am grateful to my advisor, John Tsitsiklis, for his invaluable guidance and tireless efforts in supervising this thesis. I have greatly benefitted from his help, suggestions, insight, and patience.

I want to thank Prof. Vincent Blondel and Prof. Ali Jadbabaie for useful conversations that have helped me in the course of my research.

I want to thank Mukul Agarwal, Constantine Caramanis, and Aman Chawla. I have learned much from my conversations with them.

I thank my parents for all the encouragement they have given me over the years.

Finally, I want to thank Angela for her constant support.

This research was supported by the National Science Foundation under a Graduate Research Fellowship and grant ECS-0312921.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Given a set of autonomous agents — which may be sensors, nodes of a communication network, cars, or unmanned aerial vehicles — the distributed consensus problem asks for a distributed algorithm that the agents can use to agree on an opinion (represented by a scalar or a vector) starting from different initial opinions among the agents and possibly severe restrictions on communication.

Algorithms that solve the distributed consensus problem provide the means by which the networks of agents may be coordinated. Although each agent acts independently of the others, because the agents can agree on a parameter of interest, they may act in a coordinated fashion when decisions involving this parameter arise. Synchronized behavior of this sort has often been observed in biological systems [19].

The distributed consensus problem has historically appeared in many diverse areas: communication networks [29, 26]. control theory [23], and parallel computation [38. 5]. Recently, the problem has attracted significant attention [23, 26, 3, 14, 29, 10. 30. 31]; research has been driven by advances in communication theory and by new connections to open problems in networking and networked control theory. We briefly describe some more recent applications.

Reputation Management in Ad Hoc Networks: It is often the case that the

nodes of a wireless multi-hop network are not controlled by a single authority or do not have a common objective. Selfish behavior among nodes (e.g., refusing to forward traffic meant for others) is possible, and some mechanism is needed to enforce cooperation. One way to detect selfish behavior is reputation management: each node forms an opinion by observing the behavior of its neighbors. One is then faced with the problem of combining these different opinions into a single globally available reputation measure for each node. The use of distributed consensus algorithms for doing this was explored in [26], where a variation of one of the methods we examine in this thesis - the "agreement algorithm" - was used as a basis for empirical experiments.

Sensor Networks: A sensor network designed for detection or estimation will need to combine various measurements into a decision or into a single estimate. Distributed computation of this decision/estimate has the advantage of being fault-tolerant (network operation is not dependent a small set of nodes) and self-organizing (network functionality does not require constant supervision) [39, 3, 4, 14].

Control of Autonomous Agents: It is often necessary to coordinate collections of autonomous agents (e.g., cars or UAVs). For example, one may wish for the agents to agree on a direction or speed. Even though the data related to the decision may be distributed through the network, it is usually desirable that the final decision depend on all the known data, even though much of it is unavailable at many nodes. Methods for the solution of this problem were empirically investigated in [41].

We lay special emphasis on a subcase of the distributed consensus problem, the distributed *averaging* problem. While a consensus algorithm combines the measurements of the individual nodes into a global value, an averaging algorithm further guarantees that the limit will be the exact average of the individual values.

This thesis studies the convergence times of distributed algorithms for the consensus and averaging problems. Our starting point is the agreement algorithm proposed in [38] for the distributed consensus problem, which we extend to some new settings involving delays (Chapter II). We also give a proof of a result that has been computer-verified in [23]: the nonexistence of quadratic Lyapunov functions for the agreement algorithm (Chapter III). We show how the agreement algorithm may be

modified for the distributed averaging problem, in a way that avoids the potential slowdown of other methods (Chapter IV). We proceed to give a variation of the agreement algorithm that has the same worst-case convergence time as optimization-based approaches (Chapter V). We provide a new algorithm for the distributed averaging problem in symmetric dynamic topology networks (Chapter VI). Our algorithm is the first to be accompanied by a polynomial bound on the worst-case convergence time in this setting. We finish with some simulation results comparing the algorithms presented here with previous approaches (Chapter VII).

1.2 Previous work

The performance of distributed algorithms for consensus and averaging has been analyzed by a number of authors in the past. For results proving convergence of such algorithms, see:

- DeGroot [12] analyzed the agreement algorithm in the context of social networks. He proved a convergence result in the static-topology case.
- Tsitsiklis [38], and Tsitsiklis, Bertsekas, and Athans [39], formulated the agreement algorithm for distributed consensus in the context of distributed computing. For a summary of this research, see the monograph [5].
- Vicsek et al [41] independently conducted some simulations with the agreement algorithm for coordination of motion of autonomous particles. This inspired the work of Jadbabaie, Lin, and Morse [23], where a convergence result was again proved.
- Olfati-Saber and Murray [32] proved the convergence of a variation of the agreement algorithm for the distributed averaging problem (previous results only covered the distributed consensus problem). See also the survey [33].

Some recent work has focused on extending convergence results to a wider class of algorithms:

- Blondel, Hendrickx, Olshevsky, and Tsitsiklis [6] prove a convergence result for the case of positive delays during the execution of the algorithm and under an "approximate" assumption of symmetry.

- Moreau [30] proved a convergence result for a general class of iterative algorithms which includes the agreement algorithm (and others).

- Angeli and Bliman [1] extended Moreau's result to cover the case when delays occur during the execution of the algorithm.

- Tanner, Jabdabaie, and Pappas [40] proved convergence for a variation of the agreement algorithm which achieves coordination for a system of moving particles, along with collision avoidance.

- Cucker and Smale [9] proved convergence for a continuous averaging scheme which does not require any assumptions on the interconnection topology.

- The survey by Fang and Antsaklis [16] contains more information on recent research in this field.

Some recent work has focused on convergence time of consensus and averaging algorithms, which is also our focus. In the case of static graphs:

- Xiao and Boyd [43] proposed a convex optimization approach to the design of fast-converging algorithms in fixed networks.

- Boyd et al. [3, 4] analyze the convergence time of a randomized scheme for averaging.

- Dimakis, Sarwate, and Wainwright propose and analyze an algorithm for averaging in geometric random graphs [14].

In the case of dynamic graphs, there are some results on the convergence time of the agreement algorithm for the consensus problem:

- Cao. Spielman, and Morse prove an exponential upper bound for consensus in the case of dynamic graphs [10].

- The Ph.D. thesis of D. Hernek contains some doubly exponential bounds for the cover time of a class of random walks on colored graphs [22]. There is a close relationship between such random walks and the agreement algorithm, and the bounds in [22] may be translated into the language of multi-agent coordination literature to imply that the agreement algorithm on a certain class of graphs will take at most doubly exponential time to converge.

Further, there has been some recent work on the extension of *averaging* algorithms to the dynamic topology case:

- Mehyar et al. [29] propose an algorithm for averaging in the dynamic-topoly case.

- Moallemi and Van Roy propose another extension to the dynamic graph case, the "consensus propagation" algorithm [31]. In contrast to the agreement algorithm, consensus propagation provides an *approximation* to the average for certain classes of *time-varying graphs* (recall that the agreement algorithm can compute the average *exactly*, as shown in section 4.1, in *fixed* graphs). However, in contrast to the load balancing algorithm presented in Chapter 6 of this thesis, the class of graphs for which consensus propagation has been shown to converge is not large enough to include all time-varying graph sequences.

  The "consensus propagation" algorithm is patterned on belief propagation: nodes pass messages that are supposed to represent their estimates of the global mean, and these estimates are accompanied by companion messages that represent their accuracy. Indeed, consensus propagation may be viewed as a special case of a belief propagation algorithm on an appropriately set-up estimation problem [31].

Finally, we make note of some related work on multi-agent systems:

- Some recent work [36], [28] has focused on characterizing graphs that arise from interactions of autonomous agents in R^2 or R^3. These results have potential implications for the consensus literature. In this thesis, as in other work in this area, we do not restrict the set of graphs that can arise as a result of nearest-neighbour interactions between agents. However, the Euclidean nature of R^2 and R^3 provides some restrictions, which may result in sharper bounds.

- A variation of the agreement algorithm was proposed in [34] for computing geographical coordinates of nodes in a sensor network.

- A distributed algorithm for optimal territorial coverage with mobile sensing networks based on a gradient descent scheme was proposed in [8].

Our contribution: We add to this literature in three ways. First, we propose a distributed algorithm for averaging in fixed graphs which has some attractive features relative to previously known methods. Second, we give some bounds on the performance of optimization-based approached in fixed graphs. Finally, we give a polynomial-time averaging method for dynamic graphs.

Chapter 2

The agreement algorithm

The "agreement algorithm," due to Tsitsiklis et al [38], is an iterative procedure for the solution of the distributed consensus problem. In this section, we describe and analyze the agreement algorithm (see [38], [39], [23] for original literature). We begin with a review of prior work in Section 2.1, where we give the basic background and a summary of known results for the agreement algorithm. In Section 2.2, we explain a connection between the convergence rate of the agreement algorithm and the joint spectral radius. In Section 2.3, we discuss extensions of the agreement algorithm that incorporate the presence of delays.

2.1 Introduction

We consider a set $N = \{1, 2, \ldots, n\}$ of agents embedded, at each time t, in a directed graph $G(t) = (N, \mathcal{E}(t))$, where t lies in some discrete set of times which we will take, for simplicity, to be the nonnegative integers.

Each agent i starts with a scalar value $x_i(0)$; the vector with the values of all agents at time t will be denoted by $x(t) = (x_1(t), \ldots, x_n(t))$. The agreement algorithm updates $x(t)$ according to the equation $x(t+1) = A(t)x(t)$, or

$$x_i(t+1) = \sum_{j=1}^{n} a_{ij}(t)x_j(t).$$

where $A(t)$ is a nonnegative matrix with entries $a_{ij}(t)$. The row-sums of $A(t)$ are equal to 1, so that $A(t)$ is a stochastic matrix. In particular, $x_i(t+1)$ is a weighted average of the values $x_j(t)$ held by the agents at time t.

We next state some conditions under which the agreement algorithm is guaranteed to converge.

Assumption 1. There exists a positive constant α such that:

(a) $a_{ii}(t) \geq \alpha$, for all i, t.

(b) $a_{ij}(t) \in \{0\} \cup [\alpha, 1]$, for all i, j, t.

(c) $\sum_{j=1}^{n} a_{ij}(t) = 1$, for all i, t.

Intuitively, whenever $a_{ij}(t) > 0$, agent j communicates its current value $x_j(t)$ to agent i. Each agent i updates its own value, by forming a weighted average of its own value and the values it has just received from other agents.

In terms of the directed graph $G(t) = (N, \mathcal{E}(t))$, we introduce an arc $(j,i) \in \mathcal{E}(t)$ if and only if $a_{ij}(t) > 0$. Note that $(i,i) \in \mathcal{E}(t)$ for all t. A minimal assumption, which is necessary for consensus to be reached and for each agent to have an effect on the final value, requires that following an arbitrary time t, and for any i, j, there is a sequence of communications through which agent i will influence (directly or indirectly) the value held by agent j.

Assumption 2. (Connectivity) The graph $(N, \cup_{s \geq t} \mathcal{E}(s))$ is strongly connected for all $t \geq 0$.

We note various special cases of possible interest.

Time invariant model: There is a fixed matrix A, with entries a_{ij}, such that, for each t, we have $a_{ij}(t) = a_{ij}$.

Symmetric model: If $(i,j) \in \mathcal{E}(t)$ then $(j,i) \in \mathcal{E}(t)$. That is, whenever i communicates to j, there is a simultaneous communication from j to i.

Equal neighbor model: Here,

$$a_{ij}(t) = \begin{cases} 1/d_i(t), & \text{if } j \in N_i(t), \\ 0, & \text{if } j \notin N_i(t), \end{cases}$$

where $N_i(t) = \{j \mid (j,i) \in \mathcal{E}(t)\}$ is the set of agents j whose value is taken into

account by i at time t, and $d_i(t)$ is its cardinality. This model is a linear version of a model considered by Vicsek et al. [41]. Note that here the constant α of Assumption 1 can be take to be $1/n$.

Assumption 3. (Bounded intercommunication times) There is some B such that $(N, \mathcal{E}(kB) \cup \mathcal{E}(kB+1) \cup \cdots \cup \mathcal{E}((k+1)B-1))$ is strongly connected for all integer k.

Theorem 1. Under Assumptions 1, 2 (connectivity), and 3 (bounded intercommunication times), the agreement algorithm guarantees asymptotic consensus.

Theorem 1 is presented in [39] and is proved in [38] (under a slightly different version of Assumption 3); a simplified proof, for the special case of fixed coefficients can be found in [5]. It subsumes several subsequent convergence results that have been presented in the literature for special cases of the model. On the other hand, in the presence of symmetry, the bounded intercommunication times assumption is unnecessary. The latter result is proved in [21], [24], [7] and in full generality in [30].

Theorem 2. Under Assumptions 1 and 2, and for the symmetric model, the agreement algorithm guarantees asymptotic consensus.

See [6], [39], [5] for extensions to the cases of communication delay and probabilistic dropping of packets.

2.2 Products of stochastic matrices and convergence rate

Theorem 1 and 2 can be reformulated as results on the convergence of products of stochastic matrices.

Corollary 1. Consider an infinite sequence of stochastic matrices $A(0), A(1), A(2), \ldots$, that satisfies Assumptions 1 and 2. If either Assumption 3 (bounded intercommunication intervals) is satisfied, or if we have a symmetric model, then there exists a

nonnegative vector d such that

$$\lim_{t\to\infty} A(t)A(t-1)\cdots A(1)A(0) = \mathbf{1}d^T.$$

(Here, $\mathbf{1}$ is a column vector whose elements are all equal to one.)

According to Wolfowitz's Theorem ([42]) convergence occurs whenever the matrices are all taken from a finite set of ergodic matrices, and the finite set is such that any finite product of matrices in that set is again ergodic. Corollary 1 extends Wolfowitz's theorem by not requiring the matrices $A(t)$ to be ergodic, though it is limited to matrices with positive diagonal entries.

The presence of long matrix products suggests that convergence to consensus in the linear iteration

$$x(t+1) = A(t)x(t),$$

with $A(t)$ stochastic, might be characterized in terms of a joint spectral radius. The joint spectral radius $\rho(M)$ of a set of matrices M is a scalar that measures the maximal asymptotic growth rate that can be obtained by forming long products of matrices taken from the set M:

$$\rho(M) = \limsup_{k\to\infty} \sup_{M_{i_1}, M_{i_2}, \ldots, M_{i_k} \in M} ||M_{i_1} M_{i_2} \ldots M_{i_k}||^{1/k}.$$

This quantity does not depend on the norm used. Moreover, for any $q > \rho(M)$ there exists a C for which

$$||M_{i_k} \ldots M_{i_1} y|| \leq C q^k ||y||,$$

for all y and $M_{i_j} \in M$.

Stochastic matrices satisfy $||Ax||_\infty \leq ||x||_\infty$ and $A\mathbf{1} = \mathbf{1}$, and so they have a spectral radius equal to one. The product of two stochastic matrices is again stochastic and so the joint spectral radius of any set of stochastic matrices is equal to one. To analyze the convergence rate of products of stochastic matrices, we consider the dynamics induced by the matrices on a space of smaller dimension.

Consider a matrix $P \in \Re^{(n-1)\times n}$ defining an orthogonal projection on the space orthogonal to span$\{\mathbf{1}\}$. We have $P\mathbf{1} = 0$, and $||Px||_2 = ||x||_2$ whenever $x^T\mathbf{1} = 0$. Associated to any $A(t)$, there is a unique matrix $A'(t) \in \Re^{(n-1)\times(n-1)}$ that satisfies $PA(t) = A'(t)P$. The spectrum of $A'(t)$ is the spectrum of $A(t)$ after removing one multiplicity of the eigenvalue 1. Let M' be the set of all matrices $A'(t)$.

Let $\gamma = \mathbf{1}^T x(t)/n$ be the mean value of the entries of $x(t)$, then

$$\begin{aligned} Px(t) - P\gamma\mathbf{1} &= Px(t) \\ &= PA(t)A(t-1)\dots A(0)x(0) \\ &= A'(t)A'(t-1)\dots A'(0)Px(0). \end{aligned}$$

Since $(x(t) - \gamma\mathbf{1})^T\mathbf{1} = 0$, we have

$$||x(t) - \gamma\mathbf{1}||_2 = ||P(x(t) - \gamma\mathbf{1})||_2 \leq Cq^t||x(0)||_2,$$

for some C and for any $q > \rho(M')$.

Assume now that $\lim_{t\to\infty} x(t) = c\mathbf{1}$ for some scalar c. Because all matrices are stochastic, c must belong to the convex hull of the entries of $x_i(t)$ for all t. We therefore have

$$||x(t) - c\mathbf{1}||_\infty \leq 2||x(t) - \gamma\mathbf{1}||_\infty \leq 2||x(t) - \gamma\mathbf{1}||_2,$$

and we may then conclude that

$$||x(t) - c\mathbf{1}||_\infty \leq 2Cq^t||x(0)||_2.$$

The joint spectral radius $\rho(M')$ therefore gives a measure of the convergence rate of $x(t)$ towards its limit value $c\mathbf{1}$. However, for this bound to be nontrivial, all of the matrices in M need to be ergodic; indeed, in the absence of an ergodicity condition, the convergence of $x(t)$ need not be geometric, and will depend in general on the particular sequence of elements of M. Indeed, if $\hat{A} \in M$ is not ergodic, then $\hat{A}^t x(t)$ may not converge to $c\mathbf{1}$ at all.

2.3 Convergence in the presence of delays.

The model considered so far assumes that messages from one agent to another are immediately delivered. However, in a distributed environment, and in the presence of communication delays, it is conceivable that an agent will end up averaging its own value with an *outdated* value of another processor. A situation of this type falls within the framework of distributed asynchronous computation developed in [5].

Communication delays are incorporated into the model as follows: when agent i, at time t, uses the value x_j from another agent, that value is not necessarily the most recent one, $x_j(t)$, but rather an outdated one, $x_j(\tau_j^i(t))$, where $0 \leq \tau_j^i(t) \leq t$, and where $t - \tau_j^i(t))$ represents communication and possibly other types of delay. In particular, $x_i(t)$ is updated according to the following formula:

$$x_i(t+1) = \sum_{j=1}^{n} a_{ij}(t) x_j(\tau_j^i(t)). \tag{2.1}$$

We make the following assumption on the $\tau_j^i(t)$.

Assumption 4. (Bounded delays) (a) If $a_{ij}(t) = 0$, then $\tau_j^i(t) = t$.
(b) $\lim_{t\to\infty} \tau_j^i(t) = \infty$, for all i, j.
(c) $\tau_i^i(t) = t$, for all i, t.
(d) There exists some $B > 0$ such that $t - B + 1 \leq \tau_j^i(t) \leq t$, for all i, j, t.

Assumption 4(a) is just a convention: when $a_{ij}(t) = 0$, the value of $\tau_j^i(t)$ has no effect on the update. Assumption 4(b) is necessary for any convergence result: it requires that newer values of $x_j(t)$ get eventually incorporated in the updates of other agents. Assumption 4(c) is quite natural, since an agent generally has access to its own most recent value. Finally, Assumption 4(d) strenghtens Assumption 4(b) by requiruing delays to be bounded by some constant B,

The next result, from [38, 39], is a generalization of Theorem 1. The proof is similar to the proof of Theorem 1: we define $m(t) = \min_i \min_{s=t,t-1,\ldots,t-B+1} x_i(s)$ and $M(t) = \max_i \max_{s=t,t-1,\ldots,t-B+1} x_i(s)$ and show that the difference $M(t) - m(t)$ decreases by a constant factor after a bounded amount of time.

Theorem 3. Under Assumptions 1-4 (connectivity, bounded intercommunication intervals, and bounded delays), the agreement algorithm with delays [cf. Eq. (2.1)] guarantees asymptotic consensus.

Theorem 3 assumes bounded intercommunication intervals and bounded delays. The example that follows (Example 1.2, in p. 485 of [5]) shows that Assumption 4(d) (bounded delays) cannot be relaxed. This is the case even for a symmetric model, or the further special case where $\mathcal{E}(t)$ has exactly four arcs (i,i), (j,j), (i,j), and (j,i) at any given time t, and these satisfy $a_{ij}(t) = a_{ji}(t) = 1/2$, as in the pairwise averaging model.

Example 2. We have two agents who initially hold the values $x_1(0) = 0$ and $x_2(0) = 1$, respectively. Let t_k be an increasing sequence of times, with $t_0 = 0$ and $t_{k+1} - t_k \to \infty$. If $t_k \leq t < t_{k+1}$, the agents update according to

$$\begin{aligned} x_1(t+1) &= (x_1(t) + x_2(t_k))/2, \\ x_2(t+1) &= (x_1(t_k) + x_2(t))/2. \end{aligned}$$

We will then have $x_1(t_1) = 1 - \epsilon_1$ and $x_2(t_1) = \epsilon_1$, where $\epsilon_1 > 0$ can be made arbitrarily small, by choosing t_1 large enough. More generally, between time t_k and t_{k+1}, the absolute difference $|x_1(t) - x_2(t)|$ contracts by a factor of $1 - 2\epsilon_k$, where the corresponding contraction factors $1 - 2\epsilon_k$ approach 1. If the ϵ_k are chosen so that $\sum_k \epsilon_k < \infty$, then $\prod_{k=1}^{\infty}(1 - 2\epsilon_k) > 0$, and the disagreement $|x_1(t) - x_2(t)|$ does not converge to zero.

According to the preceding example, the assumption of bounded delays cannot be relaxed. On the other hand, the assumption of bounded intercommunication intervals can be relaxed, in the presence of symmetry, leading to the following generalization of Theorem 2, which is a new result.

Theorem 4. Under Assumptions 1, 2 (connectivity), and 4 (bounded delays), and for the symmetric model, the agreement algorithm with delays [cf. Eq. (2.1)] guarantees asymptotic consensus.

Proof. Let

$$\begin{aligned} M_i(t) &= \max\{x_i(t), x_i(t-1), \ldots, x_i(t-B+1)\}, \\ M(t) &= \max_i M_i(t), \\ m_i(t) &= \min\{x_i(t), x_i(t-1), \ldots, x_i(t-B+1)\}, \\ m(t) &= \min_i m_i(t). \end{aligned}$$

An easy inductive argument, as in p. 512 of [5], shows that the sequences $m(t)$ and $M(t)$ are nondecreasing and nonincreasing, respectively. The convergence proof rests on the following lemma.

Lemma 1: If $m(t) = 0$ and $M(t) = 1$, then there exists a time $\tau \geq t$ such that $M(\tau) - m(\tau) \leq 1 - \alpha^{nB}$.

Given Lemma 1, the convergence proof is completed as follows. Using the linearity of the algorithm, there exists a time τ_1 such that $M(\tau_1) - m(\tau_1) \leq (1-\alpha^{nB})(M(0) - m(0))$. By applying Lemma 1, with t replaced by τ_{k-1}, and using induction, we see that for every k there exists a time τ_k such that $M(\tau_k) - m(\tau_k) \leq (1-\alpha^{nB})^k(M(0) - m(0))$, which converges to zero. This, together with the monotonicity properties of $m(t)$ and $M(t)$, implies that $m(t)$ and $M(t)$ converge to a common limit, which is equivalent to asymptotic consensus. **q.e.d.**

Proof of Lemma 1: For $k = 1, \ldots, n$, we say that "Property P_k holds at time t" if there exist at least k indices i for which $m_i(t) \geq \alpha^{kB}$.

We assume, without loss of generality, that $m(0) = 0$ and $M(0) = 1$. Then, $m(t) \geq 0$ for all t, because of the monotonicity of $m(t)$. Furthermore, there exists some i and some $\tau \in \{-B+1, -B+2, \ldots, 0\}$ such that $x_i(\tau) = 1$. Using the inequality $x_i(t+1) \geq \alpha x_i(t)$, we obtain $m_i(\tau + B) \geq \alpha^B$. This shows that there exists a time at which property P_1 holds.

We continue inductively. Suppose that $k < n$ and that Property P_k holds at some time t. Let S be a set of cardinality k containing indices i for which $m_i(t) \geq \alpha^{kB}$, and let S^c be the complement of S. Let τ be the first time, greater than or equal to t,

at which $a_{ij}(\tau) \neq 0$, for some $j \in S$ and $i \in S^c$ (i.e., an agent j in S gets to influence the value of an agent i in S^c). Such a time exists by the connectivity assumption (Assumption 2).

Note that between times t and τ, the agents ℓ in the set S only form convex combinations between the values of the agents in the set S (this is a consequence of the symmetry assumption). Since all of these values are bounded below by α^{kB}, it follows that this lower bound remains in effect, and that $m_\ell(\tau) \geq \alpha^{kB}$, for all $\ell \in S$.

For times $s \geq \tau$, and for every $\ell \in S$, we have $x_\ell(s+1) \geq \alpha x_\ell(s)$, which implies that $x_\ell(s) \geq \alpha^{kB}\alpha^B$, for $s \in \{\tau+1, \ldots, \tau+B\}$. Therefore, $m_\ell(\tau+B) \geq \alpha^{(k+1)B}$, for all $\ell \in S$.

Consider now an agent $i \in S^c$ for which $a_{ij}(\tau) \neq 0$. We have

$$x_i(\tau+1) \geq a_{ij}(\tau)x_j(\tau_j^i(\tau)) \geq \alpha m_i(\tau) \geq \alpha^{kB+1}.$$

Using also the fact $x_i(s+1) \geq \alpha x_i(s)$, we obtain that $m_i(\tau+B) \geq \alpha^{(k+1)B}$. Therefore, at time $\tau+B$, we have $k+1$ agents with $m_\ell(\tau+B) \geq \alpha^{(k+1)B}$ (namely, the agents in S, together with agent i). It follows that Property P_{k+1} is satisfied at time $\tau+B$.

This inductive argument shows that there is a time τ at which Property P_n is satisfied. At that time $m_i(\tau) \geq \alpha^{nB}$ for all i, which implies that $m(\tau) \geq \alpha^{nB}$. On the other hand, $M(\tau) \leq M(0) = 1$, which proves that $M(\tau) - m(\tau) \leq 1 - \alpha^{nB}$. **q.e.d.**

The symmetry condition $[(i,j) \in \mathcal{E}(t)$ iff $(j,i) \in \mathcal{E}(t)]$ used in Theorem 4 is somewhat unnatural in the presence of communication delays, as it requires perfect synchronization of the update times. A looser and more natural assumption is the following.

Assumption 5. There exists some $B' > 0$ such that whenever $(i,j) \in \mathcal{E}(t)$, then there exists some τ that satisfies $|t-\tau| < B'$ and $(j,i) \in \mathcal{E}(\tau)$.

Assumption 5 allows for protocols such as the following. Agent i sends its value to agent j. Agent j responds by sending its own value to agent i. Both agents update their values (taking into account the received messages), within a bounded time from receiving the other agent's value. In a realistic setting, with unreliable

communications, even this loose symmetry condition may be impossible to enforce with absolute certainty. One can imagine more complicated protocols based on an exchange of acknowledgments, but fundamental obstacles remain (see the discussion of the "two-army problem" in pp. 32-34 of [2]). A more realistic model would introduce a positive probability that some of the updates are never carried out. (A simple possibility is to assume that each $a_{ij}(t)$, with $i \neq j$, is changed to a zero, independently, and with a fixed probability.) The convergence result that follows remains valid in such a probabilistic setting (with probability 1). Since no essential new insights are provided, we only sketch a proof for the deterministic case.

Theorem 5. Under Assumptions 1, 2 (connectivity), 4 (bounded delays), and 5, the agreement algorithm with delays [cf. Eq. (2.1)] guarantees asymptotic consensus.

Proof. A minor change is needed in the proof of Lemma 1. In particular, we define P_k as the event that there exist at least k indices l for which $m_l(t) \geq \alpha^{k(B+B')}$. It follows that P_1 holds at time $t = B + B'$.

By induction, let P_k hold at time t, and let S be the set of cardinality k containing indices l for which $m_l(t) \geq \alpha^{k(B+B')}$. Furthermore, let τ be the first time after time t that $a_{ij}(\tau) \neq 0$ where exactly one of i, j is in S. Along the same lines as in the proof of Lemma 1, $m_l(\tau) \geq \alpha^{k(B+B')}$ for $l \in S$; since $x_l(t+1) \geq \alpha x_l(t)$, it follows that $m_l(\tau + B + B') \geq \alpha^{(k+1)(B+B')}$ for each $l \in S$. By our assumptions, exactly one of i,j is in S^c. If $i \in S^c$, then $x_i(\tau + 1) \geq a_{ij}(\tau) x_j(\tau_j^i(\tau)) \geq \alpha^{k(B+B')+1}$ and consequently $x_i(\tau + B + B') \geq \alpha^{B+B'-1}\alpha^{k(B+B')+1} = \alpha^{(k+1)(B+B')}$. If $j \in S^c$, then there must exist a time $\tau_j \in \{\tau + 1, \tau + 2, \ldots, \tau + B' - 1\}$ with $a_{ji}(\tau_j) > 0$. It follows that:

$$\begin{aligned} m_j(\tau + B + B') &\geq \alpha^{\tau+B+B'-(\tau_j+1)} x_j(\tau_j + 1) \\ &\geq \alpha^{\tau+B+B'-\tau_j-1} \alpha x_i(\tau_j) \\ &\geq \alpha^{\tau+B+B'-\tau_j-1} \alpha \alpha^{\tau_j-\tau} \alpha^{k(B+B')} \\ &= \alpha^{(k+1)(B+B')}. \end{aligned}$$

Therefore, P_{k+1} holds at time $\tau + B + B'$ and the induction is complete. **q.e.d.**

Remark: It is easy to see that Theorem 5 provides a bound on the convergence rate of the agreement algorithm. Indeed, it has been shown that $M(k(B+B')) - m(k(B+B')) \leq (1-\alpha^{nk(B+B')})(M(0)-m(0))$, showing that the sequence sampled at integral multiples of $B+B'$ converges geometrically with rate α^n. In the case of nearest-neighbour model, $\alpha = 1/n$, so that the convergence rate is $1/n^n$. A similar result was proved in [10], where a rate of convergence on the order of $1/n^n$ was also derived.

Chapter 3

Nonexistence of Quadratic Lyapunov Functions

Convergence results for the agreement algorithm typically show a decrease in the "span" $\max_i x_i(t) - \min_i x_i(t)$. However, bounds on the "span" typically give exponential bounds on the rate of convergence - see the previous chapter and [10]. A natural question, therefore, is whether other Lyapunov functions might give improved results.

In [23], this question was investigated for quadratic Lyapunov functions. A computer search showed they do not exist for the agreement algorithm in the fixed nearest-neighbour regime. The goal of this chapter is to give a proof of this fact.

Consider the iteration $x_{t+1} = A_{G(t)}x(t)$ where $A_{G(t)}$ is the matrix corresponding to an equal-neighbour iteration on the graph G (see Section II for definitions). A quadratic Lyapunov function is a function $L(x)$ of the form $L(x) = x^T Q x$ for some symmetric non-negative definite Q. The function $L(x)$ must be nonincreasing after each iteration, i.e. $L(x(t+1)) \leq L(x)$ and must achieve its minimum of 0 on the subspace $\alpha\mathbf{1}$. Then, if it is possible to show that $L(x)$ strictly decreases often enough, a convergence result in the spirit of Theorem 1 of Section II would follow.

The following theorem shows that such a function $L(x)$ cannot exist.

Theorem: Let $n \geq 12$. There does not exist a symmetric, nonnegative definite and nonzero matrix Q such that:

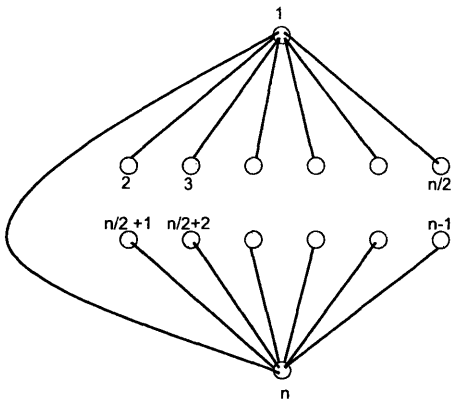

Figure 3-1: The nodes perform an iteration of the nearest-neighbor model with this graph. Every node has a self-loop which is not shown in this picture.

1.

$$(A_G(t)x(t))^T Q(A_G(t)x(t)) \leq x^T Q x, \tag{3.1}$$

for all connected graphs G on n vertices.

2.

$$\mathbf{1}^T Q \mathbf{1} = 0,$$

where $\mathbf{1}$ is the column vector of all ones.

Idea of proof: Before jumping into the proof, let us briefly describe the main idea.

1. Suppose we want to show that the sample variance, defined by, $\sigma^2(t) = \sum_{i=1}^n (x_i(t) - \bar{x}(t))^2$, where $\bar{x}(t) = (1/n)\sum_{i=1}^n x_i(t)$, is not a Lyapunov function. Consider an initial vector $x(0)$ defined by $x_1(0) = 1$, $x_n(0) = -1$, $x_k(0) = 0$ for $k = 2, \ldots, n-1$. For simplicity, let us assume that n is even. Consider the outcome of the equal neighbour iteration on the graph of Figure 3-1 (note that the graph does not show self-loops, which are actually present at each vertex). After the iteration, we will have $x_1(1) = \frac{1}{n/2+1}(1 + (-1) + 0 + \cdots + 0) = 0$ and similarly $x_n(1) = 0$. However, $x_k(1) = (1/2)(1 + 0) = 1/2$ for $k = 2, \ldots, n/2$ and, similarly, $x_k(1) = -1/2$ for $i = n/2 + 1, \ldots, n-1$.

We now see that the sample variance increased from time 0 to time 1: $\sigma^2(0) = 2$ while $\sigma^2(1) = (1/4)(n-2)$ so that for $n > 10$, $\sigma^2(0) < \sigma^2(1)$. This shows that the sample variance cannot be a Lyapunov function.

2. Let us now show that no diagonal Lyapunov function is possible. A diagonal Lyapunov function is of the form $L(x) = x^T D x$, where $D = \text{diag}(d_{11}, d_{22}, \ldots, d_{nn})$. It must be that $L(x(0)) \geq L(x(1))$ for the $x(0)$ and $x(1)$ we have computed in item 1. Writing this out explicitly,

$$d_{11} + d_{nn} \geq \frac{1}{4} \sum_{k \in \{1,2,\ldots,n\} - \{1,n\}} d_{kk}.$$

We next derive a similar equation for the sum $d_{II} + d_{JJ}$ where I now is any index in $\{1, \ldots, n/2\}$ and J is any index in $\{n/2+1, \ldots, n\}$. Recall that our choice of $x(0)$ and the starting graph of Figure 3-1 is completely arbitrary. For any $I \in \{1, \ldots, n/2\}$ and $J \in \{n/2+1, \ldots, n\}$, we pick an $x(0)$ defined by $x_I(0) = 1$, $x_J(0) = -1$, and $x_k(0) = 0$ for $k \notin \{I, J\}$. We pick a corresponding graph by switching vertices 1 with I and n with J in the graph of Figure 3-1. Then, the analog of the above equation is,

$$d_{II} + d_{JJ} \geq \frac{1}{4} \sum_{k \in \{1,2,\ldots,n\} - \{I,J\}} d_{kk}. \tag{3.2}$$

In other words, $d_{II} + d_{JJ}$ must be large relative to $\sum_{k \in \{1,2,\ldots,n\} - \{I,J\}} d_{kk}$. But this must hold for all d_{II} and d_{JJ}! We will derive a contradiction from this fact. Indeed, let us sum Eq. (3.2) over the $(n/2)^2$ ways to pick $I \in \{1, \ldots, n/2\}$ and $J \in \{n/2+1, \ldots, n\}$.

(a) Let $i \in \{1, \ldots, n\}$, and let us count how many times d_{ii} appears when we sum the left hand side of Eq. (3.2) over all possible choices $I \in \{1, \ldots, n/2\}$ and $J \in \{n/2+1, \ldots, n\}$. We claim that d_{ii} appears exactly $n/2$ times. Indeed, if $i \leq n/2$, then d_{ii} will appear when $I = i$; there are precisely $n/2$ such choices - exactly one choice of I and any of the $n/2$ possibilities for

J. Similarly, $i \geq n/2+1$, then d_{ii} will appear when $J = i$; there are $n/2$ such choices as well, exactly one for J and any of the $n/2$ possibilities for I.

(b) Similarly, each d_{ii} will appear $(n/2)(n/2-1)$ times on the right. Indeed, if $i \leq n/2$, then d_{ii} will appear on the right if $I \neq i$; there are $(n/2)(n/2-1)$ such choices. Similarly, if $i \geq n/2$, then d_{ii} will appear on the right if $J \neq i$, which happens $(n/2)(n/2-1)$ times.

Thus, performing the summation, we have,

$$(n/2)\text{tr}(D) \geq \frac{1}{4}(n/2)(n/2-1)\text{tr}(D).$$

Because D is nonnegative definite, we have that $d_{ii} \geq 0$ for all i. Therefore, because D is nonzero, we have that tr(D) > 0. Canceling $(n/2)\text{tr}(D)$ from both sides,

$$1 \geq \frac{1}{4}(n/2-1),$$

which is a contradiction when $n > 10$.

We next generalize this proof to the case of arbitrary quadratic Lyapunov functions.

Proof:

1. Using the notation $Q = [q_{ij}]_{i,j=1,\ldots,n}$, we can rewrite the requirement $\mathbf{1}^T Q \mathbf{1} = 0$ as,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij} = 0.$$

2. Once again, we begin with the vector $x(0)$ defined by $x_1(0) = 1$, $x_n(0) = -1$, $x_i(0) = 0$ for $i = 2, \ldots, n-1$. After an equal-neighbor model iteration on the

graph G of Figure 3-1 [1], the result is $x_1(1) = 0$, $x_n(1) = 0$, $x_2(1), \ldots, x_{n/2}(1) = 1/2$, $x_{n/2+1}(1), \ldots, x_{n-1}(1) = -1/2$. By Eq. 3.1, we obtain

$$x_0^T Q x_0 \geq x_1^T Q x_1.$$

Writing this out in terms of the elements of the matrix Q,

$$\begin{aligned} q_{11} + q_{nn} - 2q_{1n} \quad \geq \quad & \frac{1}{4} \cdot [\sum_{i,j \in \{2,\ldots,n/2\}} q_{ij} + \sum_{i,j \in \{n/2+1,\ldots,n-1\}} q_{ij} \\ & -2 \sum_{i \in \{1,\ldots,n/2\}-\{1\}, j \in \{n/2+1,\ldots,n\}-\{n\}} q_{ij}]. \end{aligned}$$

3. However, our choice to assign initial values of $x_i(0) = 1$ to nodes 1 and n was completely arbitrary. Let us say we pick nodes I, J instead with $x_I(0) = 1$ and $x_J(0) = -1$ and disjoint sets A_I, A_J each containing $n/2$ elements with $I \in A_I$ and $J \in A_J$. We assign $x_k = 0$ for $k \notin \{I, J\}$. Then, the analog of the above inequality is:

$$q_{II} + q_{JJ} - 2q_{IJ} \geq \frac{1}{4} (\sum_{i,j \in A_I - \{I\}} q_{ij} + \sum_{i,j \in A_J - \{J\}} q_{ij} - 2 \sum_{i \in A_I - \{I\}, j \in A_J - \{J\}} q_{ij}). \quad (3.3)$$

4. Let us sum this equation over the $\binom{n}{n/2}$ ways to select the two sets A_I and A_J and the $(n/2)^2$ ways to select I and J once A_I, A_J are given.

   (a) To perform this summation, we note some basic combinatorial identities. Given nodes i, j the number of partitions $\{A_I, A_J\}$ of $\{1, \ldots, n\}$ where i, j are in the same set (A_I or A_J) is $2\binom{n-2}{n/2-2}$. The number of partitions $\{A_I, A_J\}$ where one of i, j is in A_I and another is in A_J is $2\binom{n-2}{n/2-1}$.

[1] Formally, this is the undirected graph on n vertices made of the edges $\{(1,i) \mid i = 2, \ldots, n/2\}$, $\{(n,i) \mid i = n/2+1, \ldots, n-1\}$, and $\{(1,n)\}$, and self-loops $\{(i,i) \mid i = 1, \ldots, n\}$

(b) Let us sum the left hand side, which is $q_{II} + q_{JJ} - 2q_{IJ}$, over all possible ways to choose A_I, A_J, I, J.

i. Each choice of A_I, A_J may be followed by $(n/2)\cdot(n/2)$ possible choices of $I \in A_I, J \in A_J$. This means each term q_{ii} appears $n/2$ times for each choice of A_I, A_J (indeed, i will be in one of the sets - say it is set A_I - and every choice of $j \in A_J$ will contribute a q_{ii} to the sum).

ii. Let us now count how many times the term q_{ij} appears in the sum when $i \neq j$. Note that the symmetry of the matrix Q implies $2q_{IJ} = q_{IJ}+q_{JI}$ so that we can write $q_{II}+q_{JJ}-2q_{IJ} = q_{II}+q_{JJ}-q_{IJ}-q_{JI}$. Using the latter representation, we can see that the term q_{ij} with $i \neq j$ appears exactly once with a negative sign in every choice of A_I, A_J where i, j are not in the same set.

iii. Therefore, the left hand side sums to

$$\binom{n}{n/2}(n/2)\mathrm{tr}(Q) - 2\binom{n-2}{n/2-1}\sum_{i\neq j} q_{ij}.$$

(c) Similarly, let us consider how often each entry of the matrix Q appears when we sum the right hand side of Eq. (3.3). The right hand side of Eq. (3.3) is

$$\frac{1}{4}\Big(\sum_{i,j\in A_I-\{I\}} q_{ij} + \sum_{i,j\in A_J-\{J\}} q_{ij} - 2\sum_{i\in A_I-\{I\}, j\in A_J-\{J\}} q_{ij}\Big)$$

Now each q_{ii} is included with a sign of $+1$ for every distinct choice of A_I, A_J, I, J when $i \neq I$ and $i \neq J$. There are $\binom{n}{n/2}(n/2-1)(n/2)$ such choices. On the other hand, q_{ij} appears with a coefficient of $+1$ for every choice of I, J, A_I, A_J where i, j are in the same set and $\{i,j\}\bigcap\{I,J\} = \emptyset$, and -1 for every choice where i, j are in a different set and $\{i,j\}\bigcap\{I,J\} = \emptyset$. Thus, the number of times q_{ij} appears with a coefficient of $+1$ is $2\binom{n-2}{n/2-2}(n/2-2)(n/2)$, while the number of times it appears with a coefficient of -1 is $2\binom{n-2}{n/2-1}(n/2-1)^2$.

Therefore, the right hand side sums to

$$\frac{1}{4}[\binom{n}{n/2}(n/2-1)(n/2)\mathrm{tr}(Q)+(2(n/2-2)(n/2)\binom{n-2}{n/2-2}$$
$$-2(n/2-1)^2\binom{n-2}{n/2-1})\sum_{i\neq j}q_{ij}].$$

Putting this all together, we have the inequality:

$$\begin{aligned}\binom{n}{n/2}(n/2)\mathrm{tr}(Q)-2\binom{n-2}{n/2-1}\sum_{i\neq j}q_{ij} \;\geq\; & \frac{1}{4}[\binom{n}{n/2}(n/2-1)(n/2)\mathrm{tr}(Q)+\\ & +(2(n/2-2)(n/2)\binom{n-2}{n/2-2}\\ & -2(n/2-1)^2\binom{n-2}{n/2-1})\sum_{i\neq j}q_{ij}].\end{aligned}$$

5. But since $\sum_{i,j}q_{ij}=0$, it follows that for real numbers α,β it holds that $\alpha\mathrm{tr}(Q)-\beta\sum_{i\neq j}q_{ij}=(\alpha+\beta)\mathrm{tr}(Q)$. Therefore, we can rewrite the above inequality as:

$$\begin{aligned}\left(\binom{n}{n/2}(n/2)+2\binom{n-2}{n/2-1}\right)\mathrm{tr}(Q) \;\geq\; & \frac{1}{4}\binom{n}{n/2}(n/2-1)(n/2)\\ & -(2(n/2-2)(n/2)\binom{n-2}{n/2-2}\\ & -2(n/2-1)^2\binom{n-2}{n/2-1})\mathrm{tr}(Q).\end{aligned}$$

If $tr(Q)=0$ then $Q=0$ since Q is nonnegative definite. Else, canceling $\mathrm{tr}(Q)$ from both sides and plugging in $n=12$, we get $6048\geq 7560$, which is a contradiction.

Chapter 4

Averaging with the agreement algorithm in fixed networks

The agreement algorithm is not guaranteed to converge to the average. However, a variation of the agreement algorithm which does converge to the average has been proposed in [32]. In this chapter, we design another variation of the agreement algorithm for the solution of the averaging problem. Our algorithm avoids a small step-size, which is required in [32] for convergence.

We will show that in the case of fixed networks, a single extra parallel pass of the agreement algorithm is sufficient for the agreement algorithm to compute averages. Moreover, this pass need not be repeated if opinions change but the network remains the same. We describe the details below.

4.1 Using two parallel passes of the agreement algorithm

Given a fixed graph G, define the matrix A by $a_{ij} = 1/d(i)$ for all $j \in N_i$, where d_i is the cardinality of $N_i = \{j \mid (j,i) \in \mathcal{E}\}$. Consider the iteration $p \to pA$ where p is some row vector with elements summing to one. Since A is the transition matrix of the random walk on the graph, and since the presence of self-loops makes the period

of the chain equal to one, we conclude that the iteration converges to the row vector π defined by $\pi_i = d_i/E$, where $E = \sum_i d_i$ is the total number number of edges, including self-loops. Thus, A^t converges to a matrix whose rows are all equal to π.

Consider next the iteration $x \rightarrow Ax$, which is just the averaging algorithm under the equal neighbor model on G. It follows from the discussion in the previous paragraph that

$$\lim_t x_i(t) = \sum_{i=1}^{n} \pi_i x_i = \sum_{i=1}^{n} d_i x_i / E. \tag{4.1}$$

The algorithm for computing the average is as follows. The nodes perform two parallel iterations of the equal-neighbor averaging algorithm. At the first iteration the initial value at the ith node is $1/d_i$. The result, which we will denote by $z = n/E$ will at the end be available at every node. The second iteration (performed in parallel with the first) sets the initial value at the ith node to $x_i(0)/d_i$. The result of this iteration is then divided by z. The final result is

$$\frac{E}{n} \sum_{i=1}^{n} \frac{d_i}{E} \frac{x_i(0)}{d_i} = \frac{1}{n} \sum_{i=1}^{n} x_i(0),$$

which is the desired average. Note that the value of n need not be known by the nodes in order to execute this iteration.

4.2 Comparison with other results

The following averaging method was proposed in [32]. The nodes first agree on some value $\epsilon \in (0, \min_i 1/d_i)$. This can be done by, for example, computing the largest degree $d_{\max}$ and agreeing on $1/(2d_{\max})$. The largest degree could be computed by a simple iterative algorithm: at each step, each node takes the maximum of its own degree and that of each of its neighbors. After at most n steps, each node knows the highest degree.

Next, the nodes run the iteration $x_i(t+1) = (1-\epsilon d_i)x_i(t) + \epsilon \sum_{j \in N(i)-\{i\}}^{n} x_j(t)$. This iteration converges to consensus (using either a Markov chain argument, as above, or as a special case of Theorem 1). In addition, the sum $\sum_i x_i(t)$ remains constant,

which implies that we have convergence to the exact average.

Compared with the algorithm proposed in this chapter, the algorithm of [32] has the disadvantage of uniformly small step sizes. If many of the nodes have degrees of the order of n, there is no significant theoretical difference between the two algorithms, as both have effective step sizes of $1/n$. On the other hand, if only a small number of nodes have $O(n)$ degrees, then the algorithm in [32] will force *all* the nodes to take small steps, in contrast to the algorithm proposed here. The advantages of the algorithm presented here will be illustrated through simulation experiments in Section 7.1 of Chapter 7.

Chapter 5

Estimates of the Convergence Time

In this chapter, we analyze the convergence time of certain time-invariant variants of the agreement algorithm. We also propose a simple heuristic with a favorable worst case convergence time.

Let X be the set of vectors whose components are all equal. The measure of convergence that we consider in this chapter is the convergence rate defined as

$$\rho = \sup_{x(0) \notin X} \lim_{t \to \infty} \Big(\frac{\|x(t) - x^*\|_2}{\|x(0) - x^*\|_2} \Big)^{1/t}, \tag{5.1}$$

where we denote $x^* = \lim_{t \to \infty} x(t)$.

The following proposition is well-known.

Proposition 1: Consider the agreement algorithm with a fixed-coefficient matrix A satisfying Assumptions 1 and 2 (so that A has an eigenvalue at 1, with multiplicity equal to 1). Then, $\rho = \max |\lambda|$, where the maximum is taken over all eigenvalues λ of A different than 1. If the matrix A is symmetric, then $\rho = \max\{|\lambda_2|, |\lambda_n|\}$, where the λ_i are the (real) eigenvalues of A, sorted in decreasing order.

5.1 Convergence time for the equal neighbor time invariant model

For the equal neighbor time invariant model, a bound on the convergence rate was computed in [27].

Theorem 3. ([27]) The convergence rate for the equal-neighbor time invariant model, for any graph on n vertices, satisfies

$$\rho \leq 1 - n^{-3}.$$

Moreover, there exists some $\gamma > 0$ such that for every $n \in Z^+$ there exists an n-node graph for which

$$\rho \geq 1 - \gamma n^{-3}.$$

Let us define the convergence time $T_n(\epsilon)$ as the first time such that $\frac{||x(t)-x^*||_\infty}{||x(0)-x^*||_\infty} \leq \epsilon$ for $t \geq T_n(\epsilon)$. Although in principle one could use any norm to define convergence time, the infinity norm is particularly suited for the consensus problem. Indeed, the final goal of all of the algorithms described in this thesis is to get a set of agents to agree on a value. Thus, the proper measure of performance is how far the agents are from agreement, i.e. $\max_i x_i(t) - \min_i x_i(t)$. Since this is most naturally bounded in terms of the infinity norm, $\max_i x_i(t) - \min_i x_i(t) \leq 2||x(t) - x(\infty)||_\infty$, we focus our analysis on the infinity norm. We note that bounds for other norms may be easily obtained from the results presented here by the equivalence of norms.

Then, the above theorem has the following corollary:

Corollary 2. The convergence time for the equal-neighbor time invariant model, for any graph on n vertices, satisfies

$$T_n(\epsilon) \leq (3/4)n^3 \log n + (1/2)n^3 \log \frac{2}{\epsilon}.$$

Proof: Let A be the coefficient matrix of an equal-neighbor time invariant model on some fixed graph. Consider a random walk on this graph which starts at node i and

moves to each of its neighbors with probability $\frac{1}{d(i)}$ (note that the random walk may remain at i because of self-loops). Note that the one-step transition probability of this random walk is A. It is known that for any random walk on a graph (Theorem 5.1 of [25]),

$$|P_j(t) - \pi_j| \leq \sqrt{\frac{d_j}{d_i}}\rho^t. \tag{5.2}$$

where $P_j(t)$ is the probability of being at node j after t steps, π_j is the stationary probability of node j, and ρ is as defined in the previous section, i.e. $\max\{\lambda_2(A), \lambda_n(A)\}$.

Since $1 \leq d_i$ and $n \geq d_j$, we can further write

$$|P_j(t) - \pi_j| \leq \sqrt{n}\rho^t.$$

and using the result of Theorem 3,

$$|P_j(t) - \pi_j| \leq \sqrt{n}(1 - n^{-3})^t.$$

This implies that for $t \geq (3/4)n^3 \log n + (1/2)n^3 \log \frac{2}{\epsilon}$, we have $|P_j(t) - \pi_j| \leq \frac{\epsilon}{2n}$ for all j.

Now $A^\infty = \lim_{t\to\infty} A^t = \pi^{\mathbf{T}}\mathbf{1}$ (see Chapter 4.1 for a proof of this). Thus for $t \geq (3/4)n^3 \log n + (1/2)n^3 \log \frac{2}{\epsilon}$ we have that

$$A^t = A^\infty + E^{(\epsilon)},$$

where $|E_{ij}^{(\epsilon)}| \leq \frac{\epsilon}{2n}$ for all i, j. Thus,

$$x(t) - x(\infty) = A^t x(0) - A^\infty x(0) = E^{(\epsilon)} x(0).$$

so that

$$||x(t) - x(\infty)||_\infty \leq \frac{\epsilon}{2n}||x(0)||_1$$

$$\begin{aligned} &\leq \frac{\epsilon}{2}||x(0)||_\infty \\ &\leq \epsilon||x(0) - x(\infty)||_\infty, \end{aligned}$$

where the last inequality follows because the entries of $x(\infty)$ lie in the convex hull of the entries of $x(0)$. **q.e.d.**

5.2 Symmetric eigenvalue minimization via convex optimization

5.2.1 Problem description

Xiao and Boyd [43] considered the following problem. Given a connected, undirected graph $G = (V, \mathcal{E})$, minimize ρ over all symmetric, stochastic matrices A that satisfy $a_{ij} = 0$ whenever $(j, i) \notin \mathcal{E}$ and $\mathbf{1}^T A = \mathbf{1}^T$. This problem arises when a network designer wishes to choose the weights a_{ij} to maximize the convergence rate, subject to the constraints induced by a certain communication topology defined by the graph. For example, the nodes may correspond to sensors which are located somewhere in R^2 or R^3. A communication constraint may be imposed by the requirement that sensors can communicate only provided they are within a certain distance of each other.

The requirement $\mathbf{1}^T A = \mathbf{1}^T$ enforces the conservation law

$$\mathbf{1}^\mathbf{T} x(t+1) = \mathbf{1}^T A x(t) = \mathbf{1}^T x(t),$$

thus ensuring that the average $(1/n)\sum_{i=1}^n x_i(t)$ is preserved throughout the execution of the algorithm. For any matrix A satisfying these constraints,

$$\lim_t x(t) = \lim_t A^t x(0) = \frac{1}{n}\mathbf{1}\mathbf{1}^\mathbf{T} x(0) = \left(\frac{1}{n}\sum_{i=1}^n x(0)\right)\mathbf{1},$$

so that the final consensus is the average of the initial values. The symmetry requirement is introduced in [43] in order to obtain a tractable optimization problem.

5.2.2 Relaxing the constraints

The averaging problem may still be solved even if the constraint $\mathbf{1}^T A \neq \mathbf{1}^T$ is not satisfied and the matrix A is not symmetric. We now describe the details of a two-step procedure for averaging in these conditions.

Provided that the graph $(V, \{(i,j)|a_{ji} > 0\})$ is connected, the Markov chain with probability transition matrix A will have a stationary distribution π (Chapter 6.6 of [18]) and

$$\lim_t A^t = \pi^T \mathbf{1} = \frac{1}{n} n\pi^T \mathbf{1}.$$

Because π is a left-eigenvector of the matrix A, it can be efficiently computed by the network designer. Moreover, it is easy to see that the connectivity of the graph implies $\pi_i > 0$ for all i.

The two-step procedure for averaging is then as follows. As before, the designer communicates to the nodes the coefficients $\{a_{ij}\}$; now, however, the designer also communicates $n\pi_i$ to node i. In the first step, the nodes of the network scale their initial value as

$$x_{\text{new}}(0) = \frac{x(0)}{n\pi_i}, \tag{5.3}$$

and execute the algorithm with the vector x_{new}. The final outcome is

$$\lim_t x(t) = \lim_t A^t x_{\text{new}}(0) = \left(\sum_{i=1}^n \frac{1}{n} n\pi_i \frac{x_i(0)}{n\pi_i} \right) \mathbf{1} = \left(\frac{1}{n} \sum_{i=1}^n x_i(0) \right) \mathbf{1}, \tag{5.4}$$

which is the vector of initial averages.

5.2.3 Our contribution

In this section, we seek to obtain bounds on ρ for connected Markov chains on finite graphs. As we have just seen, such Markov chains may be used to solve the distributed averaging problem. By obtaining bounds on ρ for such chains, we obtain bounds on the speed of a class of distributed averaging algorithms.

We derive bounds which do not assume symmetry or double stochasticity of the

matrix A. We show that ρ may approach zero for arbitrary chains of this type. We use this to motivate a constraint which will result in a class of numerically good algorithms, and bound ρ for the class of chains which satisfy this constraint. We then provide an extended discussion of the convergence time bounds that result from this approach. Finally, we show that a much simpler way of choosing the coefficients results in the same worst-case convergence time.

5.2.4 Convergence rate on a class of spanning trees

We first describe an elementary way to pick a Markov chain on any graph with ρ arbitrarily close to zero.

Given an undirected graph $G = (V, \mathcal{E})$, let $T = (V, \mathcal{E}_T)$ be a spanning tree of G. Pick any vertex of this tree, and designate it to be the root; we will call this vertex r. We will use $N_T(i)$ to refer to the neighbors of node i in T.

Consider a Markov chain on the nodes of G with the following transition probabilities: $P(i,j) = 1$ if j is the parent of i in T (i.e. j is on the path from i to the root), and $P(r,r) = 1$. All other transition probabilities are set to zero. This chain takes at most n steps to converge to its stationary distribution; it follows from the definition of ρ (see Eq. (5.1)) that $\rho(P) = 0$.

Next, we will approximate this (reducible) chain with a sequence of irreducible Markov chains. Consider the following probability assignment: for any nonroot vertex i, $P(i,j) = 1 - \epsilon|N_T(i)|$ if j is the parent of i in T, $P(i,j) = \epsilon$ if j is the child of i on the tree, and $P(i,i) = \epsilon$. For the root r, we define $P(r,r) = 1 - \epsilon|N_T(r)|$ and $P(r,j) = \epsilon$ where $j \in N_T(r)$. Since eigenvalues are continuous functions of matrix elements, in the limit as $\epsilon \to 0$ we have that

$$\rho \to 0$$

Thus, picking ϵ small, we can get a Markov chain with ρ arbitrarily small. This Markov chain can be used to solve the distributed averaging problem as described in Section 5.2.2.

Remark: Picking the above Markov chain possesses some serious drawbacks. As $\epsilon \rightarrow 0$, we will have that $\pi_i \rightarrow 0$ where i is any nonroot vertex. Consequently, $n\pi_i$ also approaches 0. It follows that in Eq. (5.3) and Eq. (5.4) the algorithm divides and multiplies by an arbitrarily small number. Such operations may lead to numerical inaccuracy [20]. In the next section, we investigate the convergence times of algorithms for which $\max_{i,j} \frac{\pi_i}{\pi_j} \leq C$ for some constant $C > 1$. For such a class of Markov chains,

$$\frac{1}{C} \leq n\pi_i \leq C, \quad \text{for all } i,$$

and thus such a class of algorithms allows us to avoid the numerical inaccuracy that arises when some components of $n\pi$ approach 0.

5.2.5 Bounding the second eigenvalue

Our goal is to bound ρ for all Markov chains with n states satisfying $\max_{i,j} \frac{\pi_i}{\pi_j} \leq C$ for some constant $C > 1$. To this end, we first introduce a simple way to bound the second eigenvalue of a **reversible** Markov chain with transition probability matrix A. The method outlined here is a variation on the results in [27].

Let π_i be the stationary probability of state i, and let $D_\pi = \text{diag}(\pi_1, \ldots, \pi_n)$. The reversibility of the Markov chain can now be stated as

$$D_\pi A = A^T D_\pi. \tag{5.5}$$

Moreover, if we define the inner product $< x, y >_\pi = x^T D_\pi y$, then A is a self-adjoint operator:

$$< x, Ay >_\pi = x^T D_\pi A y = x^T A^T D_\pi y =< Ax. y >_\pi .$$

Define $S_\pi = \{< x, \mathbf{1} >_\pi = 0, < x, x >_\pi = 1\}$. Since the largest eigenvalue is 1 with an eigenvector of $\mathbf{1}$ (the vector with all components equal to 1), we use the variational characterization of the eigenvalues of a self-adjoint matrix (Chapter 7, Theorem 4.3

of [37]):

$$\begin{aligned}\lambda_2 &= \max_{x \in S} < x, Ax >_\pi \\ &= \max_{x \in S_\pi} \sum_i \sum_j a_{ij} \pi_i x_i x_j \\ &= \frac{1}{2} \max_{x \in S} \sum_i \sum_j a_{ij} \pi_i (x_i^2 + x_j^2 - (x_i - x_j)^2).\end{aligned}$$

For $x \in S_\pi$ we have $\sum_i \sum_j a_{ij} \pi_i (x_i^2 + x_j^2) = 2 \sum_i \pi_i x_i^2 = 2 < x, x >_\pi = 2$, which yields

$$\lambda_2 = 1 - \frac{1}{2} \min_{x \in S_\pi} \sum_i \sum_j a_{ij} \pi_i (x_i - x_j)^2. \tag{5.6}$$

In particular,

$$\lambda_2 \geq 1 - \frac{1}{2} \sum_i \sum_j a_{ij} \pi_i (y_i - y_j)^2, \qquad \forall y \in S_\pi. \tag{5.7}$$

Moreover, note that we can multiply both sides of Eq. (5.5) by some number K to obtain that A is self-adjoint with respect to the inner product

$$< x, x >_{K\pi} = x^T K D_\pi x$$

and thus we may analogously conclude

$$\lambda_2 \geq 1 - \frac{1}{2} \sum_i \sum_j a_{ij} K \pi_i (y_i - y_j)^2, \qquad \forall y \in S_{K\pi} \tag{5.8}$$

5.2.6 Eigenvalue optimization on the line

First, we develop some notation we will use to refer to line graphs. Let $L_{a,b}$ with $a < 0 < b$ be the graph with nodes $\{a, a+1, \ldots, b\}$ and edges (i, j) for every pair i, j that satisfies $|i - j| = 1$ (see Figure 5-1 for a picture).

Lemma 2. Let A be the transition matrix of a reversible Markov chain on $L_{-n,n}$ whose stationary distribution satisfies

Figure 5-1: The line $L_{a,b}$.

$$\max_{i,j} \frac{\pi_i}{\pi_j} \leq C, \tag{5.9}$$

Then

$$\rho \geq 1 - \frac{3C^2}{(n-2)^2}.$$

Proof: Let us choose $K = n$ so that

$$\frac{1}{n} \sum_{i=1}^{n} K\pi_i = 1$$

We will use the notation $\pi'_i = K\pi_i$. Define $\pi'_{\max}$ to be $\max_i \pi'_i$. By Eq. (5.9), we will have that $\pi'_{max} \leq C$.

To bound ρ using Eq. (5.7), we will use the vector $y_k = k\pi'_{\max}\alpha_n/\pi'_k n^{1.5}$, where α_n is a positive number chosen so that $< y, y >_\pi = \sum_{k=-n}^{n} \pi'_k y_k^2 = 1$ (computing the sum explicitly, we see that $\alpha_n^2 \leq 3/2$, for any n). Further, $< y, \mathbf{1} >_{\pi'} = \sum_{i=-n,\ldots,n} \pi'_i y_i = 0$. Therefore,

$$1 - \lambda_2 \leq \frac{1}{2} \sum_i \sum_j a_{ij}\pi'_i \frac{(\pi'_{max})^2}{(\pi'_i)^2}(y_i - y_j)^2 \leq \frac{1}{2} C^2 \sum_i \sum_j \frac{\alpha_n^2}{n^3} \leq \frac{3C^2}{4n^2},$$

Adjusting the coefficient in front of the $1/n^2$ for graph size, we obtain

$$\rho \geq 1 - \frac{3C^2}{(n-1)^2}.$$

Note that the line $L_{-n,n}$ has an odd number of vertices. If the number of vertices is even, we can simply duplicate the zero vertex and repeat the above argument. We get:

$$\rho \geq 1 - \frac{3C^2}{(n-2)^2}.$$

q.e.d.

Theorem 4. For every n, there exists a symmetric graph with n nodes, such that every irreducible Markov chain on the graph which satisfies

$$\max_{i,j} \frac{\pi_i}{\pi_j} \leq C$$

also satisfies

$$\rho \geq 1 - \frac{3C^2}{(n-2)^2}. \tag{5.10}$$

Proof: Let P by the probability transition matrix of the Markov chain on the line. By irreducibility, we must have $P(i, i+1) > 0$ and $P(i, i-1) > 0$ for all i. But such a Markov chain is reversible, because at all times,

$$|\text{ number of transitions from } i \text{ to } i+1 - \text{ number of transitions from } i+1 \text{ to } i| \leq 1$$

and thus,

$$\begin{aligned} \pi_i P(i, i+1) &= \lim_t \frac{\text{number of transitions from } i \text{ to } i+1 \text{ up to time } t}{t} \\ &= \lim_t \frac{\text{number of transitions from } i+1 \text{ to } i \text{ up to time } t}{t} \\ &= \pi_{i+1} P(i+1, i) \end{aligned}$$

so that the chain is reversible. Thus the bound from Lemma 2 applies. **q.e.d.**.

Corollary 3. For all integer n, there exists a graph with n nodes such that when the two-step scheme of Section 5.2.2 is applied with an irreducible, stochastic matrix A with stationary probability satisfying,

$$\max_{i,j} \frac{\pi_i}{\pi_j} \leq C$$

Then, for n large enough, the convergence time can be bounded as,

$$T_n(\epsilon) \geq \frac{1}{6C^2}(n-2)^2 \log \frac{1}{\epsilon}.$$

Proof: First, we note that we need to take n large so that the bound on ρ from Eq. (5.10) is less than 1. For simplicity, we will actually take n to be large enough so that this bound is less than $1/2$, i.e.

$$1 - \frac{3C^2}{(n-2)^2} \leq \frac{1}{2}$$

Let the graph be the line on n vertices constructed in Theorem 4. Let v_ρ be an eigenvector corresponding to the eigenvalue ρ (recall that $\rho = \max\{|\lambda_2|, |\lambda_n|\}$). Take $x(0)$ to be $x_i(0) = n\pi_i v_\rho$. Because $A^t x_{\text{new}}(0) = A^t v_\rho = \rho^t v_\rho$ which approaches zero as $t \to \infty$, we have that $x_{new}(\infty) = 0$. Moreover,

$$\frac{||x_{\text{new}}(t) - x_{\text{new}}(\infty)||_2}{||x_{\text{new}}(0) - x_{\text{new}}(\infty)||_2} = \rho^t.$$

So that,

$$\begin{aligned}
T_n(\epsilon) &= \log_\rho \epsilon \\
&\geq \log_{1-\frac{3C^2}{(n-2)^2}} \epsilon \\
&= \frac{\log \frac{1}{\epsilon}}{-\log[1 - \frac{3C^2}{(n-2)^2}]} \\
&\geq \frac{\log \frac{1}{\epsilon}}{2\frac{3C^2}{(n-2)^2}} \\
&= \frac{1}{6C^2}(n-2)^2 \log \frac{1}{\epsilon}.
\end{aligned}$$

q.e.d.

5.3 Convergence time for spanning trees

Theorem 5. The convergence rate of the equal neighbor time invariant model consensus algorithm on a bidirectional spanning tree with n nodes satisfies

$$\rho \leq 1 - \frac{1}{3n^2}.$$

Proof: In [27], it was proved that if each node i in a symmetric graph has degree d_i, then the second eigenvalue is real and satisfies

$$\lambda_2 = 1 - \frac{1}{2} \min_{\sum_i d_i x_i = 0, \sum_i d_i x_i^2 = 1} \sum_i \sum_j (x_i - x_j)^2. \tag{5.11}$$

We use the methods of [27] to show that for trees, λ_2 can be upper bounded by $1 - 1/n^2$. Indeed, suppose that x satisfies $\sum_i d_i x_i = 0$ and $\sum_i d_i x_i^2 = 1$, and let $x_{\max}$ be such that $|x_{\max}| = \max_i |x_i|$. For a tree, we have $\sum_i d_i = 2(n-1) + n < 3n$, where the extra factor of n is due to the self-loops. Thus,

$$1 = \sum_i d_i x_i^2 \leq 3n x_{\max}^2,$$

and it follows that $|x_{max}| \geq 1/\sqrt{3n}$. Without loss of generality, assume $x_{\max} > 0$ (else, replace each x_i by $-x_i$). Since $\sum_i d_i x_i = 0$, there exists some i for which $x_i < 0$; let us denote such a negative x_i by x_{neg}. Then,

$$\begin{aligned} \frac{1}{\sqrt{3n}} \leq x_{\max} - x_{\text{neg}} &= (x_{\max} - x_{k_1}) + (x_{k_2} - x_{k_3}) + \cdots \\ &\quad + (x_{k_{r-1}} - x_{\text{neg}}), \end{aligned}$$

where $k_1, k_2, \ldots, k_{r-1}$ are the nodes on the path from x_{max} to x_{neg}. Then, by Cauchy's inequality,

$$\frac{1}{3n} \leq \frac{n}{2} \sum_{(i,j) \in \mathcal{E}} (x_i - x_j)^2, \tag{5.12}$$

where $\mathcal{E}$ is the set of directed edges in the tree. (The factor of $1/2$ in the right-hand side arises because the sum includes both terms $(x_{k_i} - x_{k_{i+1}})^2$ and $(x_{k_{i+1}} - x_{k_i})^2$.)

Thus,

$$\sum_{(i,j)\in\mathcal{E}} (x_i - x_j)^2 \geq \frac{2}{3n^2}.$$

which proves the bound for the second largest eigenvalue.

For the smallest eigenvalue, we have that if G is not bipartite, the following is true [27]

$$\lambda_n = -1 + \min \sum_{(i,j)\in\mathcal{E}} (x_i + x_j)^2.$$

Analogously to Eq. (5.12), we can write

$$\begin{aligned} \frac{1}{\sqrt{3n}} \leq x_{max} - x_{neg} &= (x_{max} + x_{k_1}) - (x_{k_1} + x_{k_2}) + \\ &\quad (x_{k_2} + x_{k_3}) - \cdots - (x_{k_{r-1}} + x_{neg}), \end{aligned}$$

where we make sure that the last sign is negative by making sure that the length of the path from x_{max} to x_{neg} is even by utilizing self-loops if necessary. We can use this to obtain a bound similar to Eq. (5.12):

$$\frac{1}{3n} \leq \frac{n}{2} \sum_{(i,j)\in\mathcal{E}} (x_i + x_j)^2, \tag{5.13}$$

which proves that

$$\lambda_n \geq -1 + \frac{1}{3n^2},$$

thus proving the theorem. **q.e.d.**

Corollary 4. The convergence time of the equal-neighbor model on trees satisfies

$$T_n(\epsilon) \leq (9/4)n^2 \log n + (3/2)n^2 \log \frac{2}{\epsilon}.$$

Proof: Repeat the proof of Corollary 2 with the above-derived bound on ρ. The new bound on the quantity P_t (see proof of Corollary 2 for definitions) is

$$|P_t(j) - \pi(j)| \leq \sqrt{n}(1 - \frac{1}{3}n^{-2})^t,$$

which implies that we need $t \geq (9/4)n^2 \log n + (3/2)n^2 \log \frac{2}{\epsilon}$ to have $|P_t(j) - \pi(j)| \leq \frac{\epsilon}{2n}$. The rest of the proof proceeds without alteration. **q.e.d.**

Example 1: We give an example to demonstrate that the bounds of Theorem 5 and Corollary 4 are essentially tight. Consider the line on n vertices, i.e. the graph with vertex set $\{1, \ldots, n\}$ and edges (i, j) if $|i - j| = 1$. Then, letting $d(i)$ denote the degree of node i, we have that [25]

$$\frac{\pi_i}{\pi_j} = \frac{d(i)}{d(j)} \leq 2$$

In Theorem 4, we showed that if one is free to choose the edge weights a_{ij}, subject to the constraint $\max_{i,j} \pi_i/\pi_j \leq C$, then $\rho \geq 1 - (3C^2)/(n-2)^2$. Since here $C = 2$, it follows that

$$\rho \geq 1 - \frac{12}{(n-2)^2}$$

Then, analogously to the proof of Corollary 3, we have that the convergence time of the equal-neighbour model on the line (for n large enough) satisfies

$$T_n(\epsilon) \geq \frac{(n-2)^2}{24} \log \frac{1}{\epsilon}$$

which shows the essential tightness of the bounds in Theorem 5 and Corollary 4.

Chapter 6

Averaging with Dynamic Topologies

We consider the problem of guaranteeing convergence to the average for the symmetric model in the presence of a dynamic topology. This question has previously been considered in the papers [29] and [31]. We present a new algorithm for averaging in this setting; our algorithm is the first to be accompanied by a polynomial-time convergence bound.

We first show in section 6.1 that in the absence of symmetry, the agreement algorithm may take exponentially long to converge. Note that the performance of the agreement algorithm in the symmetric model with dynamic topology is an open question. Motivated by this we introduce of an algorithm, in section 6.2, for averaging in symmetric, dynamic environments; a polynomial-time convergence bound for this algorithm is proved in section 6.3.

6.1 Exponential Convergence Time for the Agreement Algorithm

We begin by formally defining the notion of "convergence time" for dynamic graph sequences. Given a sequence of graphs $G(t)$ on n vertices such that Assumption 3

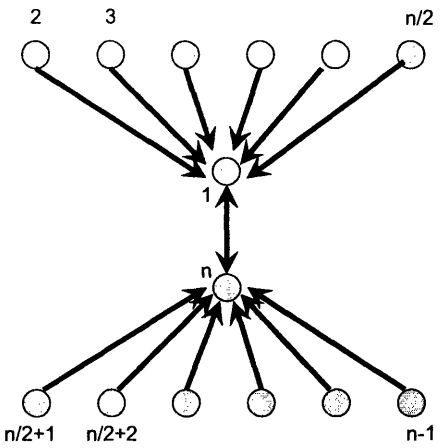

Figure 6-1: The iteration graph at time 0. Self-loops which are omitted in the figure are present at every node.

of Chapter 2 is satisfied for some $B > 0$, and an initial condition $x(0)$, we define the convergence time $T_{G(t)}(x(0), \epsilon)$ as the first time when each agent is within ϵ of the final consensus. In other words, $t = T_{G(t)}(x(0), \epsilon)$ is the smallest t such that $||x(t) - \lim_t x(t)||_\infty \leq \epsilon$.

In this subsection, we will show that there exists a sequence $\hat{G}(t)$ of directed graphs with $B = n+1$, and a corresponding set of initial values $\hat{x}(0)$, such that $T_{\hat{G}(t)}(\hat{x}(0), \epsilon)$ is at least $\frac{(n/2)^n(n+2)(n+1)}{4} \log \frac{1}{\epsilon}$. This shows that the bounds on convergence time of the equal-neighbour model derived in Section 2.3 are essentially tight if the graphs $G(t)$ may be directed.

In the following subsection, we give the details of the construction of $\hat{G}(t)$ and $\hat{x}(0)$.

6.1.1 Construction

The initial condition $\hat{x}(0)$ is defined as $\hat{x}_i(0) = 1$ for $i = 1, \ldots, n/2$ and $\hat{x}_i(0) = -1$ for $i = n/2+1, \ldots, n$ (here n is assumed to be even). Let the graph at time 0 be the graph shown at Figure 6-1 (note that the figure omits self-loops, which are present at every node) [1].

[1]Formally, this is the graph composed of the edges $\{(j,1)|j \in \{1,\ldots,n/2\}\}$, $\{(j,n)|j \in \{n/2+1,\ldots,n\}\}$, $\{(1,n),(n,1)\}$, and self-loops $\{(i,i) \mid i = 1,\ldots,n\}$.

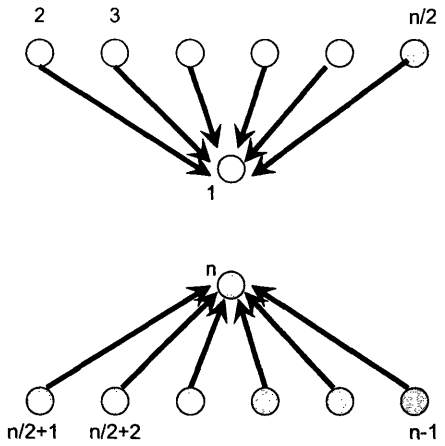

Figure 6-2: Iteration graph at times $1, \ldots, B-2$. Self-loops which are omitted in the figure are present at every node.

The result of the equal-neighbour iteration will be:

$$\hat{x}_1(1) = \frac{-1 \cdot 1 + (n/2) \cdot 1}{n/2 + 1} = \frac{n-2}{n+2},$$

and by symmetry $\hat{x}_n(1) = -(n-2)/(n+2)$; all other $\hat{x}_i$ remain the same at time 1 as they were at time 0, i.e. $\hat{x}_2(1) = \hat{x}_3(1) = \cdots = \hat{x}_{n/2}(1) = 1$ and $\hat{x}_{n/2+1}(1) = \hat{x}_{n/2+2}(1) = \cdots = \hat{x}_{n-1}(1) = -1$.

Next, for $t = 2, \ldots, B-2$, we perform an equal-neighbor iteration on the graph in Figure 6-2 [2].

Clearly this iteration does not alter the values of $\hat{x}_i$ for $i \neq 1, n$. Thus, after all these iterations are completed at time $B-1$, we will still have $\hat{x}_2(1) = \hat{x}_3(1) = \cdots = \hat{x}_{n/2}(1) = 1$ and $\hat{x}_{n/2+1}(1) = \hat{x}_{n/2+2}(1) = \cdots = \hat{x}_{n-1}(1) = -1$.

We now consider what happens to $\hat{x}_1$ between times $t = 2$ and $t = B-1$. If at some time t the value of $\hat{x}_1$ is $1-\alpha$, then at time $t+1$,

$$\hat{x}_1(t+1) = \frac{(1 \cdot (1-\alpha) + (n/2-1) \cdot 1}{n/2} = 1 - (2/n)\alpha.$$

Thus, the gap between the value of x_1 and 1 shrinks by $(2/n)$ after such an iteration. Since initially the gap is $4/(n+2)$, we have that $x_1(B-1) = 1 - (4/(n+2))(2/n)^{B-2}$.

[2]Formally, this is the graph composed the edges $\{(j,1)|j \in \{1,\ldots,n/2\}\}$. $\{(j,n)|j \in \{n/2+1,\ldots,n\}\}$, and self-loops $\{(i,i) \mid i = 1,\ldots,n\}$.

By symmetry, $x_n(B-1) = -1 + (4/(n+2))(2/n)^{B-2}$.

Finally, at time $B-1$, we iterate on the complete graph over vertices $\{1,\ldots,n/2\}$ and the complete graph over vertices $\{n/2+1,\ldots,n\}$. The result is that nodes $\{1,\ldots,n/2\}$ will have value $1-(4/(n+2))(2/n)^{B-1}$ and nodes $\{n/2+1,\ldots,n\}$ will have value $-(1-(4/(n+2))(2/n)^{B-1})$.

6.1.2 Analysis

Using the above expressions we have just derived for the components of $x(B)$, we can infer that,

$$\frac{|\max_i \hat{x}_i(B) - \min_i \hat{x}_i(B)|}{|\max_i \hat{x}_i(0) - \min_i \hat{x}_i(0)|} = \frac{2-2(4/(n+2))(2/n)^{B-1}}{2} = 1-(4/(n+2))(2/n)^{B-1}.$$

Moreover, because $\hat{x}(B)$ is simply a scaled $\hat{x}(0)$, it is clear that applying this scheme repeatedly we have $\frac{|\max_i \hat{x}_i(tB)-\min_i \hat{x}_i(tB)|}{|\max_i \hat{x}_i(t)-\min_i \hat{x}_i(t)|} = (1-(4/(n+2))(2/n)^{B-1})^t$. Taking $B = n+1$, we have that the time to until $\frac{|\max_i \hat{x}_i(tB)-\min_i \hat{x}_i(tB)|}{|\max_i \hat{x}_i(t)-\min_i \hat{x}_i(t)|}$ is less than ϵ will take

$$t = B \cdot \frac{1}{-\log[1-\frac{42^n}{(n+2)n^n}]}(\log\frac{1}{\epsilon}) \geq \frac{(n/2)^n(n+2)(n+1)}{4}\log\frac{1}{\epsilon},$$

which is exponentially large in n.

On the other hand, it is easy to see that this graph sequence satisfies Assumption 3 of Chapter 2. Indeed, letting $\hat{\mathcal{E}}(t)$ denote the edge set of the graph $\hat{G}(t)$, we have that the graph $(\{1,\ldots,n\}, \bigcup_{i=0}^{B-1}\hat{E}(t))$ - which Assumption 3 requires to be strongly connected - is simply two copies of the bidirectional complete graph $K_{n/2}$ joined by a bidirectional edge. Clearly, this graph is strongly connected.

Since B is a linear function of n, and we have lower bounded the convergence time by an exponential in n, it follows that no polynomial upper bound on convergence time in terms of n and B is possible.

6.2 Description of the Algorithm

Our algorithm is a variation of an old *load balancing* algorithm (see [11] and chapter 7.3 of [5]). Intuitively, a collection of processors with different loads try to equalize their respective loads. In particular, processors with higher loads send some of their load to neighbors with smaller loads. As load propagates through the network in this manner, the disparate loads at the nodes approach equality.

Similarly, at each step of our algorithm, each node offers some of its value to its neighbors, and accepts or rejects such offers from its neighbors. Once an offer from i to j to send δ has been accepted, the updates $x_i \leftarrow x_i - \delta$ and $x_j \leftarrow x_j + \delta$ are executed.

As before, we assume a time-varying graph sequence $G(t)$. We only make two assumptions on $G(t)$: symmetry and bounded intercommunication times (see Chapter II for definitions).

We next describe the formal steps the nodes execute at each time t. For definiteness, we refer to the node executing the steps below as node A. Moreover, the instructions below sometimes refer to the neighbors of node A; this always means current neighbors at time t, when the step is being executed (since $G(t)$ is not assumed to be constant with t, the set of neighbors of A may be time-varying).

1. Node A broadcasts its current value x_A to all its neighbors.

2. Going through the values it just received from its neighbors, Node A finds the smallest value less than itself. Let B be a neighbor with this value. Node A makes an offer of $\frac{1}{2}(x_A - x_B)$ to node B.

   If no node has a value smaller than x_A, node A does nothing at this stage.

3. Node A goes through the incoming offers. It sends an acceptance to the sender of the largest offer and a rejection to all the other senders. It updates the value of x_A by adding the value of the accepted offer.

   If no offers were received, node A does nothing at this stage.

4. If an acceptance arrives from the offer made by node A, node A updates x_A by subtracting the value of the offer.

Theorem 6. Let $x(t)$ be the vector of values at the beginning end of stage t. Then, in the symmetric model with Assumption 3, $\lim_{t\to\infty} x_i(t) = \frac{1}{n}\sum_{k=1}^{n} x_k(0)$, for all i.

While Theorem 6 admits an independent proof, it is omitted here as the result follows from the bounds in the next section.

6.3 Convergence Time

No bounds on the convergence time can be obtained without the assumption of bounded intercommunication times. (Indeed, by inserting arbitrarily many empty edge sets $\mathcal{E}(t)$ into the sequence, convergence can be made arbitrarily slow). However, given an upper bound B on the length of the intercommunication times, it is possible to obtain a bound on the performance of the algorithm, which we describe below.

Define T_ϵ to be the first time such that $||x(t) - (1/n)\sum_k x_k(0)\mathbf{1}||_2 \leq \epsilon$ for all $t \geq T_\epsilon$, where $\mathbf{1}$ is the vector with all entries equal to 1.

Theorem 7. Suppose that $\sum_k x_k^2(0) = 1$. Then, there exists a constant $C > 0$ such that for every n and B (this is the constant in Assumption 3), and every sequence of symmetric edge sets $\mathcal{E}(t)$ that satisfy Assumption 3,

$$T_\epsilon \leq CBn^3 \log \frac{1}{\epsilon}, \tag{6.1}$$

for all $\epsilon > 0$.

Proof:

1. Define $V(t) = ||x(t) - \frac{1}{n}\sum_{i=1}^{n} x_i(0)\mathbf{1}||_2^2$. We will show that $V(t)$ is nonincreasing in t and that

$$V((k+1)B) \leq \Big(1 - \frac{1}{2n^3}\Big)V(kB).$$

where k is any positive integer.

These two claims readily imply the theorem. To see this, note that the condition $||x(t) - (1/n)\sum_k x_k(0)\mathbf{1}||_2 \leq \epsilon$ is equivalent to $V(t) \leq \epsilon^2$. Moreover, if $V(t)$ decreases by a factor of $1 - \frac{1}{2n^3}$ every B steps, then the amount of time we have to wait until $V(t) \leq \epsilon^2$ is $B \log_{(1-\frac{1}{2n^3})} \epsilon^2$. This latter quantity can be upper bounded by $CBn^3 \log \frac{1}{\epsilon}$ for a constant C. Finally, since $V(t)$ is nonincreasing, $V(t)$ stays below ϵ^2 henceforth.

2. We first show that the "variance" $V(t)$ is nonincreasing. We argue that while rejected offers clearly do not change $V(t)$, each accepted offer at time t results in a decrease of $V(t+1)$. We will view the changes at time t as a result of a series of sequentially accepted offers, each of which must reduce the "variance" V.

   To make this formal, let us break time t into n separate time periods as follows. Order the nodes from smallest to largest so that $x_1(t) \leq x_2(t) \leq \cdots \leq x_n(t)$, breaking ties in any way. Let $A_i(t)$ denote the amount of the offer accepted by node i at time t (if any). If the node accepted no offers at time t, set $A_i(t) = 0$. Further, if $A_i(t)$ is strictly positive, let $\mathcal{A}_i(t)$ be the the index of the node whose offer node i accepted.

   - During the first time period, we will have node 1 execute $x_1(t) \leftarrow x_1(t) + A_1(t)$ and we will have node $\mathcal{A}_1(t)$ execute $x_{\mathcal{A}_1(t)}(t) \leftarrow x_{\mathcal{A}_1(t)}(t) - A_1(t)$. In other words, both 1 and the node whose offer it accepted execute their update at time 1. On the other hand, if $A_1(t) = 0$, nothing happens at this period.
   - Similarly, during the kth time period, we will have node k execute $x_k(t) \leftarrow x_k(t) + A_k(t)$ and we will have node $\mathcal{A}_k(t)$ execute $x_{\mathcal{A}_k(t)}(t) \leftarrow x_{\mathcal{A}_k(t)}(t) - A_k(t)$. In other words, both k and the node whose offer it accepted execute their update in period k. On the other hand, if $A_k(t) = 0$, nothing happens at this period.

   We note that every offer accepted at time t appears at some stage in the

above sequence. We next argue that each offer decreases V. This will complete the proof that $V(t)$ is nonincreasing in t.

Let us suppose that after breaking down time t into n time periods as described above, an offer occurs at the ith period. Then the offer from some node must be accepted by node i; let us refer to the sender of the offer as node j. Now because nodes only send offers to lower valued nodes, $x_j(t) > x_i(t)$ at the beginning of time t. After we break down time t into n periods as described above, we claim that this relationship remains the same by the time the ith time period is reached. Indeed, x_j is unchanged from the beginning of time t (it can only send one offer, which was to x_i; and if it has received any other offers, this must be counted later at period $j > i$); moreover, x_i could not have increased (since it is only allowed to accept one offer). Therefore, x_j is still strictly greater than x_i by the time the ith period arrives.

Without loss of generality, assume that $\sum_i x_i(0) = 0$ (translation does not change the argument presented here). The contribution of these two nodes to $V(t)$ is $x_i^2 + x_j^2$. It is easy to see that this expression is bigger than $2(\frac{x_i+x_j}{2})^2$, which is the contribution of these two nodes to V after the update.

Summarizing, we can sequentialize the offers accepted at time t in such a way that each accepted offer will decrease V. Thus, $V(t)$ is nonincreasing.

3. We argue that that the quantity $V(t)$ decreases by a factor of at least $1-1/(2n^3)$ during one update between times $t_{\text{initial}} = kB + 1$ and $t_{\text{final}} = (k+1)B$.

   Without loss generality, we (i) take 0 to be the initial time; (ii) assume that $\sum_i x_i(0) = 0$ (translation does not change the argument presented here); (iii) Let $\max_i |x_i(0)| = 1$ so that all the values lie in the interval $[-1, +1]$. It follows that $V(t) \leq n$.

   Let us order the nodes in increasing order of values at time 0 so that $x_1(0) \leq x_2(0) \leq x_3(0) \leq \ldots x_n(0)$. The largest gap between any two consecutive $x_i(0)$ must be at least $1/n$. Let i, j be two consecutive nodes whose difference achieves or exceeds this bound. We argue that over the time interval $t = 1, \ldots, B$

some node must accept an offer whose value is at least $1/2n$. Indeed, since $\bigcup_{s=1,\ldots,B} \mathcal{E}(t)$ is connected, there will be a first time at which a node A with index at least j communicates with a node whose index is at most i. Because node A is required to make an offer to the smallest neighbor with value less than x_A, it will make an offer to some node with index at most i. Let C be this node. The gap between x_A and x_C is at least $1/n$. Thus, the value of the offer is at least $1/2n$. Since C accepts the largest offer, either this offer or a larger offer is accepted.

Let D be the node whose offer node C accepts (it may be that $D = A$, or D may be different from A). Since the offer accepted by C is at least $1/2n$, a simple computation shows that $x_D^2 + x_C^2$ is reduced by at least by $1/(2n^2)$ due to the communication of D and C. It follows that the percentage decrease in the variance $V^2(t)$ is at least $1/(2n^3)]$ by time B.

Chapter 7

Simulations

This thesis proposes three new algorithms for the distributed consensus problem. For one of these, namely the tree heuristic of Chapter 5.3, the theoretical performance has been characterized completely - indeed Corollary 4 provides an upper bound on the convergence time, while Example 1 provides a lower bound. Moreover, as $\epsilon \rightarrow 0$, the two bounds differ only by a multiplicative constant.

In this chapter, we provide simulations for the remaining two algorithms.

7.1 Averaging in fixed networks with two passes of the agreement algorithm

In Chapter 4.1, we proposed a method based on the agreement algorithm for averaging in fixed graphs. We speculated that the presence of a small number of vertices of high degree would make the performance of our algorithm attractive relative to the algorithm of [32], which uses a step size proportional to the inverse of the largest degree (we used step size of $\epsilon = \frac{1}{2d_{\max}}$ in our implementation). Figures 7-1 and 7-3 reflect simulations of the two algorithms. In each simulation, we generate an Erdos-Renyi random graph $G(c, n)$, i.e., each edge is independently present in the graph with probability c/n. Thus c is the expected degree of each node; in our simulations, we keep c fixed while letting n get large. Moreover, it is known that $c = 1$ is a transition

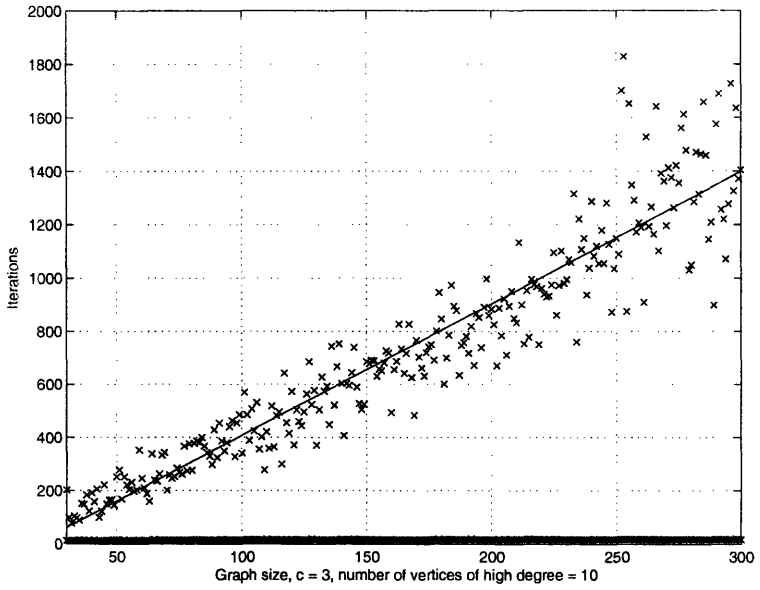

Figure 7-1: Comparing averaging algorithms when $c = 3$. The top line corresponds to the algorithm of [32], and the bottom line corresponds to two parallel passes of the agreement algorithm.

occurs in random graphs for $c = 1$ [15]; for this reason we provide simulations for both cases with $c < 1$ and $c > 1$.

We next change the random graph $G(c, n)$ by picking n_d vertices at random and adding edges randomly making the degree of these vertices linear in n ($n_d = 10$ in both figures), and making each edge incident to these edges is present with probability $1/3$. If the resulting graph is disconnected, we reject it and do not run the algorithm. Else, we run the algorithm until the largest deviation from the mean was at most $\epsilon = 10^{-3}$.

The outcomes, shown in Figures 7-1 and 7-3, are the average of three runs. We conclude that the algorithm of [32] grows at a considerably faster pace on these graphs than the algorithm we propose here.

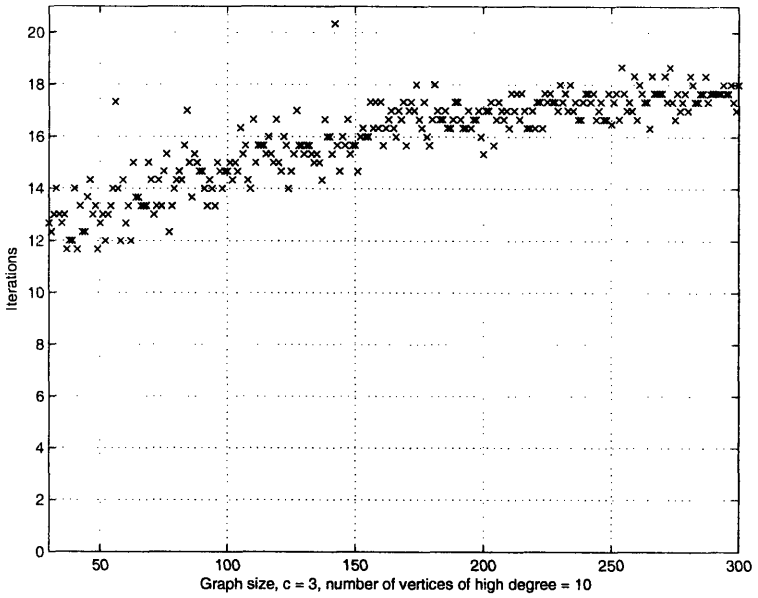

Figure 7-2: A blow up of the performance of the agreement algorithm when $c = 3$.

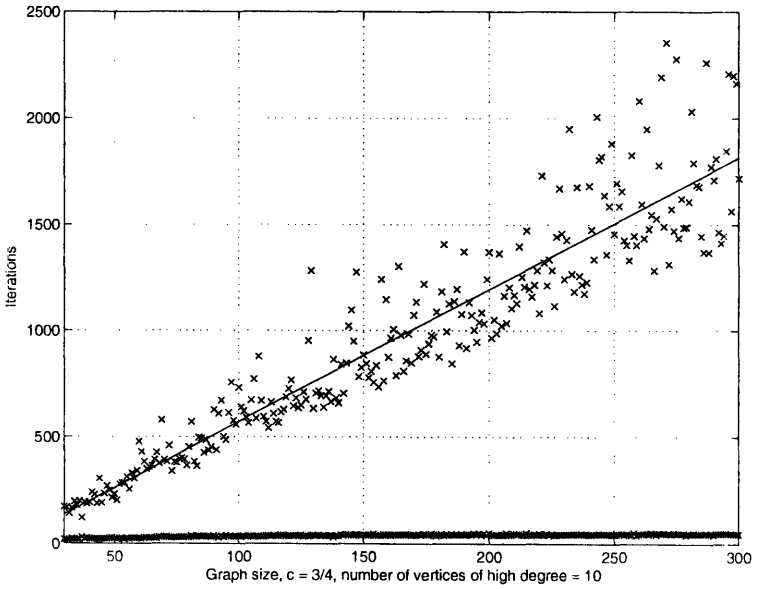

Figure 7-3: Comparing averaging algorithms when $c = 3/4$.The top line corresponds to the algorithm of [32], and the bottom line corresponds to two parallel passes of the agreement algorithm.

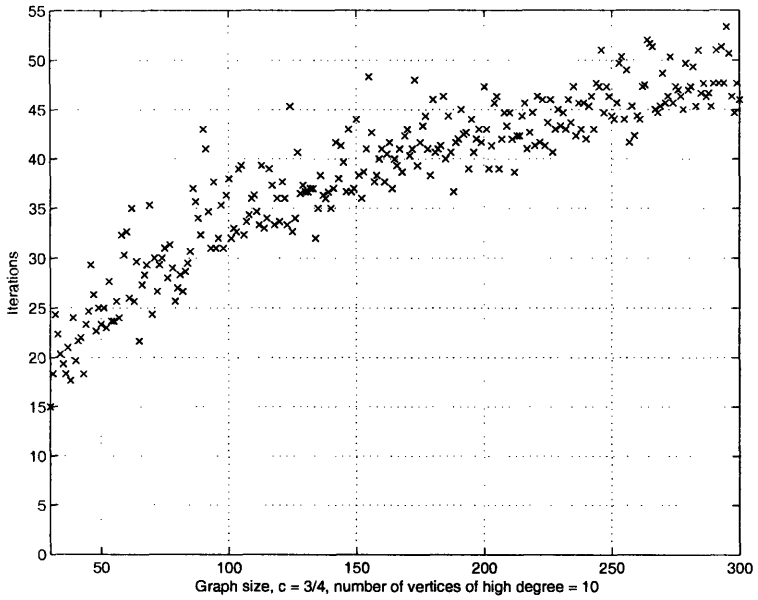

Figure 7-4: A blow up of the performance of the agreement algorithm when $c = 3/4$.

7.2 Averaging in time-varying Erdos-Renyi random graphs

We report on simulations involving the load-balancing algorithm on time-varying random graphs. At each time t, we independently generate an Erdos-Renyi random graph $G(t) = G(c, n)$ - see the previous section for basic background on random graphs. As before, we simulate both the $c < 1$ case and the $c > 1$ case. If the largest deviation from the mean is at most $\epsilon = 10^{-3}$, we stop; else, we perform another iteration of the load-balancing algorithm.

The results are summarized in Figures 7-5 and 7-6, which again show averages of three runs. Figure 7-5 shows that the procedure takes only a sublinear number of iterations in dense random graphs ($c > 1$). Figure 7-6 shows a slow, approximately linear growth when the random graph is sparse ($c < 1$).

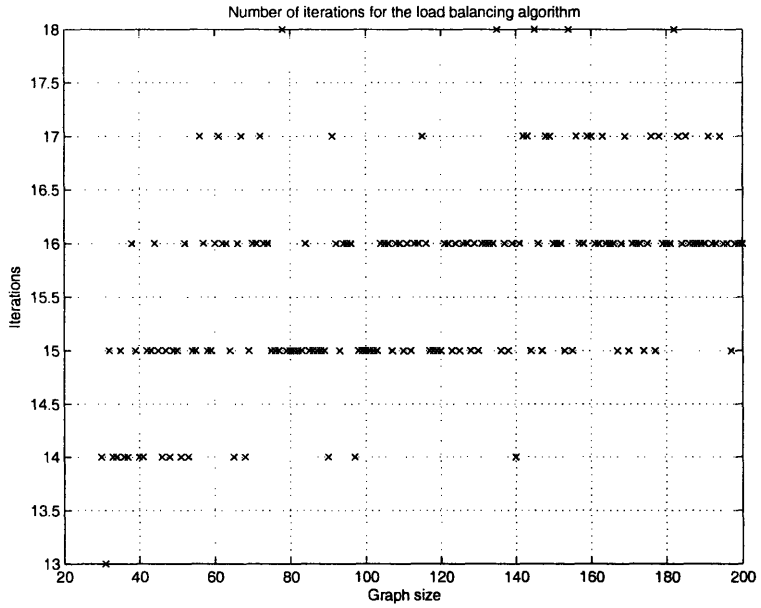

Figure 7-5: Averaging in time-varying Erdos-Renyi random graphs with the load balancing algorithm. Here $c = 3$ at each time t.

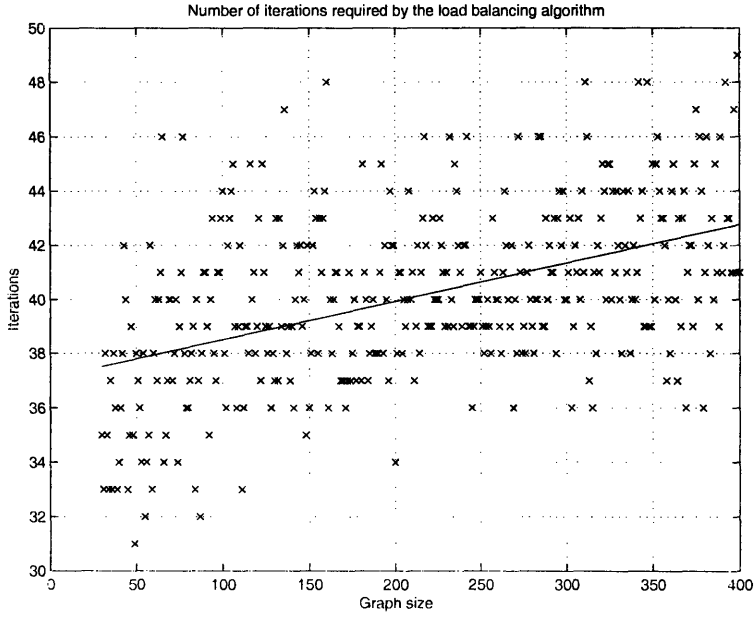

Figure 7-6: Averaging in time-varying Erdos-Renyi random graphs with the load balancing algorithm. Here $c = 3/4$ at each time t.

Bibliography

[1] D. Angeli, P.A. Bliman, "Stability of leaderless multi-agent systems. Extension of a result by Moreau," http://www.arxiv.org/abs/math.OC/0411338.

[2] D. P. Bertsekas and R. G. Gallager,"Data Networks, Prentice Hall, 2nd edition, 1991.

[3] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, "Gossip Algorithms: Design, Analysis, and Application", *IEEE INFOCOM 2005.*

[4] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, "Randomized Gossip Algorithms," to appear in a joint special issue of the IEEE Transactions on Information Theory and the ACM/IEEE Transactions on Networking, June 2006.

[5] D. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods,* Prentice Hall, 1989.

[6] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in Multiagent Coordination, Consensus, and Flocking," in *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Seville, Spain, December 2005.

[7] M. Cao, A. S. Morse, B. D. O. Anderson, Coordination of an Asynchronous, Multi-Agent System via Averaging, preprint, 2004.

[8] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks," *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004

[9] F. Cucker. S. Smale, "Emergent Behaviour in Flocks." preprint.

[10] M. Cao, D.A. Spielman, A.S. Morse, "A Lower Bound on Convergence of a Distributed Network Consensus Algorithm," *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Seville, Spain, December 2005.

[11] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *Journal of Parallel and Distributed Computing*, Vol. 7, No. 2, 1989.

[12] M. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association,* Vol. 69, No. 345, pp. 118-121, 1974.

[13] P. Diaconis, D. Stroock, "Geometric bounds for the eigenvalues of Markov chains," *The Annals of Applied Probability,* Vol. 1, No. 1, pp. 36-61, 1991.

[14] A.G. Dimakis, A.D. Sarwate, M. J. Wainwright, "Geographic Gossip:Efficient Aggregation for Sensor Networks," http://www.arxiv.org/abs/cs.IT/0602071

[15] P. Erdos, A. Renyi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci,* 1960.

[16] L. Fang, P.J. Antsaklis, "On Communication Requirements for Muli-Agent Consensus Seeking," *NESC,* 2005.

[17] J. Friedman, J. Kahn, E. Szemeredi, "On the second eigenvalue of random regular graphs," *Proceedings of the twenty-first annual symposicum on the theory of computing (STOC) 1989.*

[18] G.R. Grimmett, D.R. Stirzaker, *Probability and Stochastic Processes,* Clarendon Press. Oxford, 1992.

[19] D. Grunbaum, S. Viscido, J.K. Parrish, "Extracting Interactive Control Algorithms from Group Dynamics of Schooling Fish," Cooperative Control, ed. V. Kumar. N. Leonard, A.S Morse, *Lecture Notes in Control and Information Sciences.* Springer. 2004.

[20] N. Higham, "Accuracy and Stability of Numerical Algorithms," SIAM, 2002.

[21] J. M. Hendrickx and V. D. Blondel, "Convergence of different linear and non-linear Vicsek models," Universite catholique de Louvain, CESAME research report, RR 2005.57, 2005.

[22] D. Hernek, "Random walks on colored graphs: analysis and applications," Ph.D Thesis, Department of Computer Science, University of California at Berkeley, 1995.

[23] A. Jadbabaie, J. Lin, and A.S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol 48, Np. 3, pp. 988-1001, 2003.

[24] S. Li and H. Wang, Multi-agent coordination using nearest-neighbor rules: revisiting the Vicsek model; http://arxiv.org/abs/cs.MA/0407021.

[25] L. Lova'sz, "Random Walks on Graphs: A Survey," *Combinatorics, Paul Erdo s is Eighty, Vol. 2,* (ed. D. Miklo's, V. T. So's, T. Szo nyi), Ja'nos Bolyai Mathematical Society, Budapest, 1996, 353–398.

[26] Y.Liu, Y.R. Yang, "Reputation Propagation and Agreement in Wireless Ad Hoc Networks," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC),* 2003.

[27] H.J. Landau, A.M. Odlyzko, "Bounds for Eigenvalues of Certain Stochastic Matrices," *Linear Algebra and Its Applications*, 38:5-15, 1981.

[28] A. Muhammad and M. Egerstedt. "Connectivity Graphs as Models of Local Interactions," *Journal of Applied Mathematics and Computation.* Vol. 168, No. 1. pp. 243-269, Sept. 2005.

[29] M. Mehyar, D. Spanos, J. Pongsajapan, S.H. Low, R.M. Murray, "Distributed Averaging on Asynchronous Communication Networks," *Proceedings of the Joint*

44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05), Seville, Spain, December 2005.

[30] L. Moreau, "Consensus seeking in multi-agent systems using dynamically changing interaction topologies," *IEEE Transactions on Automatic Control,* vol 50, No. 2, 2005.

[31] C. Moallemi and B. Van Roy, "Consensus Propagation," *Proceedings of Neural Information Processing Systems,* Vancouver, Canada, December 2005.

[32] R. Olfati-Saber and R. Murray,"Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transacations on Automatic Control,* Vol. 49, No. 9, September 2004.

[33] R. Olfati-Saber, J. A. Fax, and R. M. Murray."Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE,* submitted.

[34] S. Rao, C. Papadimitriou, S. Shenker, I. Stoica, "Geographic Routing Without Location Information," *Proceedings of the 9th annual international conference on mobile computing and networking,* 2003

[35] S. Ross, *Stochastic Processes,* John Wiley, 1996.

[36] P. Tabuada, G.J. Pappas, and P. Lima, "Feasible formations of multi-agent systems," *Proceedings of the American Control Conference,* 2001.

[37] S. Treil, *Linear Algebra Done Wrong,* available at

`http://www.math.brown.edu/~treil/papers/LADW/LADW.html`

[38] J. N. Tsitsiklis."Problems in Decentralized Decision Making and Computation," Ph.D. Thesis. Department of EECS, MIT, 1984

[39] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," *IEEE Transactions on Automatic Control,* Vol. 31, No. 9, 1986.

[40] "Flocking in fixed and switching networks," H. Tanner, A. Jadbabaie, and G.J. Pappas, submitted to *IEEE Transactions on Automatic Control*, April 2005.

[41] T. Vicsek, E. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet. "Novel Type of Phase Transitions in a System of Self-Driven Particles," *Physical Review Letters*, 75, 1995.

[42] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society,* Vol. 15, 1963, pp. 733-737.

[43] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters,* Vol. 53, pp. 65-78, 2004.