© 2009 Kiran Lakkaraju

AGREEMENT, INFORMATION AND TIME IN MULTIAGENT SYSTEMS

BY

KIRAN LAKKARAJU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Doctoral Committee:

Professor Les Gasser, Chair
Professor Gul Agha
Assistant Professor Indranil Gupta
Associate Professor Piotr Gmytrasiewicz, University of Illinois-Chicago
Dr. Samarth Swarup, Virginia Bioinformatics Institute, Virginia Tech

Abstract

This dissertation studies multiagent agreement problems – problems in which a population of agents must agree on some quantity or behavior in a distributed manner. Agreement problems are central in many areas, from the study of magnetism (Ising model), to understanding the diffusion of innovations (such as the diffusion of hybrid corn planting in Illinois), to modeling linguistic change.

The thesis of this dissertation is that the ability for agents to optimally allocate resources towards 1) gaining information from which to infer the agreeing population's global agreement state ("information gathering") and 2) effectively using that information to make convergence decisions that move towards agreement ("information use"), are the fundamental factors that explain the performance of a distributed agreement-seeking collective, and that variations on these processes capture all prevalent styles of agreement problems.

In this dissertation we develop a taxonomic framework that organizes a wide range of agreement problems according to constraints on information gathering and information use. We explore two specific instances of agreement problems in more depth; the first modulates information gathering by constraining the ability of agents to communicate; the second modulates information use by constraining the ability of agents to change states.

An understanding of these two components will allow the application of insights from fields such as statistical physics, distributed algorithms, and multiagent systems to bear on language – and in turn carry insights from linguistic agreement to these fields. Note, however, that the purpose of this dissertation is not to model natural phenomena, but rather to explore, through abstract models, some of the fundamental processes that underlie natural phenomena.

Our first contribution is to develop the *Distributed Optimal Agreement* framework – a taxonomic framework through which we can formally identify potential constraints on the two processes of information gathering and use.

Our second contribution is to develop an understanding of the *Fundamental Agreement Tradeoff*, which is a relation between the effort an agent expends to gather information, the accuracy of the information

gathered, and the amount of time it takes for a population to reach agreement.

We develop the Sampled Majority Vote process as a way to explore the fundamental agreement tradeoff by modulating the amount of effort an agent can expend, which in turn affects the accuracy of information gathered. We show, surprisingly, that a population can reach agreement quickly even with a minimal expenditure of effort. This result has impact for any setting in which communication is a resource intensive procedure (e.g., energy constrained sensor networks). We provide extensive numerical simulations of the Sampled Majority Vote process in a variety of settings. In addition, we we analytically show that the Sampled Majority Vote process reaches agreement under a mean-field assumption.

Our third contribution is to study agreement in complex spaces with boundedly rational agents where there are significant restrictions on communication. We develop the *Distributed Constraint Agreement* problem (which itself is a type of agreement problem that can be captured by the DOA framework) in order to explore the impact of bounded rationality and communication on agreement in complex spaces.

As an example scenario we abstractly model the linguistic phenomenon of the Great English Vowel Shift (GEVS) – a shift in the pronunciation of certain vowels that took place between 1450 and 1750. We define a simple algorithm and through extensive simulation show that a vowel shift could have occurred if a new population of linguistic users, with slightly different pronunciations, entered the linguistic community. These results lend support to the "migration" hypothesis for the GEVS – that due to casualties from the Black Death the linguistic composition of upper class England changed to incorporate individuals with different pronunciations.

Together, these three contributions move us closer to forming a general theory of agreement.

Dedicated to my father, Rao S. Lakkaraju.

Acknowledgements

This dissertation would not have been possible without the help and support of many many people.

I am deeply indebted to my advisor Les Gasser. When I started grad school I had only the vaguest ideas of what research was, and how to go about doing it. He took me on and provided an immense depth and breadth of knowledge, as well as constant encouragements and support. He was always able to find the bright side of all the issues the cropped up.

His ability to take a complex problem and find the core underlying issues, and then be able to clearly and elegantly convey them was a privilege to behold and a talent that I aspire to. I learned from him the importance of not getting too distracted by the details and to always focus on the core issues. Through his example I learned how to do research, and more importantly, why.

Samarth Swarup has been a colleague, mentor and friend throughout these years. He has always listened to my ideas and found a way to extract an interesting issue, even from the most ill-concevied ones. He has gone out of his way to help me become a better researcher.

Without Les and Samarth, this dissertation could not have been written.

Sylvian Ray has provided encouragement and great wisdom during my early years as a graduate student. I wish you were here to celebrate this occasion.

Bill Yurcik and Adam Slagell for supporting me all these years and introducing me to whole new areas of computer science.

The members of my committee, Indranil Gupta, Gul Agha and Piotr Gmytrasiewicz, for providing insightful questions and suggestions that has led to a stronger dissertation.

Ratna Bearavolu, Subhangan Venkatesan, and Ramya Nagarajan for their friendship throughout the years. You made this process bearable.

David Alber, Lee Baugh, Brian Greskamp, Rajesh Karmani, Ramesh Chandrasekharan, Sriram Narayanan, Divya Chandrasekhar, Brant Chee, and Anshu Gaur for providing interesting discussions and great fun throughout the years.

Viraj Kumar and Gloria Rendon for listening to hours on end of practice talks and still providing great suggestions and insight and for providing great company on innumerable number of Wok trips.

Anand Jayaraman for helping me out in a time of stress.

Anusha Anantharaju, for her love and strength and always being there for me. She showed me the path to be a better person and has helped me realize my full potential.

To my parents, Rao and Bhargavi Devi Lakkaraju, and my sister Srilatha Lakkaraju. Their love, support, and confidence in me gave me the freedom to explore the world of science and has been the main reason for my success. My first thoughts of getting a Ph.D. came from my father who has always been a scientist. It it to him that I dedicate this dissertation.

Table of Contents

Chapter 1

Introduction

1.1 Multiagent agreement problems

This dissertation studies *Multiagent Agreement Problems (MAP)* – problems where a set of autonomous, distributed agents must agree on some issue.

A classic example of a MAP is the *distributed commit problem* [Lynch, 1997]. In this scenario, a set of databases must agree upon whether to abort or commit a distributed transaction. Each database has made a decision based on their local information about the transaction. The goal is for all the databases to come to a common decision. If at least one database decides to abort a transaction, then all the databases must abort the transaction. Thus the main issue is the dissemination of local decisions. Once the local decision of every database is known to every other database agreement can be easily reached – if there is at least one abort, everyone aborts, else commit. In this problem agents are databases, and they can be in either the *abort* state or the *commit* state.

A solution to a MAP is called an *Agreement Protocol* and it specifies the behavior of agents such that the system reaches agreement. For the distributed commit problem the "two-stage commit" protocol is an agreement protocol [Lynch, 1997].

MAPs appear in many other domains as well. In recent years there has been a burst of research on distributed, cooperative behavior among unmanned autonomous vehicles (UAVs). In most of the cases the vehicles must make decisions based on limited, imperfect information from other vehicles.

One type of cooperative behavior is alignment – where the UAVs all align themselves to the same direction and speed. To achieve this objective it is necessary for vehicles to communicate their speed and direction. However, due to the context of communication (i.e. while moving and in unknown terrain) there might be significant communication delays as well as failures of communication. Thus the main issue in this case is the transmission of information under communication restrictions.

Alignment can be viewed as an agreement problem, where agents are vehicles and the state of an agent is its current velocity [Fax and Murray, 2004].

Agreement is often a necessary prerequisite to other outcomes. For instance, leader election is a prerequisite to many other algorithms in distributed systems; we can view algorithms to select a leader as agreement problems where agreement is over which node should be the leader. Solutions to this problem involve the passing of information between nodes [Barbosa, 1996].

MAPs are also used as models of physical systems, perhaps the best known is the Ising model used in statistical physics. The Ising model is a simplified microscopic description of ferromagentism. The Ising model defines a discrete set of variables called *spins*. Each spin can take on a value of -1 or +1. Each node in a graph is assigned a spin. Originally, the spins were used as a representation of electrons – when all the electrons are in the same state the material exhibits ferromagnetism [Giordano, 1997].

There are many instances of agreement problems in social systems. Consider the study of innovation diffusion. An innovation is novel and thus oftentimes learning about it from others is the only means by which an innovation is adopted by entities[1] (be they people or organizations). Thus innovation diffusion studies have looked at how communication patterns and properties of the innovation affect adoption in a population. Classic examples include studies of the spread of hybrid corn among Illinois farmers and the diffusion of technologies such as the telephone and the diffusion of prescription drugs [Strang and Soule, 1998].

Innovation diffusion is a type of agreement problem. The agents are the entities that can adopt or reject an innovation. Each agent can be in one of two states, either they have adopted the innovation in question, or they have not. When all agents have adopted an innovation (or decided not to adopt an innovation) the system is in agreement. Thus, we can view models of innovation diffusion as instances of a general agreement problem.

There are numerous other instances of agreement problems in social systems. The study of disease propagation is an agreement problem where each person can be viewed as either having a disease or not having a disease [Goldstone and Janssen, 2005]. Cultural dissemination models study the adoption of cultural practices [Axelrod, 1997]. Note that in some of these models the objective is to understand when agreement is *not* reached – clearly knowing when agreement is reached can help fulfill this objective.

Of particular interest to this dissertation are agreement problems in linguistics. Language is an extremely important part of human society and indeed, is often thought of as what makes humans human. Language is not a fixed entity though; there are changes at all levels: grammatically (e.g., the decline in use of the absolute construction [Baron et al.,]), lexically (e.g. note the widespread use of the terms “to google”, “aerobicized”, “crunk”) and even phonetically (e.g. like the Great English Vowel shift [Perkins, 1977]).

[1]As opposed to each entity reinventing the innovation themselves.

How do these changes come about? Clearly there is no central coordinator of language that identifies how language changes, rather, variation in individual languages propagate through a population.

Why do individuals change their language? After childhood language learning is a difficult process. However, linguistic change occurs even in adulthood (consider the new phrase "to google"). The reason is that language is useful only if it is shared with others. Thus, even though it means significant effort, people undertake to modify their language in order to better communicate with others. This makes language a constantly evolving entity.

Linguistic variations come from individuals. There is no centralized language authority that stipulates how a language changes. Instead, individual variations in language spread throughout a population. A shared language is necessary for people to interact and coordinate in order to solve complex tasks. The pressure to be able to communicate with others leads people towards modifying their language as they interact with others. Thus linguistic variations can spread through a population.

In the context of an agreement problem, we can view people as agents and the state space as the space of all possible languages. As humans interact with each other and learn new words/meanings, their language changes. When a population of humans are using the same (or very similar) language they are in agreement.

1.1.1 Why understanding MAPs is important

We make a distinction between two general categories of multiagent agreement problems. The first we call *technological* MAPs. An example of this category is the distributed commit problem from above. In technological MAPs the agents are often intentionally seeking agreement.

In contrast, in what we call *sociological* MAPs, agents are not necessarily intentionally seeking agreement. A state of agreement emerges through agents responding to other, possibly local, stimuli. A great example of this is linguistic agreement. Individuals are maximizing their ability to communicate with their local neighborhood – their family and friends. Through these local interaction a global phenomenon emerges, that of agreement between all individuals, even though any two individuals might never have met before.

We believe that studying MAPs from both of these general categories will yield fundamental insights that span both categories. The impact of this research would be to provide better insight into MAPs as well as allow for cross-fertilization of ideas between categories.

1.1.2 Fundamental issues in agreement problems

The fundamental issues in agreement problems concern the role of information. For agreement to occur, agents must know the states of others in the population. The solution to every agreement problem has two parts: Information gathering and Information use.

Information gathering is the process by which agents gain information about the states of other agents. Agreement is between agents; and thus to agree an agent must know the states of other agents. *Information use* is the process by which agents use the information they have gained to change their state.

The difficulty of solving an agreement problem is a function of the constraints on information gathering and information use. To illustrate, let us consider a simple, synchronous agreement protocol that takes place in a discrete time system:

1. A subset of agents becomes "active" at each time step – that is they take actions at this time step.

2. Each active agent gathers information about the distribution of states in the population.

3. Each active agent changes its state to the "majority state": the state that is the majority in the population. In case of ties, there is a globally known preference ordering over states.

It is clear that this protocol can induce agreement; the number of agents in the majority state will only increase. However, this protocol relies on several assumptions; listed below are the three most important assumptions[2]:

Assumption of Global Communicability It is assumed that every agent has a communication channel to every other agent – that is the state of every agent is known to every other agent. Clearly this is not generally true: for instance, geographical and social boundaries limit communication in social systems. Considering the UAV example, obstructions between UAVs might reduce or eliminate the ability to communicate.

Assumption of Accurate Communication Each agent can accurately communicate its state to another agent. The assumption is that there is no noise in the communication channel. In the distributed commit example messages between databases might be lost or modified in transit.

Assumption of Accessibility Every state is *accessible*: an agent can change its state to any other state. The algorithm above does not consider the fact that state changes might be constrained. For example, adult second language learners are heavily constrained compared to children; some levels of fluency in the second language are inaccessible to adults, but not to children.

[2]Examples of other assumptions: Every agent is chosen to be active at least once (otherwise some agents will never change state); the assumption that there is a globally known preference ordering over states; etc. While these are important to consider, they are not fundamental to understanding the behavior of agreement problems.

The assumptions above impact the information agents can gather and how they use the information. If the global communicability assumption was violated an agent would not be able to obtain information about the majority state of the population. Without accurate information, an agent could commit an error and change to a minority state rather than a majority state. If this behavior occurred enough times the system would not reach agreement. Similarly, if there was inaccurate communication agents could misinterpret the state of other agents; and thus once again commit an error of changing to a minority state.

If the accessibility assumption was violated agents might not be able to move to certain states – more realistically, the cost of moving to certain states might be inordinately high.

1.2 Research aim

It is surprising that with agreement problems being central to so many domains that there exists very little study of agreement problems in general. While there is much work in specific domains, there exists very little work that characterizes the space of agreement problems and identifies the critical issues that make agreement problems difficult: that is, there is no general theory of of agreement problems.

The broad aim of this research is to develop a general theory of agreement problems which must, we believe, be oriented around the central issues of information gathering and information use.

A general theory of MAPs should answer the following questions:

1. How do restrictions on information gathering and use impact agreement? What types of constraints will allow for agreement, and which disallow it completely?

2. How do restrictions on information gathering and use impact the time till agreement?

3. How do restrictions on information gathering and use impact the state that is agreed upon?

The first question addresses whether agreement occurs or not. For instance, two agents that cannot communicate will not be able to reach agreement, unless by chance.

The second question addresses how long it takes for agreement to be reached. This is important in many domains – in the UAV domain quicker agreement times mean better reactivity to external stimuli.

Finally, in situations where there are numerous states it is important to understand which state is agreed upon. For instance, in innovation diffusion models the goal is to understand when an innovation, modeled as a particular state, is accepted by all agents.

The thesis of this dissertation is that the process of agreement is fundamentally about information. We argue that different agreement problems are actually variations on two central processes: information gathering and information use. Thus, in order to understand agreement at a general level, one must understand how agents gather information about others and how they use this information to change their behavior. In this dissertation, we begin to lay the foundations for just such a general theory of agreement.

1.2.1 Why is developing a general theory of agreement hard?

Developing a general theory of agreement is difficult because in many of the domains we consider agreement is an *emergent* phenomenon – it is the product of numerous simple interactions between agents. An elegant example of this is the "v" shape that occurs in groups of birds. This occurs not because a bird is selected as a leader and all other birds follow the leader; but rather through each bird changing its position based on its neighbors position/velocity. The "v" shape is not planned, but rather is an outcome of the local interaction between agents. Modeling and understanding emergent phenomena is difficult because it requires modeling a large population of entities with many interactions [Sawyer, 2005].

Another difficulty in developing a general theory of agreement problems is that there are a large number of possible ways to restrict information gathering and information use processes.

1.3 Dissertation goals

The aim of this dissertation is to begin forming a general theory of agreement. We have four goals:

1. Develop a taxonomic framework to organize the realm of agreement problems under a common terminology and conceptual framework in order to find the common processes as well as formalize the differences between agreement problems.

2. To begin to understand the fundamental interactions between information, cost and time in agreement problems.

3. Characterize some aspects of *linguistic agreement* as an instance of an agreement problem.

4. Explore agreement in complex, constrained state spaces where there are significant limitations to the cognitive and communicative capabilities of agents.

The multitude of agreement problems is a testament to the ubiquity of agreement problems in many domains; however because of the lack of a common conceptual framework there is a wide variety of notations, assumptions, and ways of viewing the same agreement problem. In order to create a general theory of agreement we need to have a common conceptual framework in which all agreement problems can be cast.

We call this the development of a *taxonomic framework* because we are using the framework as a way of classifying agreement problems.

Communication leads to information, and information is what allows agreement to occur. Most studies, with some notable exceptions[3], however, fail to consider the cost of communication in developing and analyzing agreement protocols. The fundamental tradeoff between cost, information and time to agreement is critical to understand.

Agreement problems occur in numerous social systems. Under our overarching taxonomic framework we hope to capture the properties of linguistic agreement problems. The benefit will be twofold:

1. Providing a formal model in which to study linguistic processes;

2. Inspiring new avenues of research in MAPs.

While there has been much work on understanding the impact of communication constraints on agreement there has been relatively little work on understanding agreement in complex state spaces. This is a critical aspect of being able to model more complex social systems, like language.

1.4 Methodology

This dissertation has two parts; in the first we develop a framework to organize the realm of agreement problems. The second part focuses on understanding some of the fundamental relationships between information gathering, information use and time to agreement.

To create the framework we used a comparative approach where we surveyed numerous agreement problems from a variety of domains in order to synthesize a common conceptual framework.

The comparative analysis of numerous agreement problems helped to develop an understanding of the variety of approaches taken to understanding agreement problems. Since agreement problems occur in many domains there have been many approaches taken to understanding them and many different types of questions asked about agreement problems. We can divide these into three general questions: (1) What is the probability of a system reaching agreement?; (2) How long does it take to reach agreement?; and (3) What state is agreed upon?

There are basically two approaches to studying these questions, through formal mathematical analysis and through empirical simulation (these two options are not mutually exclusive).

From the statistical physics literature the focus is on understanding time till agreement which is often studied as a dynamical system, that is a system which evolves through time. However, most of the results

[3]In distributed systems *message complexity* captures this element of cost.

are for extremely simple systems ([Sood and Redner, 2005, Sood et al., 2008]).

Economists studying innovation diffusion have used percolation theory to understand the spread of innovations in a system. One of the central questions in this area is identifying which nodes should initially be "innovators" so that the innovation spreads throughout the population [Kleinbergn, 2007, Kempe et al., 2003] – that is they are focused on answering the first question.

As the complexity of the scenarios grows it is harder to apply analytical methods, and thus empirical methods are used. Agent based models are used to determine the answers to all three questions. Empirical methods are used in understanding linguistic agreement [Beule, 2006, Steels and Wellens, 2006, Steels, 2005a] although in recent years there have been formal results on the *naming game* [Baronchelli et al., 2008, Baronchelli et al., 2006, Vylder and Tuyls, 2006, Vylder, 2007].

We are primarily interested in understanding the question of time till agreement and thus we draw from work on dynamical systems. However, a large part of our work is in complex settings where formal analysis is very difficult. Thus, the predominant methodology employed in this dissertation is numerical simulations of multi-agent systems.

1.5 Overview and summary of contributions

We summarize the main contributions of this dissertation below.

1. We demonstrate the centrality and ubiquity of agreement problems by providing a partial atlas of agreement problems in numerous domains. We also focus on identifying agreement problems in linguistic domains.

2. We develop the Distributed Optimal Agreement (DOA) framework – a taxonomic framework for the organization of multi-agent agreement problems around the principles of information gathering and information use.

3. We describe the *Fundamental Agreement Tradeoff* between the cost of information gathering, accuracy of information and time to agreement.

4. We develop the Sampled Majority Vote process as a way of exploring the fundamental agreement tradeoff in the binary state, static complex graph setting by directly modulating the amount of information an agent can gather through a *sampling fraction* parameter.

5. We develop a new metric, the Information-Centric Convergence Cost (ICCC), to measure the total cost of agreement that takes into account the cost of communication and the cost of time *not* spent in

agreement.

6. We show, through the Sampled Majority Vote process, that individual agents do not need much information in order to come to agreement – this result has great bearing on agreement problems in high communication cost situations, such as energy-constrained sensor networks.

7. We argue and show that linguistic convergence is actually an instance of a more general multi-agent agreement problem. We provide examples of linguistic agreement and the properties of various models of linguistic agreement.

8. We develop a model of agreement in the presence of complex constraints called the Distributed Constraint Agreement (DCA) model.

9. We use the DCA model to explore the linguistic phenomenon of chain shifts in vowel spaces. We show that a simple iterative improvement algorithm can lead to chain shifting phenomena.

Together, these contributions move us closer towards understanding the relationship between information gathering, information use, and agreement problems, which is a principal goal of this dissertation.

The organization of this dissertation is as follows.

In chapter 2 we provide a survey of agreement problems from a multitude of domains.

In chapter 3 we start by identifying the fundamental similarities in all agreement problems which we codify as the Generalized Agreement Process. We design a new taxonomic framework – *Distributed Optimal Agreement* (DOA) – as a way of formally capturing some of the basic differences in agreement problems. Through the DOA framework and by characterizing the differences in solution approaches, we can organize a wide variety of agreement problems under the DOA framework.

In chapter 4 we focus our attention on the process of information gathering. This leads to the development of the *fundamental agreement tradeoff* between the effort to gather information, accuracy of the information, and time to agreement inherent in every agreement problem. We develop the Sampled Majority Vote process to explore how cost and information can be manipulated to yield an optimal tradeoff. Through extensive numerical simulations we find the optimal tradeoff between cost and time to agreement in binary state, static complex interaction graph agreement processes.

We develop the Information-Centric Cost Metric that captures the total cost of agreement which includes the cost to gather information and the lost opportunity cost when a system is *not* in agreement.

In chapter 5 we focus our attention on the process of information use – how an agent uses information to change its state. We develop the Distributed Constraint Agreement (DCA) problem to capture the notion of agreement in a complex state space where features of the state constrain each other. We use the DCA problem to model, at a very high level, a process of phonological change called “chain shifting”.

Finally, in chapter 5.9 we summarize our contributions and discuss ways in which this work can be extended.

Chapter 2

A survey of multiagent agreement problems

2.1 Introduction

One major goal of this thesis is to design a taxonomic framework that will describe agreement problems from a variety of domains. In order to find the salient aspects of agreement problems we need to know the field of agreement problems. In this chapter we provide a survey of Multiagent Agreement problems from a variety of domains. We will be making references to parts of this chapter throughout the dissertation.

2.1.1 Related reviews

There are several review articles that provide overviews of specific areas. [Castellano et al., 2007] surveys a significant amount of work from the sociophysics side of things. [Mason et al., 2007] studies social influence processes from the sociological aspect, a part of the work focuses on agreement, although the real idea is to model contrasting opinions.

[Goldstone and Janssen, 2005] reviews agent based models (ABMs) of collective behavior, with particular emphasis on group pattern formation, contagion and cooperation behavior.

[Wagner et al., 2003] provides an excellent (but somewhat dated) review of computational models of language evolution.

2.2 Distributed function calculation

In the Distributed Function Calculation (DFC) problem a set of agents must calculate a function of the values of all agents in the population in a distributed manner.

The setting:

1. A set of nodes call them $\mathcal{X} = \{x_1 \dots x_N\}$;

2. A graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ on which the nodes are arrayed. $\mathcal{E} \subset \mathcal{X} \times \mathcal{X}$;

3. A function $f : \mathbb{R}^N \rightarrow \mathbb{R}$ known to all nodes.

Each node has some value $x_i[t] \in \mathbb{R}$ where t denotes an iteration.

The goal in DFC is that after some time all agents will have the same value and that value is:

$$f(x_0[0], x_1[0], \ldots, x_N[0])$$

.

That is, all agents have a value that is a function of the initial values of the nodes. In much of the work a *linear iteration* scheme is utilized, where:

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in N} w_{ij}x_j[k]$$

.

Every node updates its value based on a weighted sum of its current value and the values of its neighbors ($w_{ij} = 0$ if x_i and x_j are not neighbors) (much of this formulation is from [Sundaram and Hadjicostis, 2008b]). The matrix W defined by $w_{ij} \forall i, j$ is called the weight matrix.

The main goals are to identify weight matrices that provide for fast convergence under different network topologies (sometime the topology will be vary with time, [Olfati-Saber and Murray, 2004]). Work has been done in situations where there is noise in [Sundaram and Hadjicostis, 2008a].

Equation 2.2 indicates the information gathering and use division. Agents get information from all their neighbors in the network, and they use this information to change their own state. Note that the value a node can take is not restricted – it can be any value in $\mathbb{R}$.

Some features of this problem setting:

Time Invariance If the network does not change the system is called *time invariant*. [Olfati-Saber and Murray, 2004] looked at time varying networks, which they called *switching networks*.

Asymptotic convergence Does the system reach asymptotic convergence or convergence in some finite amount of time? Asymptotic results are provided in : [Blondel et al., 2005]

Topology of network Arbitrary, fully connected, etc.

What type is f? Linear or non-linear, constrained in some other way?

Type of updating Linear or non-linear?

Noise Whether agents get a noisy version of their neighbors opinion or a correct version. See [Sundaram and Hadjicostis, 2008a] for situations in which there is noisy updating.

Communication Delay Do nodes get a delayed version of another nodes value? [Olfati-Saber and Murray, 2004]

2.3 Consensus problems

There has been a significant amount of work on what are called *Consensus problems* in distributed systems [Lynch, 1997].

In the distributed commit problem there are a set of processes that are deciding on whether to *commit* or *abort* a transaction. Each process is executing a part of the transaction, and all parts of the transaction must be successful for the transaction to be committed.

The processes can communicate with each other through messages. The processes are part of a synchronous network system, defined by a directed graph $G = (V, E)$. The nodes are processes, and the links represent the communication channels available. The graph G is not necessarily complete, thus communication between all pairs of processes might not be available. In addition both processes and communication channels may fail at any time.

Algorithms on how to solve the distributed commit problem can be found in [Lynch, 1997, Coulouris et al., 2005]. The basic algorithm (two-phase commit) involves agents electing a coordinator agent that aggregates the abort/commit decisions of every process and makes a decision on whether the population of processes should abort or commit.

Some features of this problem setting:

Time Invariance The communication network might vary with time.

Byzantine Behavior Processes might act in a random manner.

Topology of the communication network Arbitrary, fully connected, etc.

Communication Delay There might be a delay in the arrival of a message.

2.4 Sociophysics models

Sociophysics is the use of statistical physics in modeling social behaviors (such as voting, opinion dynamics, and language propagation). There is quite a bit of work in this area and instead of trying to summarize the numerous models I will describe two important models in this domain. This field is also known as *opinion dynamics*.

2.4.1 Voter model and variants

The voter model (and its variants) are a set of simple model that have received significant analysis. The setting is:

1. A set of nodes call them $\mathcal{X} = \{x_1 \dots x_N\}$;

2. A graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ on which the nodes are arrayed. $\mathcal{E} \subset \mathcal{X} \times \mathcal{X}$

The value of a node at time t is denoted by: $x_i[t] \in \{-1, +1\}$. A node can only take on one of the two values

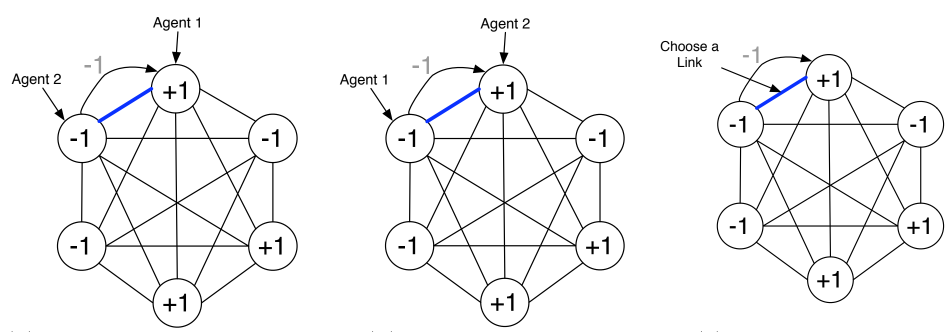
Within this general specification there are several types of dynamics [Sood et al., 2008]:

Voter Process Pick, uniformly randomly, a node i and a neighbor j of node i. Let $x_i[t+1] = x_j[t]$.

Reverse Voter or Invasion Process Pick, uniformly randomly, a node i and a neighbor j of node i. Let $x_j[t+1] = x_i[t]$.

Link Dynamics Pick one edge $(i, j) \in \mathcal{E}$, and randomly decide whether i adopts j's state or the reverse.

Figure 2.1 illustrates these three dynamics. A sizable body of literature has emerged that tries to ascertain the probability of convergence on -1 or $+1$ based on the initial distribution of states, and how long it will take for the system to converge [Sood et al., 2008, Sood and Redner, 2005, Suchecki et al., 2008, Sood, 2007, Suchecki et al., 2004].



(a) Standard Voter Model (b) Invasion Process (c) Link Dynamics

Figure 2.1: Three voter dynamics based on the voter model.

In the voter model each agent can gather information from only a single other neighboring agent – and they can use this information to change to the corresponding state.

Some features of this problem setting:

Topology of network Arbitrary, fully connected, etc.

2.4.2 Bounded confidence models

Bounded Confidence is an aspect of interaction where agents only interact and change their state if they are already somewhat similar to each other. The Deffuant-Weisbuch model is a model that uses bounded confidence. The setting is:

1. A set of nodes call them $\mathcal{X} = \{x_1 \ldots x_N\}$;

2. A set of confidence bounds $\epsilon_1, \epsilon_2 \ldots \epsilon_N > 0$. When $\epsilon_1 = \epsilon_2 \ldots = \epsilon_N > 0$ the model is called homogeneous, otherwise heterogeneous.

3. A norm $||\cdot||$ defined over $\mathbb{R}$.

Each node has some value $x_i[t] \in \mathbb{R}$ where t denotes an iteration (this can be easily extended to the case where nodes can take on values in $\mathbb{R}^d$, here we choose $d = 1$ for ease of exposition).

At each time step t two random agents i, j are chosen. Agent i changes in this manner:

$$x_i[t+1] = \begin{cases} \frac{x_i[t]+x_j[t]}{2} & \text{if } ||x_i[t] - x_j[t]|| \leq \epsilon_i \\ x_i[t] & \text{Otherwise} \end{cases} \tag{2.1}$$

And agent j changes in exactly the same way, except using ϵ_j as the confidence bound.

$$x_j[t+1] = \begin{cases} \mu * (x_j[t] + x_i[t]) & \text{if } ||x_j[t] - x_i[t]|| \leq \epsilon_j \\ x_j[t] & \text{Otherwise} \end{cases} \tag{2.2}$$

Only if the difference between the states of the two agents are below a certain value (the confidence threshold) can the agents influence each other. Figure 2.2 illustrates this process where the two red nodes change in the manner described above, with $\mu = .5$

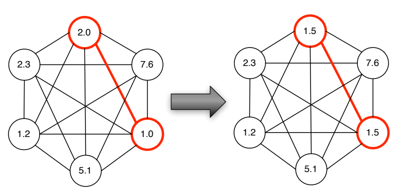


Figure 2.2: Deffuant-Weisbuch Process with $\epsilon = 1.0\ \forall_i$

It has been shown, in the homogeneous case that the system settles to a "limit profile" [Lorenz, 2007], where for every two nodes x_i, x_j either they are equal or else $||x_i - x_j|| \leq \epsilon$.

Many variations on this basic model exist, such as discrete opinions, and allowing for μ to vary. See [Castellano et al., 2007, Subsection III.F] for a good review of several of these variations.

Some features of this problem space:

Topology of the network Arbitrary, fully connected, etc.

State space Discrete or continuous.

Heterogeneous vs. homogeneous confidence thresholds Different confidence thresholds for agents vs. the same threshold for all agents.

2.5 Models of the emergence of norms and conventions

Norms and conventions are collective behavioral restrictions that are an important part of multi-agent systems. The emergence of norms and conventions is important for designing MAS. How do norms and conventions emerge from interactions between agents?

Many of these models utilize a stochastic game framework. The problem setting is:

1. A set of nodes call them $\mathcal{X} = \{x_1 \ldots x_N\}$;
2. A graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ on which the nodes are arrayed. $\mathcal{E} \subset \mathcal{X} \times \mathcal{X}$
3. A set of actions, denoted by $\mathcal{A}$.
4. A payoff matrix, $M : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$.

The payoff matrix identifies the reward an agent gets for executing an action – the reward depends upon the actions of others as well.

An example 2-agent, 2-action payoff matrix:

$$M = \begin{pmatrix} 1,1 & -1,-1 \\ -1,-1 & 1,1 \end{pmatrix} \tag{2.3}$$

.

The top row indicates the actions for the first agent, the column indicates actions for the second agent. Each entry denotes the payoff to agent 1 and agent 2 in that order. The example is called a *coordination* *game* because agents are rewarded for coordinating and executing the same behavior and punished if not. Coordination games were studied extensively in [Lewis, 1969]. Much of the work on coordination games, however, takes place in 2-choice scenarios.

Walker [Walker and Wooldridge, 1995] studied the emergence of conventions in a simulated food gathering situation. Shoham & Tennenholtz [Shoham and Tennenholtz, 1997] provided simulations and analysis in a stochastic games situation. Details of the model are described in 4.2.

[Pujol et al., 2005, Delgado et al., 2003, Delgado, 2002] extended [Shoham and Tennenholtz, 1997] to complex communication graphs – agents were vertices that interacted with only their neighbors.

Some features of this problem setting:

Topology of the network Arbitrary, fully connected, etc.

Decision rule How does the agent make a decision on the action to execute?

2.6 Other models

2.6.1 Innovation diffusion

The study of innovation diffusion – how certain practices spread throughout an organization or population has received a large amount of study. In recent years many formal models and analyses have emerged. The basic situation is once again a graph on which agents are situated as nodes. The edges between agents could be weighted or not, and directed or undirected. A subset of agents are called *innovators* and they start with the value 1, whereas all other agents start with value 0.

On each time step, *every* agent evaluates which state it should be in depending upon the states of its neighbors. Different interaction rules have been specified:

Majority If the majority of an agents neighbors are in state 1, take change to state 1. Otherwise do not.

Linear Threshold Model Calculate the influence on an agent as a weighted sum of the edge weights and the agent value. Each agent switches if the influence is greater than some threshold value.

The key questions asked are, at what threshold value does the value 1 diffuse through the entire population? This is called the "contagion threshold". Which nodes should be initially chosen to be 1 in order for the entire population to eventually take upon the innovation? A set of agents initially set to 1 which diffuse through an entire population are called a "contagion set".

In some cases the *progressive* assumption is made – once agents switch to the new behavior 1, they cannot switch back. However, it is shown that there exits a finite contagious set in the non-progressive case if and only if there is a finite contagious set in the progressive case [Morris, 2000, Kleinbergn, 2007]. This allows one to study the progressive case and apply its insights to the non-progressive case – for a few types of questions only of course. See [Kleinbergn, 2007, Kempe et al., 2003, Morris, 2000, Watts, 2002] for more details.

2.6.2 Epidemics

An epidemic is a situation where a disease spreads through a large fraction of a population. Models of epidemics captures aspects of agreement problems, where agents "agree" to be infected with a disease.

Several of the models are meant to capture the dynamics of epidemics, and thus use epidemic terminology which we will follow as well. We will use the terms "infected" (I) which means an agent has a particular disease/infection, and "susceptible" (S) which means that the agent has the possibility of being infected, although the agent is healthy right now. A "recovered" (R) agent is one that has been infected and cannot be infected again. An epidemic is a situation where an infection spreads throughout a population.

We can view an epidemic as a type of agreement, where the state space is $\{I, S, R\}$. There has been much work on creating abstract models of epidemics.

[Moore and Newman, 2000, Newman and Watts, 1999] study epidemics through percolation theory. The basic idea is that there is a graph where the vertices and edges can be occupied or not occupied. If a vertex is occupied the agent is susceptible to being infected by a neighboring, infected vertex. If an edge is occupied, it means an infection can spread between the vertices. Suppose there is one infected individual in the population, and at each time step the infection spreads to the individuals neighbors, depending upon whether the edges and vertices are occupied. The main question Newman et. al. want to answer is, "For what fraction of occupied vertices/bonds will a giant occupied component occur?" They call this the *percolation threshold* and develop analytical solutions to find this for a variety of graphs. The size of this component indicates the reach of this infection into the population.

[Pastor-Satorras and Vespignani, 2001b, Pastor-Satorras and Vespignani, 2001a] also study epidemics but in a significantly different manner. Imagine now that an agent becomes infected with some probability v if their neighbor is infected, and that an infected individual recovers with probability δ. In this model there is no concept of occupied/unoccupied. Let $\lambda = v/\delta$ be the *effective spreading rate*. The question Pastor-Satorras et. al want to answer is, "For what value of λ will the number of infected individuals be comparable to the size of the entire population?" This value is called the *epidemic threshold*. [Pastor-Satorras and Vespignani, 2001b, Pastor-Satorras and Vespignani, 2001a] show severeal interesting things, such as the fact that there is no epidemic threshold for scale free graphs – infections with very small effective spreading rates can cause an epidemic.

2.6.3 Cultural diffusion

Cultural diffusion models capture the spread of ideas and beliefs (i.e. culture) throughout a population. In [Axelrod, 1997] the author proposes a model of cultural diffusion based on local interaction. A population of agents are arrayed on a 2-D lattice. Every agent has a certain set of beliefs which are modeled as a set of variables that can take on some value from a finite set.

Agents can interact with neighbors who are similar to them, similar to the bounded confidence models described above. In an interaction the agents randomly modify one variable to match. [Axelrod, 1997] studied situations in which multiple groups of homogeneous agents existed side by side.

2.7 Models of linguistic agreement

In recent years there has been great interest in studying the emergence and evolution of a language through computational simulations. One of the major areas of work has been to develop models of the emergence of a shared lexicon – a mapping between a set of words and a set of concepts. This can be viewed as an agreement problem, where the space of possibilities is the space of all possible mappings between words and concepts. Agent interaction provides information about the languages of the interactors, allowing them to modify their language in order to be more communicable.

One of the frequently used paradigms for agent interaction are *language games* [Wittgenstein, 1953]. There are many types of language games, but they usually follow the pattern of having a speaker and a hearer exchanging sentences about the world. Different types of language games can impact the rate of convergence, and the stability of convergence.

As an example, consider the *Observation Game*.

In the observation game (also called the naming game [Vylder and Tuyls, 2006]) the speaker and hearer establish joint attention on some part of the environment. This could be an object, or a complex scene describing several objects together. Regardless, the entity which is agreed upon will be called the *topic*.

The speaker agent produces a sentence that represents the topic. This sentence is passed to the hearer. The hearer determines if their language would produce the same sentence for the same meaning. If the hearers language would do so, the game is successful. If not, the game is a failure.

To play an observation game, it is assumed that both agents are "sharing the same situation, have established joint attention, and share communicative goals" [Steels, 2005b]. In real-world systems, satisfying these assumptions might not be feasible. The effort required to satisfy the assumptions in order to play an observation game places a limit on the number of interactions agents can be involved in.

Much of the work in language evolution has focused on lexicon formation and alignment in a society. The language models employed in these cases are thus very simple and suited for representing the core lexical knowledge in languages but cannot represent compositional languages.

A common approach to modeling a lexicon is as a bidirectional association between a set of words and a set of meanings. An association function is usually defined, between a set of words, a set of meanings, and the natural numbers (called the score). Every word meaning pair is associated with a score which indicates the strength of correlation between the word and meaning. The score can be used to determine which word/meaning to use when given a meaning/word.

Significant work has been with language models of this type. Empirical simulation of lexical agreement include [Steels, 1998, Oliphant and Batali, 1997, Steels and Vogt, 1997, Oudeyer, 1999, Smith, 2004, Vylder and Tuyls, 2006].

A large body of work has emerged that applies evolutionary dynamics work (in particular the replicator-mutator equations) to language evolution. In this body of work language is considered a trait that is passed on to children with some probability of error (that is children will get a different language than their parents) ([Komarova, 2004, Komarova and Nowak, 2001]). The evolutionary dynamics method was also applied to the emergence of grammar ([Nowak et al., 2000, Komarova et al., 2001]).

Cucker, Smale & Zhou provide a machine learning approach to lexicon alignment in [Cucker et al., 2004]. Language is a function whose domain is the set of meanings and the range is a set of linguistic entities. Agents gather examples from other agents in the population, then change to a language that best fits the examples they have gathered. Cucker, Smale & Zhou showed that under certain conditions, including for instance the condition that there are no disjoint sets of agents that do not interact with any other agent, the population of agents would converge to a single language.

Vylder and Tuyls prove that using the naming game with a lexical matrix will result in convergence in [Vylder and Tuyls, 2006].

Baronchelli et. all study lexicon alignment in complex topologies where agents are using a naming game interaction, [Baronchelli et al., 2006, Dall'Asta and Baronchelli, 2006, Dall'Asta et al., 2006a, Dall'Asta et al., 2006b, Baronchelli et al., 2005].

Gmytrasiewicz et. al propose negotiation as the means by which agents can develop an ACL in [Gmytrasiewicz, 2002, Gmytrasiewicz et al., 2002].

Chapter 3

Distributed optimal agreement – A taxonomic framework for organizing agreement problems

3.1 Introduction

Chapter 2 provided a survey of agreement problems from a variety of domains, including control theory, linguistics, sociophysics, etc. In this chapter we set out to synthesize the large number and variety of agreement problems under a common *taxonomic framework* that captures the key similarities and differences between agreement problems.

Before we begin, it pays to be clear on what agreement is. Based on the examples presented in the previous chapter, we view agreement as a state of affairs where a population of agents are "behaving" in a similar way.

What does the term "behaving" mean? We use the term loosely to mean a great many things, from physical movement (in the UAV example) to actions taken by abstract computational processes (such as the action to abort of commit a transaction), to actions taken by humans (such as linguistic behaviors or cultural mores). The key idea is that there is some set of possible behaviors, and through some process a population of agents settles on one (or a few) behaviors to execute. Since the term behaviors connotes some kind of actions, we use the neutral term "state space" to denote the space of behaviors over which agents can agree. Every agent is "in" some state at every point in time – agents can change their state as time progresses.

As we described in the introduction, we are interested in the process by which agreement is reached. That is, how do agents change their state such that after some time agreement has been reached? We are interested in distributed situations – where agents must act on local knowledge in a global, shared environment. We want to define how agents act so that agreement is reached in some finite amount of time.

It is important to note that we are interested in what can be called *on-line and oblivious* systems. By on-line we mean that agents must act with the information at hand at the current time; there is no concept of repeated trials and offline learning. Any learning of the environment by agents must be done as agents act. By oblivious, we mean that agents cannot base their behavior on the identity of other agents. While agents may, and in fact we show that they must, communicate with other agents, no knowledge of the identify of an agent is revealed.

By studying systems that use these two assumptions we can better capture agreement processes in social systems.

The goal of this chapter is to develop an understanding of the fundamental processes underlying agreement problems. We call this set of processes the Generalized Agreement Process(GAP). There are three basic components to the GAPthat we think are necessary for agreement to occur. First, some agents need to be active and to change state. We call this set of agents the *Active Agent set*. Which agents become active can greatly influence the dynamics of agreement as can be seen in evolutionary approaches to agreement.

The second component involves information gathering. The active agent set must communicate with other agents in order to learn about their states. We will show through an example that information gathering is a necessary process in agreement.

Finally, the third component involve information use. The active agents must use the information they gathered in the previous process to change their state.

We claim that the primary differences between the large number of agreement problems outlined in Chapter 2 are oriented around differences in the three processes of the GAP. Primarily, differences in how, and what, information is gathered and how this information is used serve to differentiate between several classes of agreement problems. In section 3.3 we develop the Distributed Optimal Agreement (DOA) framework as a formal way of describing how these processes can vary.

This chapter sets the stage for a detailed and systematic analysis of the processes of information gathering and information use in chapters 4 and 5.

In section 3.5 we show how to map some classic agreement problems into the DOA framework.

Section 3.7 identifies several commonly occurring agreement problems. Definitions via the DOA framework are given.

A principle goal of this thesis is to understand linguistic agreement. Through the DOA framework we can place, in context, the linguistic agreement problems. Section 3.8 identifies three issues that make greatly complicate agreement in the linguistic domain.

Portions of this chapter were previously published as [Lakkaraju and Gasser, 2008b, Lakkaraju and Gasser, 2007] with coauthor Les Gasser.

3.2 The Generalized Agreement Process information gathering and information use

Suppose we have a population of n agents that must choose between a set of states labelled $\{b_0 \dots b_{10}\}$; where each state provides some intrinsic utility to the agent. Initially, each agent is assigned a random state – we will assume that initially the agents are in different states and thus the system is not in agreement. We wish to understand what are the necessary processes that will lead to a situation of agreement.

Clearly, for agreement to occur some agents must change their state. Thus, the first process is the choice of agents to change state. We call this the *Agent Activation* process because a set of agents become "active" and possibly change state.

Now given that some agents are active, what should these agents do in order to reach agreement? Lets first consider the situation in Figure 3.1(a) which graphs the intrinsic utility of the different states. Lets assume that Agent 1 is currently active and is in state b_3. Suppose Agent 1 could calculate the intrinsic utility of its neighboring states, then it would know that state b_4 has a higher intrinsic utility than state b_3. This could motivate Agent 1 to move to b_4. What would happen if all active agents did the same process?

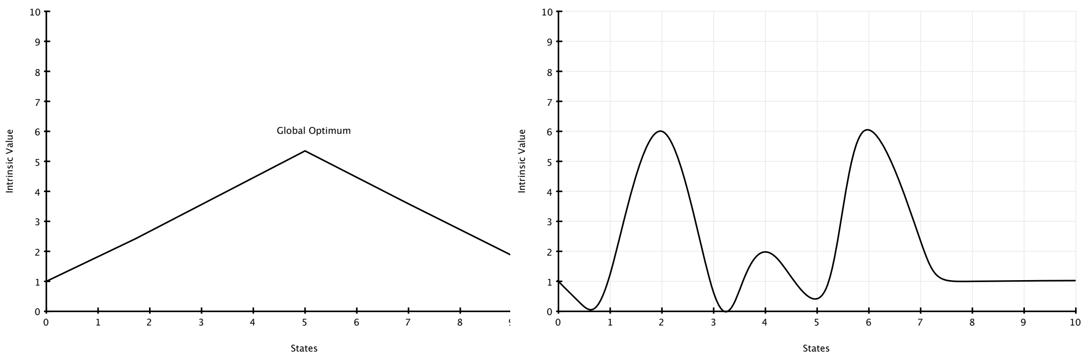
If all agents used some type of gradient ascent algorithm then the entire population would reach agreement by autonomously acting in a manner to maximize their utility. Since b_5 is the state with the highest value all agents would eventually be in that state.

Now consider the situation in Figure 3.1(b), where there are multiple states that have high, and equal, utility. In this case, if each agent followed the same protocol as before and locally optimized their value agreement would most likely not occur. Some of the agents would converge on state b_2 and some of the agents would converge on state b_6 – both of which are equally good in terms of intrinsic value.

The only way to break this impasse is for agents to have access to information about the states of the other agents. That is, the state an agent choose to move to is impacted by the states of other agents. Thus, for agreement to occur, agents must get information about the states of other agents. We call this process *Information Gathering* and we can see that it is necessary once there are multiple equivalent (in terms of intrinsic utility) states.

Once information about the states of other agents is know an agent must make the decision to move to the other states. This is dependent upon several factors. First, how accurate is the information about the states of others? If the accuracy is low one should not change state as readily. Secondly, changing state requires some expenditure of resources. Can an agent afford this expenditure?

The value of a state is a combination of the intrinsic value of a state as well as the *frequency-dependent*



(a) Single Equilibrium (b) Multiple Equilibria

Figure 3.1: Intrinsic value of states $\{b_0 \ldots b_{10}\}$. The x-axis represents the state space and the y-axis represents the intrinsic values of these states.

value. The frequency-dependent value of a state depends upon the number of other agents in that state. Considering Figure 3.1(b) suppose there are many agents in state b_4, but very few in state b_6. Clearly state b_6 is the better state for everyone to agree upon; however, the fact that many agents are in state b_4 might make the value of the state greater than b_6.

Information Use is the process by which an agent chooses the state it should change to by addressing all these issues.

These three components make up what we call the Generalized Agreement Process(GAP) – the basic outline of agent dynamics that lead to agreement. To summarize:

Agent Activation Some subset of the agents are chosen to be *active* at a timestep.

Information Gathering The active agents gather information about the states of other agents in the population.

Information Use The active agents use the information gathered to change their state.

Every agreement problem outlined in Chapter 2 can be cast as an instance of the Generalized Agreement Process, with different constraints on each of the components. Through the Distributed Optimal Agreement (DOA) Framework we provide a taxonomic framework that highlights the differences and similarities between different agreement problems.

3.2.1 Emergence of agreement

This dissertation combines insight from numerous different disciplines. Because of this, it is important to be aware of the significant differences in assumptions in various domains.

We can distinguish between two types of models of agreement phenomena, *intentional* and *unintentional* models. In intentional models agreement is a designed end goal of the system. In unintentional models, agreement is a consequence of actions that are for a different purpose.

As a concrete example, consider the distributed algorithm for agreement described in [Sundaram and Hadjicostis, 2008b]. In this case, there is a multi-phase protocol involving an initial phase where agents learn the graph topology, then another phase to achieve consensus. These agents were designed with agreement in mind, and thus the model of agreement is intentional.

In contrast, consider the agents described in [Steels, 2005a, Steels, 2003, Steels, 1998]. Once again, the goal is to understand the time it takes for the system to reach agreement. However, in this case individual agents are focused on just maximizing their success in communication with neighbors. Agreement is an emergent property of this simple interaction. Agreement is an unintentional consequence of their behavior, which is justified by appealing to its similarity to a human process.

We view agents in [Steels, 2005a, Steels, 2003, Steels, 1998] as *unintentional* – agreement is a property that emerges because of other behavior. The questions to ask are whether the justification for the underlying behavior is valid. The results of work with unintentional models is often to indicate something interesting about the systems which are being modeled.

In intentional models it is more plausible to have complex cognitive processing and multi-phase protocols (such as a phase for choosing a leader, then a phase to exchange information via the leader, as in the protocol to solve the distributed commit problem [Lynch, 1997]).

Both intentional and unintentional systems involve the same core three processes of the Generalized Agreement Process but the plausibility of certain behaviors are different. The distinction between intentional and unintentional systems is exactly the underlying difference between sociological and technical systems.

In this thesis we will view agreement as emergent, even though in some cases we are talking about intentional models.

3.3 The distributed optimal agreement framework

Section 3.2 identified two fundamental processes of agreement. The examples in Section 2 show how variants of these processes have been explored. For instance, we can see in the Distributed Function Calculation problem (see Section 2.2) how the topology of the graph affects the gathering of information. In the emergence of norms work, the agents memory size influences how long information is retained by the agent and thus determines how agents change their state.

We developed the *Distributed Optimal Agreement* framework as a way of categorizing the variety of settings and constraints present in agreement problems. Fundamentally, we are categorizing different constraints and assumptions on the processes of information gathering and use.

The formalization of differences outlined in the DOA model are only some of the possible differences. Based on our survey of agreement problems we found that these issues clearly differentiated between many problems.

We view the process of agreement as a search. Each agent moves about in a *possible agreement space* that comprises a number of *possible agreement states (PAS)*. Any of the PASes might be the substance of an agreement, depending on its own qualities and the number of agents that have settled on it. A *complete agreement* is the condition that all searching agents have arrived at the same PAS. If there is a distance metric on the space of states, a MAP may enjoy the concept of *complete* ϵ*-agreement*, i.e. all agents being within ϵ distance of each other. For example, an *accessibility relation* over the possible activity states allows us to define distance as path length between states, and ϵ as the largest diameter of the accessibility graph for states agents are in. Analogously, a k*-agreement* is the condition that at least k agents have settled on a single state.

Defining a MAP in the DOA framework involves defining the characteristics of the possible agreement space, accessibility relation, solution criteria, and so on. We present the more formal DOA model below.

3.3.1 Formal problem model

An agreement problem in the DOA framework is defined by the 7-tuple:

$$\{A, \Sigma, \Delta, \Theta, \rho, S, \Omega\}$$

where:

1. **Agents:** $A$is a set of N *agents*, $\alpha \in A$. Agents are the active processes in the DOA model, whose actions take place in an interval on a time line T. At each timestep $t \in T$, an agent is said to be "in" some Possible Agreement State (see below).

2. **Possible Agreement Space:** The substance of an agreement in the DOA model is the *possible agreement state (PAS)*, denoted by σ: a state of the world on which agents could agree. For instance, a PAS could be a language an agent chooses to speak, an offer in a negotiation, a candidate strategy for a convention, or a decision to commit/abort a transaction. Some previous work in this area uses the term "strategy" where we use "Possible Agreement State", but we prefer PAS because we aim to

capture many more kinds of agreement than just shared strategy choices. We will use the terms PAS and state interchangeably. $\Sigma$is the set of all PASes, thus $\sigma \in \Sigma$. We use $\sigma_{\alpha_i,t}$ to denote the PAS that agent α_i is "in" at time t.

Configurations

Let Σ^n be the set of all possible associations of PASes with all the agents in A. Σ^n is thus an n-dimensional space. At time t the *configuration* of the entire system is $s_t \in \Sigma^n$—that is, one specific association of all agents with states.

3. **Accessibility Relation:** $\Delta : A \times \Sigma \times \Sigma \rightarrow \{\mathbb{R} \bigcup \infty\}$ is the *accessibility relation* for PASes. $\Delta(\alpha, \sigma_j, \sigma_k)$ describes the (possibly infinite) cost for some agent α_i to move from σ_i to σ_j. $\Delta(\cdot)$ models the structure of the possible agreement space Σ from the perspective of each agent. An agent with more limited capabilities might have a higher cost for changing from one PAS to another, or one PAS might be inherently more difficult (or impossible) to reach directly. For example, representing languages as binary strings and assuming only single point mutations as transition operators [Matsen and Nowak, 2004] results in a hypercube-structured $\Delta(\cdot)$ for the language space.

4. **Interaction Relation:** $\Theta : A \times A \times T \rightarrow \{\mathbb{R} \bigcup \infty\}$ is the *interaction relation.* $\Theta(\alpha_i, \alpha_j, t_i)$ describes the cost for an agent α_i to interact with (e.g. sense, observe, communicate with) some other agent α_j at time $t_i \in T$. Cost is a very general basis for an interaction relation. For example, a close interaction neighborhood for some agent can be defined as the set of agents with which communication is cheap relative to other agents. If cost is inversely related to probability of interaction over time, then $\Theta(\cdot)$ describes agent-to-agent interaction frequencies, and can be used to model a type of *frequency-weighted social network.* In many MAPs the interaction relation is already specified as a graph, where the nodes are agents and the weighted edges reflect the probability with which the agents interact. This is easily represented in the DOA framework. Section 3.6.1 describes how to transform interaction relations to graphs and vice versa.

   The interaction relation is conditioned on time to capture changing topologies of interaction (cf. [Olfati-Saber and Murray, 2004]).

5. **Intrinsic Value:** $\rho : A \times \Sigma \rightarrow \mathbb{R}$ defines the *intrinsic value* of an agent being in a particular PAS. $\rho(\alpha_i, \sigma_j)$ defines the reward agent α_i receives from being in PAS σ_j. $\Sigma(\cdot)$ can be seen as a landscape with hills and valleys corresponding to $\rho(\cdot)$. Since $\rho(\cdot)$ is defined based only on the agent and what state it is using, and not on what states other agents have, we consider $\rho(\cdot)$ to be the *intrinsic* value of

the state with respect to an agent. In many cases $\rho(\cdot)$ is independent of the agent as well. We define $max(\rho)$ as the set $\{(\alpha_i, \sigma_j)\}$ with the highest $\rho(\alpha_i, \sigma_j)$

6. **Starting Configurations** The set of possible starting configurations, $S \subseteq \Sigma^n$. $s_0 \in S$ is the initial configuration of the population.

7. **Termination Configurations** The set of possible termination configurations, $\Omega \subseteq \Sigma^n$. There are many types of termination configurations. Here are several interesting ones:

Simple Consensus Configurations in Ω are *agreements*. A *complete agreement* is formed by a set of agents all being "in" the same PAS, for example all choosing to subscribe to a *particular* language, negotiation offer, convention strategy, etc. This is denoted as a configuration with the following property:

$$s \ni \forall i, j, \sigma_{\alpha_i,t} = \sigma_{\alpha_j,t} \tag{3.1}$$

Other consensus-oriented configuration types include those for the ϵ- and k- agreements as described informally above.

Consensus+Optimization At some t $\sigma_{\alpha_i,t} = \sigma_{\alpha_j,t}, \ \forall i, j$ and $(\alpha_0, \sigma_{\alpha_0,t}) \in max(\rho)$. This is the set of configurations in which every agent is in the same state, and that state has the highest intrinsic value.

Consensus+Computation Given a function $\chi : \Sigma^n \rightarrow \Sigma$, at some t, $\sigma_{\alpha_i,t} = \chi(s_0), \ \forall \ i$. The set of configurations where every agent is in the same state, and that specific state is a function of the initial states of the entire population.

3.3.2 Generalized agreement process in DOA

We can specify the GAP more formally via the DOA. Solving an instance of a DOA problem involves specifying the behavior of the agents such that the system moves from a configuration $s_0 \in S$ to a configuration $s_\omega \in \Omega$ in some finite amount of time.

The specification of the Generalized Agreement Process by a model constitutes an *agreement protocol*. Thus through the DOA we can specify an agreement problem by first defining an agreement setting above, then specifying a protocol below.

We assume a turn-based system, where at each time step t the three-step process of active agent selection, information gathering, and information use occurs, as follows:

1. **Agent Activation**: A subset $C_t \subset A$ (called the *active agent set*) becomes active for this timestep.

2. Information Gathering: Information (call it ψ) is necessary for efficient search. Complete information about a system configuration is costly, being influenced by Θ, N, and $|\Sigma|$ and the history of activity represented by T. Thus strategic selection of information sources at each time step is necessary in a MAP. Once this choice is made, the actual interactions occur which provide the actual information.

2a. Interaction Choice Each active agent $\alpha_i \in C_t$ chooses some other subset of agents $I_{i,t} \subset A$ (called the *interaction set* of α_i), from which to gather information about the current configuration. This substep is purely the choice of a set of other agents to observe or communicate with. Let I_t be the set of all agents in any agents interaction set.

2b. Interaction) α_i interacts with the agent(s) in $I_{i,t}$ via an *Information Gathering Interaction* (IGI). This interaction produces some *information* for α_i about the current configuration. In some cases, the IGI provides information to the agents in $I_{i,t}$ as well.

3. Information Use Finally, all active agents and agents involved in interactions get the opportunity to change state by applying a *decision process.* The decision process is influenced by the information an agent has, ψ, the cost of changing states, Δ, the utility of different states, $\rho$and inherent limitations in the capability of agents to change state. The result of the decision process is a state that the agent could move towards. After the agent has moved, a reward based on $\rho$and the number of agents in the current state might be provided to the agent.

Agent activation

In the agent activation stage certain agents become active, these comprise the active agent set. This is a synchronous timing model – we assume that the active agents take steps simultaneously and at the same speed [Lynch, 1997]. We consider this a reasonable approximation for situations in where the time between interactions is much greater than the time to execute an interaction.

We use terminology that seems to indicate a centralized controller – i.e. some entity that decides who should be active. This is merely an easy way of describing the process and does not reflect some underlying centralization.

In this discrete-time model we must decide which agents are active at each time step.

There are three options for this choice: Random, State-based, or Complete.

In Random activation a subset of agents is chosen at random from the population. This is the most often used model of agent activation. For instance, modeling physical systems (such as the voter model [Sood et al., 2008, Sood and Redner, 2005]) and the evolution of norms and conventions and stochastic games models [Shoham and Tennenholtz, 1997])

In Complete agent activation all agents are chosen to be active at each time step. An example of this kind of activation occurs in Particle Swarm Optimization (PSO) systems ([Kennedy and Eberhart, 1995]).

Finally, in State-based activation, an agent or set of agents are chosen for activation based on some attribute of the state they are in. For instance, the probability of choosing an agent might be proportional to the intrinsic value of the agent's state. The system described in Lieberman et. al., ([Lieberman et al., 2005]) exhibited this property - active agents were chosen according to the intrinsic value of the state they were in.

Information gathering

Once the set of agents C_t has been chosen, each active agent gathers information from other agents in the population. There are two issues: which other agents are accessible for information, and what types of information can be gathered. We describe this process at a high level here, chapter 4 goes into more detail.

We define an information gathering event as an *interaction*, and it is governed by the interaction relation Θ. The decision of what agents to interact with is influenced by the interaction cost. In much of the literature, interaction cost is implemented as a social network in which vertices are agents and edges denote the probability of interactions between the agents at their ends. In some MAP work ([Shoham and Tennenholtz, 1997], [Delgado, 2002], [Pujol et al., 2005]) agents choose interaction sets with a neighbor in their interaction relation, where neighbor-ness is defined by the weight of the edge between them.

On the other hand, in PSO system and ([Olfati-Saber and Murray, 2004]) agents interact with all of their neighbors. In this case, the edge weights do not indicate the probability of interaction, but rather the degree of influence of one agent on another.

Note that if the social network is not defined we assume that it is complete. Thus a situation where an agent picks some other agent at random from the population can still be modeled as a random choice from its neighborhood - which just happens to be every other agent in the population.

Once an agent decides on an interaction set, an interaction will take place. The purpose of an interaction between two or more agents is for the agents to gain information about each others' states. The information could be direct knowledge of the agents state, as in PSO systems or [Olfati-Saber and Murray, 2004], or it could be based on a task that the agents must do. This latter case was modeled in [Shoham and Tennenholtz, 1997] for instance.

In PSO systems and the systems studied by Olfati-Saber et. al., Σ is usually continuous - oftentimes it is the space of reals, $\mathbb{R}$. In this case direct knowledge of the strategy of the other agents can allow agents to find the "average" or "center" strategy.

In these two stages the goal of an agent is to form an estimate of the state of the entire system. Since this is impossible, the agent relies on an approximation. This approximation is influenced by two factors, who an agent interacts with and the information that can be garnered from an interaction. The more potential agents to interact with provides a better estimate of the state of the entire system. The interaction between the interaction set, interactions, information and agreement time is explored extensively in chapter 4.

Information usage

In the final step, all agents that were active in this time step (including both C_t and I_t) have the opportunity to change their state. We call this a *decision process* or *decision rul* because an agent will decide which state to move to. This process will be described in extensive detail in Chapter 5 so we provide only a summary of the main insights here.

The decision process addresses three questions:

1. What are the states of the other agents in the population?

2. How much effort does it require to change state?

3. Should the agent change state (intrinsic vs. frequency-dependent value)?

The first question is tied with the information gathering process and depends upon the information the agent gets.

The second question is influenced by the inherent capabilities of an agent to change state, which is partly captured by the accessibility relation Δ. In chapter 5 we describe the concept of bounded effort – a formalization of an agents *bounded rationality*. Bounded effort essentially constrains the movement of an agent in the state space.

The third question focuses on the reward an agent could get from changing state. As we described in Section 3.2, an agent might choose to stay in a certain state because it provides more intrinsic reward than another state.

At the end of the information use process a reward is provided that is based on the value of the state.

3.4 Formal descriptions of the central questions in understanding agreement

In Section 1.2 we described three questions that a general theory of agreement should answer. Via the DOA framework we can phrase these questions more formally. The three questions are listed below.

1. How do restrictions on information gathering and use impact agreement? What types of constraints will allow for agreement, and which disallow it completely?

2. How do restrictions on information gathering and use impact the time till agreement?

3. How do restrictions on information gathering and use impact the state that is agreed upon?

Firstly, what are restrictions on information gathering and use? In Chapters 4 and 5 we discuss the two processes of information gathering and information use in detail and lay out a space of differences. While a full exploration of these processes is left to the respective chapters we begin to lay out differences in the interaction relation in Section 3.6. Interaction restrictions are one of the most well studied aspects of agreement and thus deserve this special attention.

What does it mean for a system to be in agreement? In the introduction we described this as agents executing the same (or very similar) behaviors. Agreement is specified in the DOA as agents being in the same state at the same time. To answer the first question we want to calculate the probability, given particular restrictions on information gathering and use, of the system resulting in agreement.

It is important to understand the definition of time in the DOA model. We view time as one run through the GAP. So it involves one set of active agents who interact and possibly change their state. This is considered 1 timestep. In other work, predominantly from the domain of computational statistical physics, time is defined slightly differently. For instance, in [Sood et al., 2008, Sood and Redner, 2005] time is defined as N runs of the GAP, where N is the number of agents in the population. In their simulations there is only one active agent chosen, so on average every timestep (composed of N runs of the GAP) every agent is chosen once as an active agent. We must be careful to make sure we understand how time is defined in different research areas. In this thesis we convert all results to our notion of time, that of one GAP per timestep.

Thus, our notion of time till agreement is the number of runs of the GAP's till the population moves from a state in S to a state in Ω.

We not only want to know how long it takes for agreement to occur, but also to which particular state (or states in the case of ϵ or k agreement) the agents agree upon. This would be primarily impacted by the topology of the state space, that is the accessibility relation.

The agent activation process also has a large impact on agreement. State based agent activation is related to evolutionary dynamics where the state space is the space of genomes and the intrinsic value of a state is a fitness measure. Similar to [Lieberman et al., 2005] we can view agents being situated on a graph where a fitter agent can export its state to one of its neighbors. In this case an agent is active with probability proportional to its fitness. [Lieberman et al., 2005] studies the fixation probability – the probability that one state takes over the population. Fixation is clearly equivalent to agreement.

Because of the dramatic impact different activation methods have on agreement we consider agreement with random, state-based and complete activation to be significantly different problems. In this work we focus on random activation methods and leave state-based and complete activation methods to future work.

3.5 Examples of agreement problems in the DOA framework

In this section we show how to model the stochastic games framework described in [Shoham and Tennenholtz, 1997] in the DOA framework.

Shoham & Tennenholtz were interested in the emergence of *social conventions*. A social convention, as defined in [Shoham and Tennenholtz, 1997], is:

> A social law that restricts the agents' behavior to one particular strategy is called a *(social) convention*.

(emphasis in the original).

A social convention is agreed upon by everyone in the society. Thus it is an instance of an multi-agent agreement problem.

Shoham & Tennenholtz use the framework of *stochastic games* to explore the emergence of social conventions. At each time step two agents are chosen. The two agents play a 2-person-2-choice symmetric game. The agents can choose between two strategies, 0 and 1. The payoff matrix for this game is:

$$M = \begin{pmatrix} 1,1 & -1,-1 \\ -1,-1 & 1,1 \end{pmatrix} \tag{3.2}$$

.

The rows and columns correspond to the strategies, 0 or 1 that the two agents play. For instance, the entry in the top left hand corner of the payoff matrix indicates that both agents played the strategy 0. The tuple in the entry indicates the reward that the first and second agent (respectively) gain from playing that strategy. When both agents use the same strategy (the top left and bottom right corners) both players receive a positive reward. When the strategies played by the players differ (the top right and bottom left corners) both agents receive a negative reward. This payoff matrix is called a *coordination game*.

Each agent used the *Highest Cumulative Reward* (HCR) rule to determine whether to change its strategy.

Each agent has a memory that allowed it to keep track of its last k strategies and the payoff each strategy received.

According to the HCR rule, an agent changes its strategy when the total reward in the past m steps for that strategy is greater than the total reward in the last m steps for the current strategy the agent is using. Shoham & Tennenholtz show, for a particular set of payoff matrices, that a population of agents using the HCR rule will reach a social convention, and stay in the social convention.

Mapping the Shoham & Tennenholtz model into the DOA framework is straightforward. First the set of agents, A is just the population of agents in S&T.

The possible agreement space, Σ is just the space of the two strategies that the agents can play. We will label them 0 and 1: $\Sigma = \{0, 1\}$.

There are no restrictions on the accessibility between states for all agents. Thus the accessibility relation will map every pair of states to 0 for every agent α_i. $\Delta(\alpha_i, \sigma_i, \sigma_j) = 0 \ \forall \ \alpha_i \sigma_i, \sigma_j$

There are no restrictions on who an agent can interact with throughout the simulation. Thus the interaction relation will specify the same, 0, cost for every pair of agents at all points in time. $\Theta(\alpha_i, \alpha_j, t) = 0 \ \forall \alpha_i, \alpha_j, t$. This corresponds to an interaction graph that is complete - every agent can interact with every other agent.

The reward an agent gets in a timestep is entirely dependent upon whether the two agents are using the same state or a different state. Thus there is no intrinsic value to a state. All that matters is that the two agents agree on a state. Thus the intrinsic value function will specify the same value, 0, for every possible agreement state and agent. $\rho(\alpha_i, \sigma) = 0 \ \forall \alpha_i, \sigma$.

The population of agents are initialized to random strategies, thus the set of start configuration encompass all possible configurations.

The termination configurations are the states where all the agents agree upon the same strategy.

The protocol agents follow can be placed in the GAP framework.

The process of information gathering is somewhat obscured by the HCR rule. Through the interaction an agent gets some reward based on the payoff matrix and the state of the agent. In the 2 strategy case if a positive reward is achieved the agent knows the state of the other agent – thus we can view this as an agent getting direct information about the state of the other agent.

The memory each agent has is a way to keep track of the state that is the majority in the population. An agent only changes state when it knows that over the last m iterations the state it is in was not encountered as many times as the opposite state.

Agent Activation Two agents are uniformly random chosen to be active.

Information Gathering Interaction Set Choice Each active agent interacts with the other active agent.

Interaction Both agents play the coordination game – which amounts to them receiving a payoff and learning the state of the other agent.

Information Use Each agent stores the information it receives in its memory and changes state according to the HCR rule.

Note that in our view of the work an agent does not receive any reward. We see this as immaterial to the essential process of agreement as basically the reward is an indicator of the state the other agent is in. Of course, in the more general case where there might be different rewards for different states this would have to change.

3.6 Common restrictions on information gathering and use

In this section we discuss some commons ways that information gathering and use are restricted. A detailed study of both of these processes are provided in their respective chapters, but it is useful to identify common interaction restrictions here as well.

3.6.1 Common interaction restrictions

Modifying the interaction relation is one way in which there can be substantial differences in different agreement problems.

A graph is often used to represent an interaction relation. The vertices of a graph represent agents and an edge between two vertices indicates the possibility of interaction between the two agents. An interaction relation can be converted to a graph and vice-versa.

Note that the interaction relation allows pairs of agents to have an infinite cost for an interaction. This represents the fact that these two agents cannot directly interact with each other.

We can convert any interaction relation into a graph using the algorithm below.

1. Let V be the vertex set. There will be n vertices, one for each agent. v_i will be the vertex for α_i. Let E_t be the set of edges at time t.

2. For each pair of agents α_i and α_j at time t where $\Theta(\alpha_i, \alpha_j, t) < \infty$ do the following:

   (a) Let $c = \Theta(\alpha_i, \alpha_j, t)$

   (b) If a **directed** graph is required, then add edge (v_i, v_j).

(c) if an **undirected** graph is required, then add edges (v_i, v_j) and (v_j, v_i).

(d) If a **weighted directed/undirected** graph is required, add the edge (v_i, v_j) with weight c (for undirected, add (v_j, v_i) with weight c as well). We assume that the weights correspond to cost; if not a suitable mapping from weight to cost must be provided.

This process creates an edge between every pair of vertices whose associated agents have a cost less than infinity. If a weighted graph is desired then the cost is used as a weight. Thus we can construct an interaction graph at every time step $G_t = (V, E_t)$. In cases where the interaction relation does not change with respect to time we drop the subscripts.

In the case of the weighted undirected graph case there is a possibility that $\Theta(\alpha_i, \alpha_j, t) \neq \Theta(\alpha_j, \alpha_i, t)$. If this occurs the transformation to a graph does not accurately represent the interaction relation. One should use a directed graph to represent the interaction relation instead.

We can convert a graph into an interaction relation by reversing the process above – any two nodes with edges between them will have a finite cost assigned to them; for all other pairs of nodes there will be infinite cost.

In the rest of the thesis, unless otherwise mentioned, we will assume that interaction relations are specified as undirected graphs. We will use the terms interaction relation and interaction graph interchangeably, depending upon the context.

Types of graphs

The interaction relation has significant effects on agreement. We discuss these in more detail in chapter 4. In this section we will merely outline several different types of interaction graphs that appear often in literature on agreement problems.

When comparing different graphs we need to know what measures to compare them upon. The following are a few terms that we will use in talking about different graphs.

Degree of a node The number of outgoing and incoming links to a vertex.

Clustering Coefficient The average, over all nodes, of the fraction of a vertices neighbors that are neighbors to each other.

Characteristic Path Length The average length of the shortest paths between every pair of vertices.

Degree Distribution The probability of a vertex having a certain degree.

Different types of graphs have varying qualities. [da F. Costa et al., 2007] provides a review of these measurements and the properties of different graphs. We provide a concise survey.

1 or 2-D Lattice A 1 or 2 dimensional lattice is a graph where every node is connected to its nearest $2 * d$ neighbors. A 1-D lattice is a line, a 2-D lattice is a grid.

***k*-Regular Graphs** Every agent is connected to k other agents.

Scale Free Graphs These are characterized by a power law degree distribution. See [Barabasi and Albert, 1999] for more details.

Small World Graphs These graphs have a high clustering coefficient, but a low characteristic path length. See [Watts and Strogatz, 1998] for more details.

Random Graphs Graph generated by randomly creating edges between nodes.

See [Strogatz, 2001] for a general introduction to many of these graph types and their applications.

3.7 Commonly occurring agreement problems

In this section we identify several commonly occurring agreement problem. Based on our survey of agreement problems we can categorize the vast majority of agreement problems as instances of one of the following problems.

First, define a *complete,constant* interaction relation as:

$$\Theta(\alpha_i, \alpha_j, t) = c_i$$

for $\forall \alpha_i, \alpha_j \in A, t \in T,$ and some real constant c_i. This means that interaction between any two agents at every point in time has the same cost. This can be modeled as a complete graph between all agents.

Define a *uniform* accessibility relation as:

$$\Delta(\alpha, s_i, s_j) = \Delta(\alpha, s_j, s_i) = c$$

for $\forall \alpha \in A, s_i, s_j \in \Sigma$ and some real, non-negative constant c. This means that for all agents, switching from any state to any other state costs the same amount. Oftentimes this cost is 0.0.

Binary Space, Complete Graph (BSC) The "simplest" of all problems. More formally:

- *Agreement Space:* $\Sigma = \{0, 1\}$
- *Accessibility Relation:* Uniform.
- *Interaction Relation:* Complete, constant interaction relation.

In a BSC problem every agent can interact with every other agent, there are no restrictions on what states an agent can be in, and there are only two states. Because of the simplicity of this model there have been a lot of analytical results. This model has been studied in many context: the emergence of norms and conventions [Shoham and Tennenholtz, 1997], majority rule process ([Chen and Redner, 2005b, Chen and Redner, 2005a, Krapivsky and Redner, 2003, Mobilia and Redner, 2003], and voter models on complete graphs [Sood et al., 2008].

Binary Space, Static Complex Graph (BSCG) This is a more complex problem where:

- *Agreement Space:* $\Sigma = \{0, 1\}$
- *Accessibility Relation:* Uniform.
- *Interaction Relation:* $\Theta(\alpha_i, \alpha_j, t_i) = \Theta(\alpha_i, \alpha_j, t_j)\ \forall t_i, t_j$

The BSCG setting increases the complexity from the BSC setting by providing a non-complete but static (does not change over time) interaction graph. We will be studying this problem in detail in section 4.4.2.

Work that focuses on this problem includes: the GSM process [Pujol et al., 2005, Delgado et al., 2003, Delgado, 2002]. Voter models on heterogeneous graphs [Sood et al., 2008, Sood and Redner, 2005], and the majority-voting processes [Lima et al., 2008, Pereira and Moreira, 2005, de Oliveira, 1992].

Continuous Space, Static Complex Graph (CSCG) Very similar to the BSCG problem, except here the space is continuous:

- *Agreement Space:* $\Sigma \subseteq \mathbb{R}$
- *Accessibility Relation:* Uniform.
- *Interaction Relation:* $\Theta(\alpha_i, \alpha_j, t_i) = \Theta(\alpha_i, \alpha_j, t_j)\ \forall t_i, t_j$

Most of the control theory work falls under this problem, such as: [Sundaram and Hadjicostis, 2008a, Sundaram and Hadjicostis, 2008b, Blondel et al., 2005, Xiao and Boyd, 2004].

Binary Space, Static Complex Graph, Progressive Assumption (DSCGPA) A significant amount of work in the innovation diffusion literature falls under this model. The major difference is that the progressive assumption holds: once an agent changes to state 1 it cannot change back to state 0.

- *Agreement Space:* $\Sigma = \{0, 1\}$
- *Accessibility Relation:*

$$\Delta(\alpha, 0, 1) = c_j$$
$$\Delta(\alpha, 1, 0) = \infty$$

- *Interaction Relation:* $\Theta(\alpha_i, \alpha_j, t_i) = \Theta(\alpha_i, \alpha_j, t_j)\ \forall t_i, t_j$

See [Kossinets et al., 2008, Kleinbergn, 2007, Kempe et al., 2003].

3.8 Aspects of linguistic agreement problems

A language is useless unless it is shared. Individuals and subgroups modify languages by adding new words, creating new grammatical constructions, etc., and propagating these changes through contact. To maintain communicability over time, the population as a whole must converge (possibly within some small diversity limit) to agreement on a "common" language.

As we argued in the introduction, we can view this process as a multiagent agreement problem — individual agents, each in its own state (e.g., speaking some language), change state through interaction to better match the states of others, with the desired end configuration being all agents converged to the same state. The *language agreement* problem (how a population of initially linguistically diverse agents agrees on a single language) is clearly a MAP – the agents' states are their languages and agents change states via learning from communicative interactions.

We suggest that most current MAP models are not applicable to language agreement problems because they do not account for three issues: the *complexity* of language, the *limited information* of language via interaction, and the *large potential agreement space* for language agreement..

Before existing, powerful work in MAPs can be applied to language agreement, MAP models must be extended to account for these properties. We describe each issue below.

3.8.1 Three issues that make linguistic agreement difficult

After a detailed survey of agreement problems in multiple domains we have identified three issues that make linguistic agreement difficult.

Large possible agreement space

For linguistic agreement, the PAS is the set of possible languages that agents could speak; agreement means speaking the *same* language from this space. This could be an extremely large space of possibilities. In most current MAP models the agreement space is assumed to be discrete and very small (e.g. $\{0, 1\}$ in [Shoham and Tennenholtz, 1997]). Clearly for language agreement problems, MAP models must handle very large agreement spaces.

Complex possible agreement space

Most current MAP models assume that agents are trying to agree upon one state from a set of unstructured possibilities. Clearly language is a structured, complex entity in which links between components are crucial.

We view a language as made up of at least three components: *meanings*, *grammar*, and *lexicon*. Meanings comprise all the issues that can be expressed. The lexicon contains relationships between lexical items and meanings. Grammar specifies how to compose lexemes, and how sentential structure expresses semantic information. These three components are interlinked, and changing one of them can have a great effect on the other components and on communicability with other agents.

For instance, changing the order in which particular semantic roles are expressed (e.g., SVO vs SOV) will have a large affect on communicability, but changing a lexicon might have a more limited effect since some lexical properties can be inferred from grammar.

Limited information gathering

Most MAPs assume that agents can unambiguously determine the state of other agents through interaction. However, for the case of language, where "state" means "language spoken," this assumption does not hold.

In the language agreement problem agents often interact by playing *language games*. There are a variety of games, and they allow two agents to exchange information about their respective languages. The information content of these exchanges is always language *samples*, and they are used by hearers to infer properties of speakers' languages. The number of samples is limited, and in general insufficient to completely determine the speaker's language. Thus agents have *limited discernibility* of others' states—their languages. This is insufficient to satisfy the typical MAP criterion of complete state discernibility.

3.9 Conclusions

Chapter 2 provided a detailed survey of a variety of agreement problems. In this chapter we identified the three processes that underlie all agreement problems, that of:

- Agent Activation.
- Information gathering.
- Information use.

The main contribution of this chapter was to describe the *Distributed Optimal Agreement* framework, which is a formalization of the constraints on information gathering and information use. We developed the DOA and also provided examples of settings that are common in the literature.

Through the DOA framework we can organize the various agreement problems and provide comparisons. Based on our systematic organization of agreement problems we found that there are three main issues in linguistic agreement problems that are not addressed in most models of MAPs. These are:

Large Agreement Space The number of possible languages to agree upon is extremely large.

Complex Agreement Space Elements of language (i.e. grammar, lexicon etc) interact and constrain each other.

Incomplete Information Agents do not get complete information about the languages of others.

In order to develop models of language convergence these issues need to be addressed.

Chapter 4

Information gathering: the tradeoff between information, effort and time to convergence

4.1 Introduction

In Chapter 3 we delineated two axes upon which we can classify agreement problems, *Information Gathering* and *Information Use*. In this chapter we will focus on the process of information gathering and its impact on time to agreement. In the Chapter 5 we focus our attention on the process of information use and agreement in complex state spaces.

Section 4.2 develops an extended example of a simple stochastic game framework (similar to the one employed in ([Shoham and Tennenholtz, 1997]). We use this lengthy example in order to provide an intuition about how information (or the lack thereof) affects agreement. We identify two different types of information, Single agent state information and aggregate state information. We show how aggregate state information is necessary for an agent to maximize reward.

Once we have identified *what* information an agent needs, one must understand *how* an agent gets information. We define the information gathering process as composed of multiple *Information Gathering Interactions* (IGIs) that were introduced in Chapter 3.3. We discuss what an IGI is, what information can be gained from an IGI and how this information might be erroneous.

An IGI can be characterized along two axes, effort – the amount of resources that are expended to execute an IGI; and accuracy – the veracity of the information provided by the IGI. We describe these two axes in more detail in Section 4.3.2.

In Section 4.4 we bring together the preceding work and describe the *Fundamental Agreement Tradeoff* – more IGIs means more accurate information, which means quicker time to agreement; however, more IGIs means more effort and thus more resources expended. An understanding of this tradeoff is paramount to developing a general theory of agreement.

We are interested in the answers to these four questions about the fundamental agreement tradeoff:

1. How much effort (resource expenditure) does an IGI require/expend?

2. How does the number of IGIs affect the accuracy of information gathered?

3. How does the accuracy of the information gathered impact time to agreement?

4. What is the best point in the tradeoff?

The rest of the chapter is focused on answering these questions.

To answer the first question, we identify IGIs from several domains and argue that they require expenditure of effort.

We investigate the second through fourth questions by studying the dynamics of a novel agreement protocol that allows for the direct modulation of the effort of an agent by controlling the frequency of IGIs that can be executed. We call this the Sampled Majority Vote (SMV) protocol. The *sampling fraction* (θ) is a real-valued parameter ranging between 0.0 and 1.0 that limits the number of interactions an agent can execute to be between $\max\{1, \lfloor\theta * |N_i|\rfloor\}$, where N_i is the number of neighbors of an agent α_i. By varying θ we can limit the number of IGIs, and thus the effort, an agent expends. This in turn causes inaccuracies in the information gathering phase.

In Section 4.6 we address the second question by identifying the space of inaccuracies that can occur in the SMV system due to limitations on the frequency of IGIs. We calculate the exact probability that inaccurate information will be generated. We show that this probability can vary significantly based on the number of neighbors of an agent and the distribution of states over the neighbors.

In Section 4.7 we address the third question. To answer the third question we provide extensive empirical simulation of the SMV protocol. We show how inaccurate information impacts time till agreement for two types of interaction graphs, complete and scale-free. The two graphs show striking differences in time to agreement as we vary θ.

Finally, to answer the fourth question we start by defining a new metric to capture the fundamental agreement tradeoff. The *Information-Centric Convergence Cost* (ICCC) is a metric for measuring the total cost of achieving agreement – this includes what can be considered the "lost opportunity" cost for time to agreement as well as the cost of expending effort.

Through extensive empirical simulations we calculate the ICCC for the complete and scale-free networks. Once again striking differences are present between the two interaction relations. Surprisingly we find that the best point in the tradeoff is surprisingly low – for complete networks $\theta = 0.1$ works quite well; and for scale-free network $\theta = 0.2$ works well. This result indicates that significant savings in cost can be achieved without sacrificing time to agreement; this will be important for areas that have very high IGI costs (such as energy constrained wireless sensor networks).

Portions of this chapter were previously published as [Lakkaraju and Gasser, 2009b, Lakkaraju and Gasser, 2009a] with coauthors Les Gasser and Samarth Swarup.

4.2 Useful information for agents

Information, as discussed in Chapter 3, is knowledge about components of the agreement problem, such as the states of other agents, the interaction topology etc. In this section we describe the utility of different types of information in an example agreement problem. As we systematically increase the complexity of the agreement problem we see how different types of information is required to achieve agreement.

We will use the stochastic games framework from [Shoham and Tennenholtz, 1997, Lewis, 1969] that is described extensively in Section 3.5. In this model, it is shown that agents trying to maximize their individual payoffs will result in global coordination. We can then use the payoff of an agent as a surrogate for global agreement – the information needed for an agent to maximize its payoff will be the information needed by the agent for agreement.

The model as presented in [Shoham and Tennenholtz, 1997] has these characteristics:

1. > 2 number of agents.

2. A complete interaction relation.

3. Two uniformly randomly chosen active agents per time step.

Under these constraints [Shoham and Tennenholtz, 1997] shows that agreement occurs with very high probability and provided a lower bound for the time till agreement.

We are going start from the simplest situation and systematically add complexity until we arrive at the setting shown above. At each stage we will identify the complexity of the setting and describe the information that will be of high utility for an agent. In some cases, in order to describe the utility of some information we will have to describe how an agent uses the information as well. For ease of exposition we will focus on a single agent, denoted "Agent 1" . We make one other change for expository purposes – only one agent of the pair selected actually changes state in a timestep. This agent is called the active agent. We do not believe this violates any of the results, but merely changes the amount of time till convergence.

We will be varying three parameters. Below we list each parameter and the values that it can take on. Some of the parameter combinations do not provide any insight or are equivalent to other combinations. We will note these as we proceed.

Number of agents In DOA terms this is the size of A, that is N. We look at two situations, when $N = 2$ and $N > 2$.

Interaction relation We study the complete, locally complete, and complex cases. A *locally complete* interaction relation is a situation where at least one agent can interact with all other agents, but there are no guarantee about who other agents can interact with. Complete and complex interaction relations are described in Section 3.6.1.

Agent Activation We study two options. In the first, only Agent 1 is active and will ever be active; this is called the *static activation* setting. In the second, a uniformly randomly chosen agent is active at a time step; this is called the *non-static activation* setting. In the non-static activation setting Agent 1 may or may not be active in a given timestep.

These particular parameters were chosen because they greatly affect the type of information that has high utility for an agent.

We are concerned with what information will allow an agent to maximize its reward over some finite (say n timesteps) period of time. We represent the interaction relation as an undirected, unweighted graph, following the method of construction from Section 3.6.1. In the following, Agent 1 will play coordination games with others in the population. The state Agent 1 chooses to play in the coordination game will only be a function of the information it has – not of the identity of the opposing agent etc. This is just a restatement of the "Obliviousness" assumption described in [Shoham and Tennenholtz, 1997] and in Section 3.5.

Note that in these examples our purpose is to figure out what information an agent needs; we are not (yet) talking about *how* an agent gets this information.

Case 1 – Two Agents, complete interaction, static activation setting

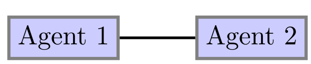


Figure 4.1: Depiction of the two agents, complete interaction relation and fixed setting.

Consider Figure 4.1 where Agent 1 is connected to only one other agent, Agent 2. For Agent 1 to maximize its reward, the state it chooses must match the state of Agent 2. The information Agent 1 needs in order to get a positive reward is information about Agent 2's state. We call this *Single Agent State Information* – information about the state of one other agent. Once Agent 1 gets this information, it will know enough to change its state in order to obtain a positive reward.

To measure the benefit of this information we can calculate the expected reward over n time steps, given the particular setting and the information. In this case and with the single agent state information about Agent 2, Agent 1 will obtain the maximum expected reward of n.

In this situation, how many times will Agent 1 need to gather this information? Since Agent 2 will never be active, and thus never change its state, Agent 1 will only need to gather this information once. We call this *Fixed Single Agent State Information* to emphasize the fact that it needs only to be obtained once in the lifetime of Agent 1.

This situation is exactly the same as the locally complete and complex scenario – since in both cases the interaction network would look exactly like Figure 4.1.

Case 2: Multiple Agents, locally complete, static activation setting.

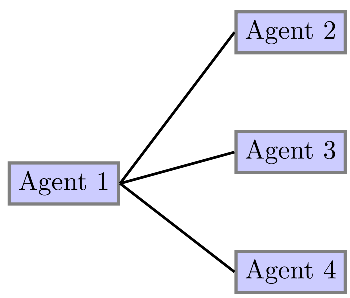


Figure 4.2: Multiple agents interacting.

Consider the situation in Figure 4.2. There are now multiple agents that can interact with Agent 1. At every time step Agent 1 plays the coordination game with one of the other agents picked uniformly randomly.

What information can be useful to Agent 1? The key difficulty here is that Agent 1 does not know which agent it is going to interact with on the current timestep. If it knew this information (say it found out that it is going to interact with Agent 3), then Agent 1 could gather fixed single agent state information and pick the correct state at each time step. However, by the "Obliviousness" assumption this type of protocol (based on the identity of the agent) is not allowed.

Suppose Agent 1 knew the state of only one of its neighbors and chooses to stay in that state for the remaining timesteps. Let us calculate the expected reward for Agent 1 from this option. First, note that since this is the static activation setting, the neighbors of Agent 1 will never change state – it is only Agent 1 that changes.

Let ρ_1 be the proportion of Agent 1s neighbors that are in state 1 and $\rho_0 = 1 - \rho_1$ be the proportion of Agent 1s neighbors who are in state 0. If Agent 1 gets information about the state of one randomly chosen neighbor, then the state of Agent 1 is 1 with probability ρ_1 and 0 with probability ρ_0. The expected reward after 1 timestep is then:

$$
\begin{aligned}
E(1) =& P(\text{Agent 1 in 1})P(s=1) + P(\text{Agent 1 in state 0})P(s=0) \\
=& \rho_1^2 + (1-\rho_1)^2 \\
=& 2\rho_1^2 - 2\rho_1 + 1
\end{aligned}
$$

where s is the state of the neighbor of Agent 1 that is playing the coordination game with Agent 1 at the current timestep.

Since each timestep is independent of the other timesteps the expected reward for Agent 1 after n timesteps is

$$E(n) = n(2\rho_1^2 - 2\rho_1 + 1) \tag{4.1}$$

.

The curve labelled "probabilistic" in Figure 4.3 shows how the expected reward changes based on ρ_1. At a minimum, when $\rho_1 = 0.5$ Agent 1 is basically picking states at random. As ρ_1 approaches 0.0 or 1.0, this protocol reaches its maximum value of n.

Based on only single agent state information, Agent 1 manages to get some reward. However, suppose Agent 1 could get even more information. Instead of having information on one of its neighbors, suppose Agent 1 had information about the distribution of states over its neighbors – that is Agent 1 knows ρ_0 and ρ_1. We call this *Aggregate State Information* – information about the aggregate properties of a set of agents.

The distribution of states over the entire population is a an important quantity – we call it the *Global Aggregate State Information.*

Given this information can Agent 1 gather more reward? Intuitively one would assume so, except that in some cases knowing this information does not help. Suppose that at every timestep Agent 1 chooses behavior 1 with probability ρ_1 and behavior 0 with probability $1 - \rho_1$. Given this decision rule, Agent 1's expected reward after n timesteps is exactly the same as with single agent state information! We will call this the *probabilistic* decision rule.

In fact, for this setting there are three rules that behave exactly the same (in terms of expected reward over n timesteps):

1. Agent 1 gets information about the state of one randomly chosen neighbor and chooses to always be

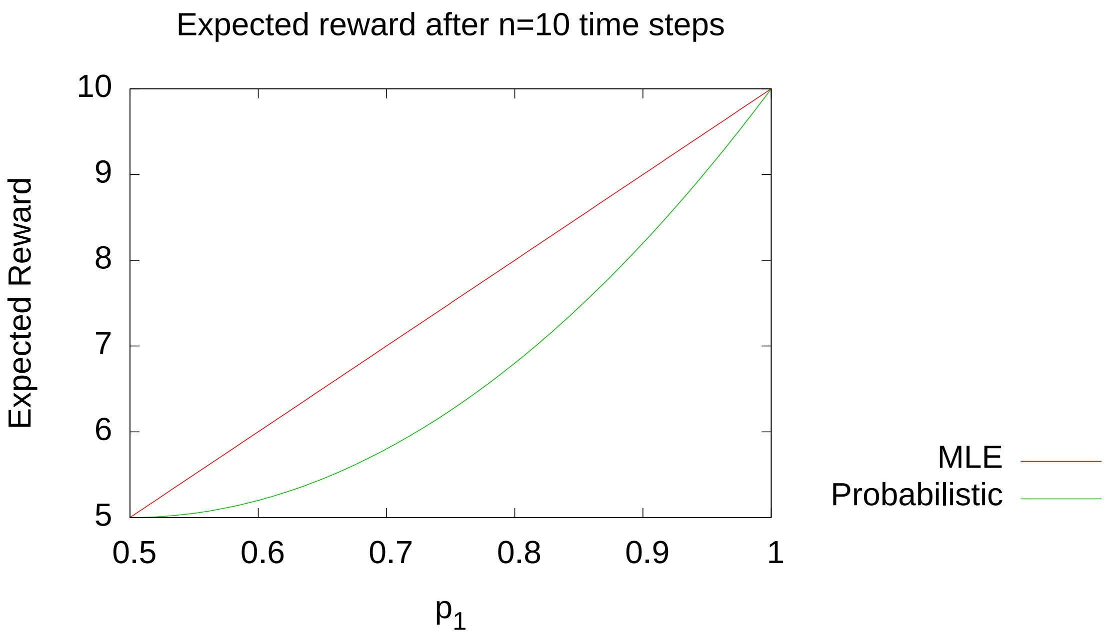


Figure 4.3: Expected Reward for Agent 1 with multiple neighbors.

in that particular state.

2. Every timestep Agent 1 gets information about the state of one randomly chosen neighbor and changes to be in that state.

3. Every timestep Agent 1 choose its state probabilistically, according to the rule given above.

Even though the information Agent 1 has is different in each case, the combination of information and the way it is being used makes the expected reward exactly the same. One of the reasons for this is that we have a static activation setting and thus Agents 2, 3, and 4 do not change between timesteps. Thus, the fact that Agent 1 chooses a new neighbor each timestep does not change the expected reward (clearly there are cases where Agent 1 can choose, randomly, the exact sequence of agents that will then be picked to play against Agent 1 – this case is balanced by the case where Agent 1 always picks the wrong agent). Rule 3 is exactly like rule 2 – instead of picking from its neighbors an agent can just simulate that process by choosing based on the aggregate information.

Note that in rules 1 and 2 Agent 1 is using single agent state information and not aggregate state information; yet the effects are exactly similar.

A better decision rule can be found. Instead of probabilistically choosing its state Agent 1 can choose the maximum likelihood estimate (MLE) state – that is Agent 1 chooses state 0 if $\rho_0 \geq \rho_1$ and behavior 1 otherwise. Without loss of generality let us assume that $\rho_0 < \rho_1$. Then the expected reward on one timestep is:

$$E(1) = P(s = 1) = \rho_1$$

and the expected reward over n timesteps is then:

$$E(n) = n\rho_1 \tag{4.2}$$

.

As Figure 4.3 shows, the MLE does better than the probabilistic rule for values of $\rho_1$far from 0 and 1.

We can see that with aggregate information and the MLE decision rule a high expected reward can be obtained. Once again the setting is fixed information so Agent 1 only needed to obtain information once. Even with aggregate information, the expected reward was reduced for Agent 1, as compared to the single agent state .

The key difference in information is the incorporation of aggregate state information – instead of information about a single agent, Agent 1 used information about the aggregate. As can be seen, in this setting aggregate information combined with the MLE decision rule had the maximal reward. However, for values of $\rho_1 = 0.0, 0.5, 1.0$ the probabilistic and MLE rule worked equally well.

Note that in this case, Agent 1's aggregate state information is very close to the global aggregate state information (the only missing part is the incorporation of Agent 1s state into the distribution).

Case 3: Multiple agents, complex network, static activation setting.

Suppose two more agents are added to the population creating an interaction relation as shown in Figure 4.4.

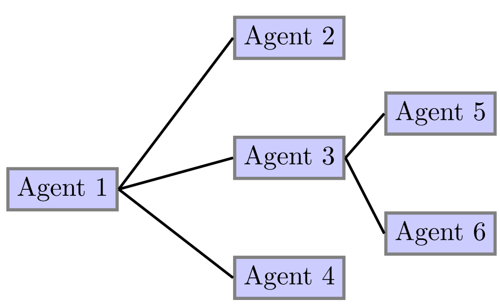


Figure 4.4: Interaction relation for Case 3.

Agent 3 has the potential to interact with both Agent 1 and Agent 5 and 6. Agent 1 is disconnected from agent 5 and 6 and can never interact with them.

In this case, what information does Agent 1 need in order to maximize expected reward? Clearly, the most useful information in Case 3 is the same as Case 4 – even with the addition of 2 agents. What this case points out is that it is not just the number of agents that impacts what information an agent needs, but also the interaction topology.

Suppose Agent 1 had global aggregate information – the density of states overall the agents in the population. Could this information help? Interestingly enough, it would *not* help Agent 1 in maximizing its reward, since Agent 1 only interacts with its neighbors. An MLE based on the global density would possibly lead Agent 1 to a wrong decision. In this case, information about more agents leads to less reward than information about fewer agents.

Case 4: Multiple neighbors, complex topology, non-static activation setting.

Let us extend the system from Case 3 by allowing any two neighboring agents to play the coordination game – this might not include Agent 1.

Once again let us consider the situation from Agent 1s perspective. Unfortunately, this situation is more complicated than the previous one. Even with fixed aggregate state information Agent 1 will not be able to obtain high reward.

The problem is that any fixed information will not reflect the changes that could possibly take place to Agent 3s state. Due to both the structure of the interaction relation and the fact that any two agents can now take part in a game, fixed aggregate information has reduced benefit because it could quickly go out of date as the neighbors change state.

To solve this problem Agent 1 must obtain *dynamic* information – information at each timestep. This can still be aggregate information though.

4.2.1 What do these examples tell us?

The point of the series of increasingly complex examples was to demonstrate how minimal increases in the complexity of the agreement setting affects the information an agent requires to maximize its reward. What can we take away from these examples?

First, we outlined two types of information, single agent state and aggregate agent state. Single agent state information is about a single agent and the agents state. Aggregate agent state information is information about the states of a set of agents. Note that we have not yet defined how aggregate agent state information can be gathered – it can be constructed from several pieces of single agent state information.

Single agent state information was only useful in Case 1 – in all other settings aggregate agent state information was extremely beneficial.

The information that can maximize reward is highly dependent upon the way it is used as well. As Case 2 indicated, single agent state information was equivalent to aggregate state information under certain decision rules.

We can see from these examples that minimal changes in the number of agents and their interaction can drastically change the type of information required and how often this information must be gathered.

4.2.2 Other types of information

In the examples above we assumed that an agent can only get information about another agents state. This is only one type of information (albeit, the most important). Other information an agent could gather includes:

Protocol of an Agent Information about the protocol of an agent – how an agent will change its state based on the information it has.

Interaction relation structure Information about who an agent interacts with.

Accessibility information Information about the way an agent can change its state.

All these pieces of information can be very useful in deciding how to change state. For instance, interaction relation structure information was used in [Swarup et al., 2006, Swarup, 2007] and resulted in quicker agreement than if agents were to randomly select partners.

Interaction relation structure along with single agent state information can be a powerful combination. Consider the interaction topology in figure 4.5. From Agent 7's perspective, single agent state information from either agent 6 or agent 8 is just that, single agent state information. However, if agent 7 knew that agent 6 was connected to a host of other agents (interaction relation structure information), and that agent 6 was using a state change rule that involved the majority state of its neighbors (information about the protocol of an agent) than the single agent state information from agent 6 actually indicates a significant amount of information about the global aggregate state.

In this dissertation our objective is to study how state information can affect agreement time. State information is necessary and sufficient for agreement to occur (see Section 3.2). While the impact of other types of information is important, we leave that to future work.

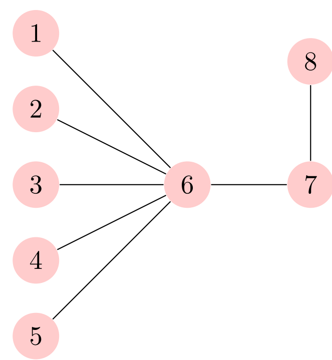


Figure 4.5: Example interaction relation, see text for details.

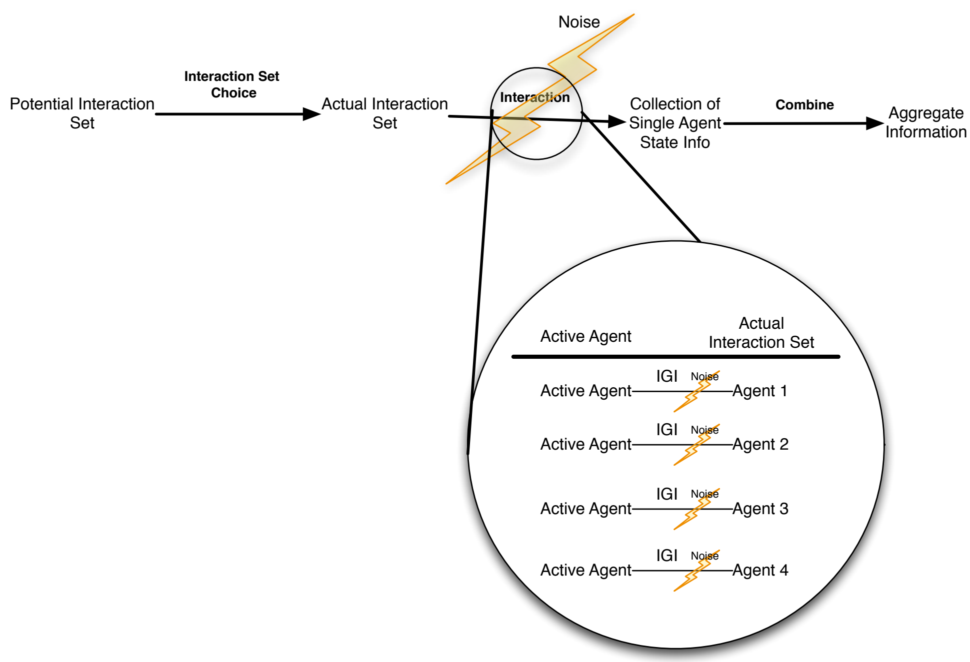


Figure 4.6: Diagram of the Information Gathering process.

4.3 Information gathering via interactions

In the previous section we outlined the different types of information an agent could use to maximize its reward in a simple 2 person coordination game. In this section we address the question of *how* an agent can get this information.

In Chapter 3 we defined the Generalized Agreement Process and described the Information Gathering phase of the agent dynamics. In this section we will delve into the details of this phase and discuss the building block of information gathering, an Information Gathering Interaction (IGI). The properties of IGIs lead directly to a statement of the Fundamental Agreement Tradeoff described in Section 4.4.

Section 3.3.2 formalized the GAP in the context of the DOA framework. There are two major steps:

1. Interaction set choice – Who does the active agent interact with.

2. Interaction – The actual interaction between agents.

To make this concept concrete, consider the simple Voter model (described in detail in Section 2.4.1). At each timestep one active agent "knows" the state of one randomly chosen neighbors. The act of knowing the state of one randomly chosen neighbor can be broken down into two steps:

Interaction Set Choice The choice of a random neighbor to form the interaction set.

Interaction An interaction that results in the active agent obtaining single agent state information.

Consider the system from [Shoham and Tennenholtz, 1997] and described extensively in Section 3.5. At each timestep two agents were chosen to play a coordination game. While technically, these agents received a reward based on their behaviors, we can model this process as an interaction where both agents exchanged information about their states. This holds true when the number of possible states is binary[1]. One play of the coordination game can be considered a single interaction for each agent.

Figure 4.6 is an breakdown of the component phases in the information gathering process that lead to an agent getting state information. We go through each phase in turn.

4.3.1 Interaction set choice

The first step is to choose a subset of agents with whom to interact with. It is important to understand not only who an agent can interact with, but also who an agent *cannot* interact with. The limitations on who an agent can interact with limits the flow of information in the system.

Define $\mathcal{P}_{i,t} \subset A$ as the set of agents agent i could *possibly* interact with at time t; call this the *Potential Interaction Set*. The composition of $\mathcal{P}_{i,t}$ is:

$$\mathcal{P}_{i,t} = \{\alpha_j \mid \Theta(\alpha_i, \alpha_j, t) < \infty\}$$

;

that is, the potential interaction set is the set of all agents that have non-infinite cost. For every active agent, the *actual* interaction set is a subset of this possible interaction set:

$$I_{i,t} \subseteq \mathcal{P}_{i,t}$$

[1] In the latter parts of [Shoham and Tennenholtz, 1997] the authors investigate *quasi-local* update rules where agents explicitly have access to the states of other agents.

.

Essentially, $\mathcal{P}_{i,t}$ defines an interaction relation over the agents. Defining $\mathcal{P}_{i,t}$ for every agent i at time t defines an undirected interaction graph that is equivalent to the one created in Section 3.6. We can divide the space of potential interaction sets into three categories [2]:

Complete Every agent is a neighbor to every other agent.

Regular In a regular graph every agent has the same degree, e.g., a 2-D lattice, or else a 1-D array, of agents.

Complex We use this as a catch all term for any connected graphs that are not complete or regular, e.g. scale free, small world, or random graphs.

Section 3.6 describes why there might be restrictions on the potential interaction set.

There are a variety of ways of choosing the actual interaction set from the potential interaction set:

Random, Single Neighbor Choose one random agent from the potential interaction set.

Random, Subset of Neighbors Choose a random (possibly fixed size) subset of the potential interaction set.

Non-Random Strategic Choose an agent based on learned agent characteristics or based on the state of an agent.

All Neighbors Choose all agents from the potential interaction set.

The "Non-Random Strategic" option represents situations in which the actual interaction set is chosen based on learned characteristics of agents or some other property of the agents. For instance, in [Swarup, 2007] the actual interaction set was chosen based on an agent indegree; in *Bounded Confidence* models the interaction set is chosen based on similarity to the active agents state [Castellano et al., 2007, Subsection F].

The choice of $\mathcal{P}_{i,t}$ and $I_{i,t}$ have a significant effect on agreement time and, for some settings, have been studied extensively. As an example, consider two extremes in binary state settings. In the complete graph voter model [Sood et al., 2008, Sood and Redner, 2005] a single active agent picks one other uniformly randomly chosen agent, gathers single agent state information from the other agent and then changes state to match the other agent. In this case, $\mathcal{P}_{i,t} =$ all agents and $|I_{i,t}| = 1$. For these models expected time to

[2] We are limiting ourselves to the cases where the undirected interaction network is connected. Clearly there are numerous possibilities where the graph can be partitioned into several connected components.

convergence scales as N ([Sood et al., 2008]) , where each unit of time is actually N instances of the GAP process, and N is the number of agents in the system[3].

On the other hand, consider the Majority-rule process described in [Krapivsky and Redner, 2003, Chen and Redner, 2005a, Chen and Redner, 2005b]. At every timestep an odd numbered group (of size G) of agents are uniformly randomly chosen. The majority state of the group is calculated and every agent changes its state to this majority state. Since every agent changes, the active agent set is the group of agents. In this scenario, $\mathcal{P}_{i,t}$ = all agents and $|I_{i,t}| = G$. Expected time to agreement is found to be $(NlnN)/G^{1.5}$.

The expected agreement time is drastically different between these two cases, even though the only change was in $I_{i,t}$. Further on in this chapter we will discuss in more detail how varying $\mathcal{P}_{i,t}$ and $I_{i,t}$ affects time to agreement in the binary state, static complex graph setting.

The motivation for different algorithms for choosing $I_{i,t}$ stems from many considerations that include environmental constraints. Additionally, some constraints are due to the nature of the interaction between agents that results in information, which is the topic of the next section.

4.3.2 From interaction to information

Once $I_{i,t}$ has been chosen the active agent engages in an *information gathering interaction (IGI)* with each agent in $I_{i,t}$. An IGI is an abstraction for a process between two agents that results in an increase in information for at least one of the agents. An IGI is used to represent many things, such as message passing in distributed systems, or disease propagation in contagion models. The important facet of an interaction is that the agents have some information about each others state after the interaction.

Consider the distributed commit problem from [Lynch, 1997]. Each agent sends messages to other agents in the system. Each message can be considered an IGI between the two agents, as it increases the information of one agent. In particular, one of the agents now has single agent state information about the other senders state.

In the linguistic agreement case a *language game* is a type of IGI. As agents interact they receive information about the language of the other agent.

IGIs can be organized around two properties, their *accuracy* and their *effort*.

Accuracy of IGIs

An IGI can result in inaccurate information. For instance, following the distributed commit problem example, if the sender agent exhibited *Byzantine* behavior – that is, unpredictable behavior – then the message that

[3]So technically, in this case time (as in number of instances of the GAP process) to convergence scales with N^2. Our work views time as the number of instances of this process before convergence.

is sent might not actually represent the state of the sender. The following are some types of inaccuracies that can occur with IGIs.

Byzantine Agents The agents taking part in an IGI might be unpredictable, due to perhaps malicious intent, or they might just be faulty in some sense.

Interaction Failure The interaction might not take place, or if it does take place, result in no information. For instance, we can consider sending packets between machines via the UDP as an IGI – since UDP does not guarantee delivery an IGI might fail because a packet might get lost on the way. In contrast, if the IGI involved a TCP/IP session packets can (and do) get lost; however TCP/IP has a method to handle this with sequence numbers and packet acknowledgement.

Erroneous Information During the IGI process information might be modified. Perhaps the transmission medium (for instance, phone lines, or wireless signal) was influenced by some other source.

Partial Information The IGI might provide only partial information about another agents state. This could be due to a limited bandwidth communication channel between the agents, or to inherent limitations on what can be communicated.

This type of issue occurs when each state is a complex entity, such as a language. For instance, consider the Naming game setting described in Section 2.7 where the state of each agent is a mapping between a set of words and meanings. On each interaction the speaker/hearer agents only learn about the meanings of a subset of each others words – each agent only has a partial view of the other agents language. As a model of human linguistic interaction this restriction on communication makes sense – humans do not, and fundamentally cannot, communicate their entire language to each other.

The case of partial information is very important and is described in more detail in Chapter 5.

All of these situations can modify the information agents receive.

Effort for an IGI

An IGI represents some type of interaction and thus it requires resources to execute. We use the term *effort* as a dimensionless quantity that measures the expenditure of resources to execute an IGI. Effort is related to different quantities in different domains.

Consider sensor networks which often have limited energy resource (due to the storage limitations of a battery pack for instance). Thus one must design protocols with energy consumption in mind. Usage of energy is a natural way to measure effort in sensor networks.

It is known that communication in sensor networks requires a lot of energy. In fact the energy required for storing and processing information by the sensor is negligible with respect to the energy required for communication [Williams et al., 2007, Pottie and Kaiser, 2000]. Thus, we would argue that the effort for executing an IGI (which will involve some communication) will be high in a sensor network.

Effort can also relate to social restrictions. For instance, an employee might have the potential to talk to their manager at any point in time – however the manager might be busy and an employee frivolously interacting with a manager would look bad on the employee. Thus each IGI between the employee and manager has an expenditure in terms of the managers respect or attitude towards the employee.

Upon execution of IGIs between the active agent and the agents in $I_{i,t}$ the active agent will have a collection of single agent state information. The next section deals with how single agent state information can be combined to form aggregate agent state information.

4.3.3 Combining information

As we saw in the example in Section 4.2 aggregate information is critical for agreement. The final step in the information gathering process is the combining of multiple single agent state information into aggregate information.

Note that this is an optional step. In some agreement protocols (the Voter model for instance) single agent state information is all that is necessary. However, there are several protocols that utilize aggregate information as well.

To create aggregate state information, an agent combines the single agent state information and calculates the distribution of states based on the information. This combination process might result in inaccurate aggregate information for two reasons:

Inaccurate IGI If the single agent state information from the IGIs was inaccurate, then the resulting aggregate information will also be inaccurate.

Choice of Interaction Set The choice of $I_{i,t}$ might not suffice to provide an accurate description of the aggregate state over all neighbors.

Since $I_{i,t}$ is a subset of the potential interaction set there is the possibility that the distribution of states will not reflect the true distribution of states of an agents neighborhood. This could result in an active agent mistakenly changing state.

This is a fundamental issue with agreement problems, and we discuss it in the next section.

4.4 The fundamental agreement tradeoff

For agreement to occur, an agent must have information – and we laid out what information is needed by an agent in different situations in Section 4.2. To get this information, agents utilize interactions, and we described properties of IGIs in Section 4.3. One of the critical properties of an interaction is effort – a measure of the resource expenditure of an agent in executing an IGI. We argue that there is a clear link between all of these elements.

More interactions means better information which means quicker time to agreement; however more interactions means more effort expended. Fewer interactions means worse information, which means slower convergence; however fewer interactions means less effort expended. Thus, there is a tradeoff between effort expended and agreement time via information. We call this tradeoff the *Fundamental Agreement Tradeoff*.

Consider again the examples of the voter model and the majority rule model from Section 4.3.1. Previously we noted a significant difference in agreement time between the two protocols. We can further differentiate between the two protocols by studying the effort expenditure. Since these are abstract models and not linked to any physical systems we will let effort be a dimensionless quantity and assume that each IGI requires 1 unit of effort.

In the case of the voter model, at each time step one IGI is executed and thus only one unit of effort is expended per timestep. Then the expected total effort expended to reach agreement in the voter model scales as N, because the expected time to agreement scales as N.

In contrast, in the majority-rule model at each time step there are G active agents and each agent needs to calculate aggregate state information from the group. Each agent would then calculate the majority state of the group by interacting with each agent in the group and then changing its state to the majority state. If G is the number of agents in the group, then this process would require G^2 IGIs per time step. The effort expended per time step is then G^2. The expected total effort expended to reach agreement would then scale as $\sqrt{G}NlnN$.

The difference in effort expended is tremendous. While more interactions provides better information and results in quicker agreement time the effort required is much larger than in the voter model. This example illustrates the fundamental agreement tradeoff.

In the majority-rule model G^2 is the upper bound on effort. Aggregate information can be calculated through different mechanisms that possibly require fewer IGIs. For instance the active agent set could elect a leader then have the leader interact with all the agents, calculate the majority state and notify every other agent. In this chapter, though, we restrict our focus to situations in which the interactions are based solely on exchanging information about the state of an agent. This is for three reasons.

First, we are interested in *open, heterogeneous* multi-agent systems where different types of agents can enter and exit the system. In these systems it is not wise to assume what capabilities agents might have. Some agents might have significant capabilities to do processing while others might be simple, reactive agents. By focusing on the simplest possible abilities the results from this work will generalize to many systems.

Second, we are interested in systems that can indicate something about general social systems. While our intention is not to fully model such social systems, we do wish to capture the fundamental ideas. The notion that a group of humans would sit down and elect a leader then change their state based on the leaders information only makes sense for very restricted situations, such as the elections.

Thirdly, leader election is itself an agreement process and requires an agreement protocol. Assuming a leader election algorithm assumes an agreement protocol already exists.

The second critical property of an IGI is accuracy – how accurate is the information that occurs via an interaction. This has a significant impact as well, since one might execute more interactions but not achieve a corresponding increase in the accuracy of information as suggested by the fundamental agreement tradeoff. In this chapter we assume that IGIs are completely accurate; we study situations in which they are not accurate in Chapter 5.

4.4.1 Four questions on the fundamental agreement tradeoff

The fundamental agreement tradeoff is a critical component of information gathering and thus an understanding of the tradeoff is necessary to form a general theory of agreement. To understand the fundamental agreement tradeoff we should answer the four following questions:

How much effort (resource expenditure) does an IGI require/expend?

How does the number of IGIs affect the accuracy of the information?

How does the accuracy of the information impact time to agreement?

What is the best point in the tradeoff?

The rest of the chapter is focused on answering these questions. We focus on a binary state, static complex graph (BSSC) agreement setting (see Section 3.7) for a number of reasons.

First, the setting has been used as a model of many issues, such as the adoption of an innovation or a cultural trait; it also is an important model in physics as it is related to the Ising Spin model.

Secondly, this setting is complex enough to produce interesting behavior and provide insight into general complexities while not being overly complex. In particular, there is a significant amount of analytical work based on the binary state space. By using this setting we will be able to leverage this work for a more formal analysis of the fundamental agreement tradeoff.

Thirdly, and in part due to the first reason, there is a substantial amount of previous work studying this setting. Settings of this type have been investigated in the domains of physics (the voter model, majority-rule model, etc), multi-agent systems (emergence of norms and conventions etc), and innovation diffusion. This allows us to leverage previous results for greater insight into the fundamental agreement tradeoff.

Fourthly, the use of a static and complex interaction graphs is a common assumption in many agreement problems. Several studies have shown that certain types of complex graphs (scale-free and small-world in particular) appear in many different types of data (such as actor collaboration graphs, etc) [Strogatz, 2001]. The static nature of the graph, however, is not an accurate representation of general social systems – oftentimes the potential for interaction changes with time. However, we assume that the time for an interaction graph to change is far longer than the time it takes for agreement to occur – in which case a static graph is an acceptable approximation.

To answer the first question we provide numerous examples of IGIs and an estimate of their effort from a variety of domains.

We investigate the second through fourth questions by studying the dynamics of a novel agreement protocol that allows for the direct modulation of effort expenditure by controlling the number of IGIs that can be executed. We call this the Sampled Majority Vote (SMV) protocol. The *sampling fraction* (θ) is a real-valued parameter ranging between 0.0 and 1.0 that limits the number of interactions an agent can execute to $\max\{1, \lfloor \theta * |N_i| \rfloor\}$, where N_i is the number of neighbors of an agent α_i. By varying θ we can limit the number of IGIs, and thus the effort, an agent expends. This in turn may cause inaccuracies in the information gathering phase.

To answer the second question we identify the space of inaccuracies that can occur in the SMV system due to limitations on the frequency of IGIs. We calculate the probability of committing a *Mistaken Majority* error – where the active agent makes a mistake in ascertaining the majority state of its neighbors. We show that this probability can vary significantly based on the number of neighbors, number of IGIs, and the distribution of states among neighbors.

To answer the third question we provide extensive empirical simulation of the SMV protocol. We show how inaccurate information impacts time till agreement for two types of interaction relations, complete networks and scale-free networks. The two networks show striking differences in their dynamics.

Finally, to answer the third question we start by defining a new metric to capture the fundamental agreement tradeoff. The *Information-Centric Convergence Cost* (ICCC) is a metric for measuring the total cost of achieving agreement – this includes what can be considered the "lost opportunity" cost for time to agreement as well as the cost of expending effort.

Through extensive empirical simulations we calculate the ICCC for the complete and scale-free networks. Once again striking differences are present between the two interaction relations. We find that the best point in the tradeoff is surprisingly low – for complete networks $\theta = 0.1$ works quite well; and for scale-free network $\theta = 0.2$ works well. This result indicates that significant savings in cost can be achieved without sacrificing time to agreement; this will be important for areas that have very high IGI costs (such as energy constrained wireless sensor networks).

Figure 4.7 is an illustrated summary of major works in the BSSC setting organized around $I_{i,t}$ and $\mathcal{P}_{i,t}$. We focus on time to agreement results. Analytical results are highlighted in blue. Some aspects of the parameter space are well studied. In particular, the voter model (the "Sood & Redner" work) has been studied quite extensively. However, we do not, yet, have a complete notion of all the points in the space, nor is there a theory that can tie these results together.

The area highlighted in green is where the Sampled Majority Vote process can provide some insight in order to fill in the gap between single agent choice and all agent choice models.

We can see that there is a significant difference in agreement time as the actual interaction set changes from being composed of a single state to being composed of all neighboring states. [Sood et al., 2008] analytically shows that convergence scales as $O(n^2)$ in the voter model. In comparison, [Delgado, 2002] showed empirically that choosing all your neighbors and applying the majority rule results in agreement time that scales in $O(n)$. Note that the voter model is a degenerate case of the majority rule – when there is only one agent in the actual interaction state its state will always be the majority state.

4.4.2 The sampled majority vote process

The Sampled Majority Vote process is designed for the binary state, static complex graph setting described in Section 3.7. At each time step, the following process is carried out:

1. **Agent Activation:** Let α_i be the active agent, a uniformly randomly chosen agent from the population. Without loss of generality let us assume that α_i is in state 1. Let N_i be the set of neighboring agents of α_i.

2. **Information Gathering: Interaction Set Choice:** Let Π be a uniformly random subset of N_i,

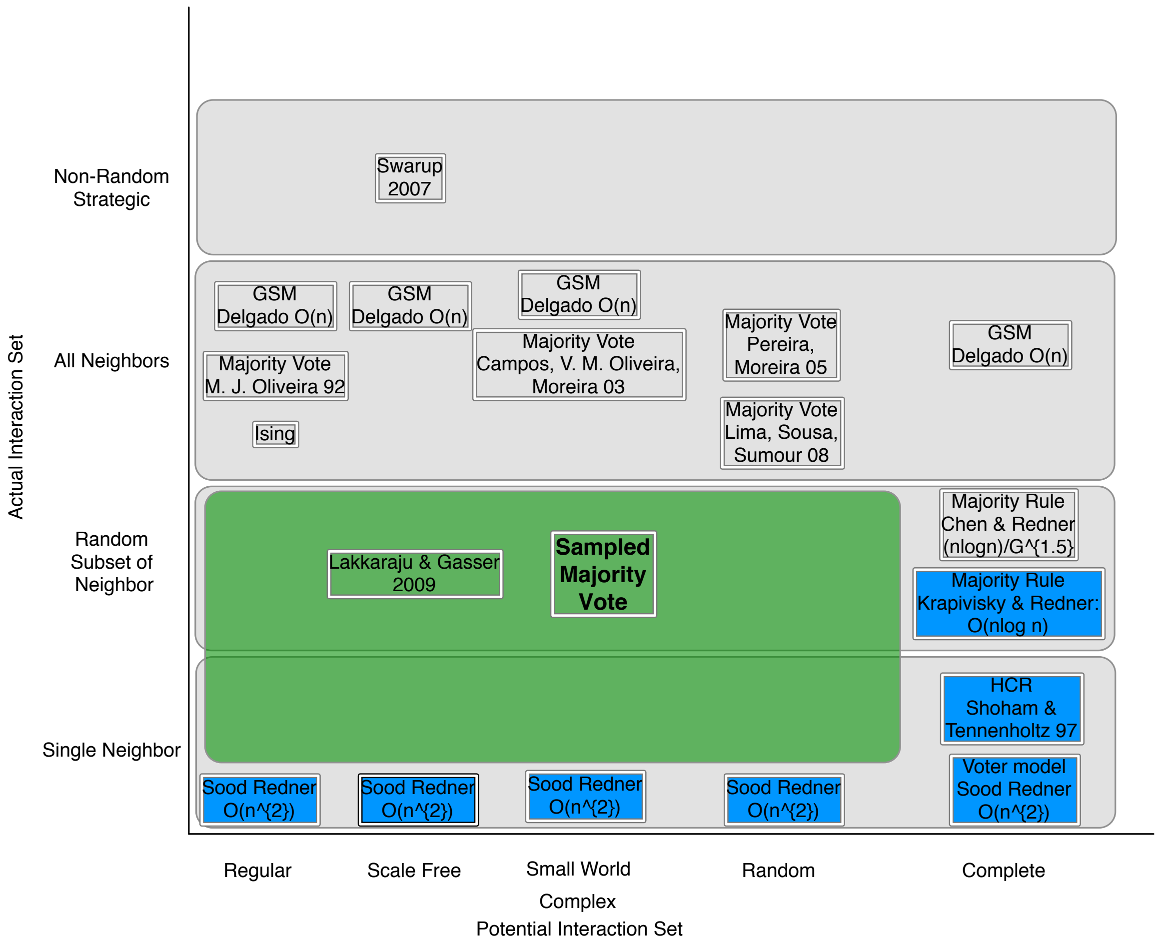


Figure 4.7: The x axis lists possible settings of the potential interaction set; the y axis lists possible settings of the actual interaction set. Blue boxes indicate analytical results. When possible, estimates of time to agreement are indicated, n refers to the number of agents. All times are in terms of number of interaction steps (see discussion in Section 3.4). "Swarup 2007" refers to [Swarup, 2007], "GSM Delgado" refers to [Delgado, 2002], "Majority vote; M.J. Oliveira 92" refers to [de Oliveira, 1992], "Majority Vote; Campos, V. M. Oliveira, Moreira 03" refers to [Campos et al., 2003], "Majority Vote; Pereira, Moreira 05" refers to [Pereira and Moreira, 2005], "Majority Vote; Lima, Sousa, Sumour 08" refers to [Lima et al., 2008], "Majority Rule: Krapivisky & Redner" refers to [Krapivsky and Redner, 2003]; "HCR; Shoham and Tennenholtz 1997" refers to [Shoham and Tennenholtz, 1997], "Sood Redner" refers to [Sood et al., 2008], "Lakkaraju & Gasser 2009" refers to [Lakkaraju and Gasser, 2009b]

chosen without replacement. The size of Π is $\max\{1, \lfloor \theta * |N_i| \rfloor\}$. θ is called the *sampling fraction.* Let π_0 be the number of agents in Π with state 0 – the opposite state.

3. **Information Gathering: Interaction:** α_i obtains the state of each agent in Π .

4. **Information Use:** α_i changes state with probability $f(\frac{\pi_0}{|\Pi|})$

Where:

$$f(x) = \frac{1}{1 + e^{2\beta(1-2x)}} \tag{4.3}$$

Figure 4.8 is a graph of $f(x)$. [Delgado, 2002] refers to $f(\cdot)$ as the *Generalized Simple Majority Rule.*

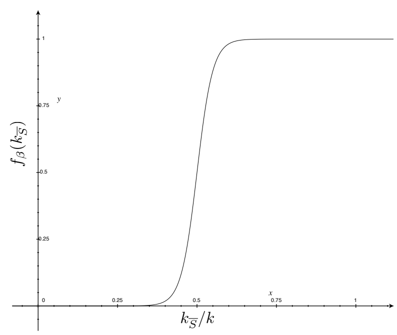


Figure 4.8: Generalized Simple Majority rule. The x-axis is the fraction of neighbors with state $\overline{S}$. The y-axis is the probability an agent will switch. β determines the steepness of the curve. As $\beta \to \infty$ the step function is recovered. In this figure, $\beta = 10$

θ provides a way to control the effort of an agent. When $\theta = 0.0$, the Voter model is recovered. For $\theta = 1.0$ the Generalized Simple Majority process [Delgado, 2002] is recovered. θ is directly related to the amount of effort expended by the system at each time step. The effort is a function of the sampling fraction and the number of neighbors.

Figure 4.9 shows the hypothesized relationship between information accuracy and agreement time as a function of θ. [4]. The x-axis is the sampling fraction, and the y axis represents some measure of the accuracy of the information an agent receives and agreement time. We hypothesize that as the sampling fraction increases, the accuracy of the information increases, while agreement time decreases.

The graphs on the bottom of Figure 4.9 show examples of an active agent and its neighbors for 3 different sampling fractions. When $\theta = 0$, we recover the voter model. When $\theta = 1.0$ we recover the GSM model from [Delgado, 2002, Delgado et al., 2003]. We can view the SMV model as interpolating between the voter and GSM models.

[4] While this is just a hypothesis it is supported by empirical results. See [Lakkaraju and Gasser, 2009b]

Since the voter model is a specific case of the SMV process we can utilize the analysis from voter models for the more general SMV case. Luckily, there are quite a few analyses of voter model dynamics – the methods of [Sood et al., 2008, Sood and Redner, 2005] are particularly applicable.

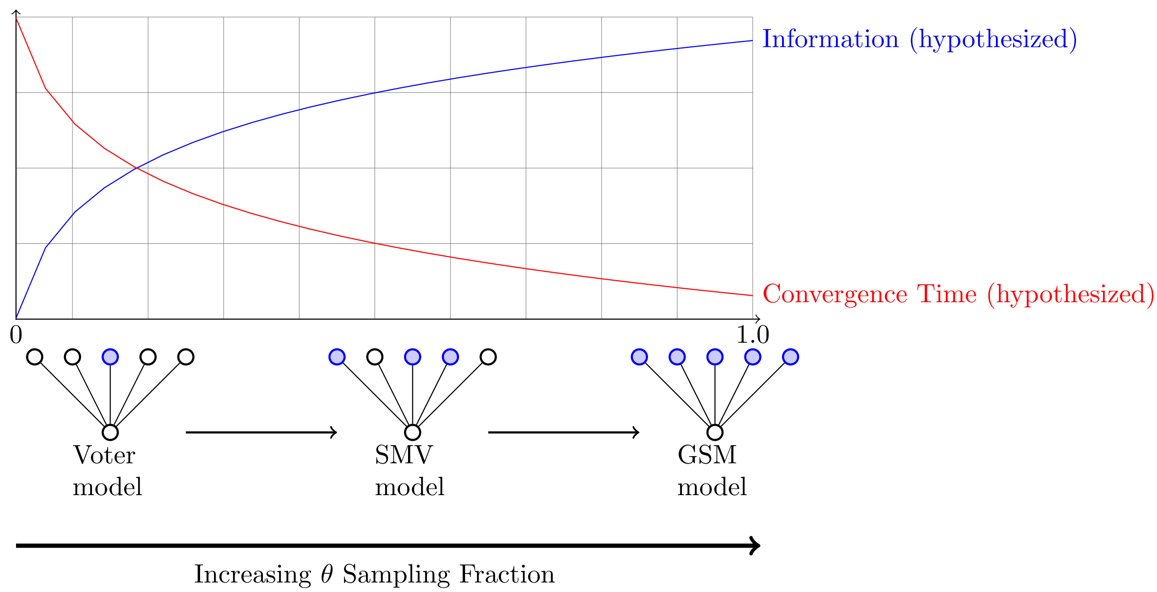


Figure 4.9: Blue nodes are in the sample set. As the sampling fraction increases from 0.0 to 1.0 the accuracy of the aggregate information gathered increases while the time to agreement decreases (hypothesized curves based on empirical evaluation).

4.4.3 Does SMV result in agreement?

When $\theta = 0.0$ we recover the Voter model on heterogeneous graphs. [Sood et al., 2008, Sood and Redner, 2005] show that the voter model results in agreement for heterogeneous degree graphs.

In the $\theta = 1.0$ case, there are quite a few empirical results, but very few analytical results.

The majority-vote process [Lima et al., 2008, Pereira and Moreira, 2005, de Oliveira, 1992] is very similar to the SMV for $\theta = 1.0$ except for one small difference. In the majority-vote process an agent changes to the majority state with probability $1 - q$ and does not take the majority state with probability q. When $q = 0.0$, the majority-vote process is equivalent to the SMV process. It has been shown through extensive experiments that agreement is reached when $q = 0.0$. Numerical simulations have been done on random, scale-free and small world graphs.

In [Pujol et al., 2005, Delgado, 2002, Delgado et al., 2003] the authors develop a process called the “Generalized Simple Majority” process where an agent samples all of its neighbors and changes state with probability based on Equation (4.3). An argument is provided for agreement on heterogeneous graphs. We follow this line to argue for agreement when $0.0 < \theta < 1.0$.

The majority-rule model described in [Chen and Redner, 2005b, Krapivsky and Redner, 2003] is similar to the SMV process on a complete graph. At each time step, an odd number of agents, G are chosen to be active. Every agent in the group changes its state to the majority state of the group.

[Krapivsky and Redner, 2003] analyze majority rule dynamics with $G = 3$ on a complete graph and a one-dimensional lattice. They show that agreement time scales with $Nlog(N)$, where N is the number of agents in the system. [Chen and Redner, 2005b] provided a continuum approximation of the process for $G > 3$, and studied majority-rule dynamics on regular lattices. They found that agreement times scales with $Nlog(N)/G^{1.5}$.

In the following we follow the line of argument from [Pujol et al., 2005, Delgado, 2002, Delgado et al., 2003] to argue that agreement will occur.

First, we will describe the well known *Hypergeometric distribution* [Johnson and Kotz, 1969] which will be useful later. Suppose there is an urn with N balls, of which k are red and $N - k$ are blue. We pick n of these balls *without replacement* and define X to be a random variable denoting the number of red balls picked. We say X follows the *hypergeometric distribution*. We use the notation:

$$h(N, k, n, x) = P(X = x)$$

where:

$$h(N, k, n, x) = \frac{\binom{k}{x}\binom{N-k}{n-x}}{\binom{N}{n}}$$

. Let $Q_H(N, k, n, x) = P(X > x)$ where

$$Q_H(N, k, n, x) = \sum_{y>x} h(N, k, n, y).$$

Let $H_{cdf}(\cdot)$ be the cdf of the hypergeometric distribution:

$$H_{cdf}(x) = \sum_{y=0}^{x} h(N, k, n, x) = 1 - Q_H(N, k, n, x).$$

The mean and variance of the hypergeometric distribution are:

$$E[\mathcal{X}] = n\frac{k}{N}, \tag{4.4}$$

and:

$$Var[\mathcal{X}] = n\frac{k(N-k)(N-n)}{N^2(N-1)} \tag{4.5}$$

from [Johnson and Kotz, 1969].

One useful symmetry is given below.

$$h(N,k,n,x) = \frac{\binom{k}{x}\binom{N-k}{n-x}}{\binom{N}{n}} = \frac{\binom{N-k}{n-x}\binom{N-(N-k)}{n-(n-x)}}{\binom{N}{n}} = h(N,N-k,n,n-x) \tag{4.6}$$

Intuitively, the probability to get x red balls in a sample of size n is the same as picking $n-x$ red balls in a sample of size n with $N-k$ red balls in the population.

We study the dynamics of the SMV process by doing a continuum approximation – we can view the discrete process as a continuous process. Let $\rho_1(t)$ be the fraction of the population that is in state 1 at time t, and let $\rho_0(t) = 1-\rho_1(t)$ be the fraction of the population in state 0 at time t. We use the term ρ_1 when the timestep does not matter. Let us assume that $\rho_1(t)$ is continuous and differentiable at all points. Then by a first order Taylor expansion over t we have:

$$\rho_1(t+\delta t) = \rho_1(t) + \frac{\partial}{\partial t}\rho_1(t)\delta t + O(\delta t^2) \tag{4.7}$$

We ignore the second order term (since we are interested in the case where $\delta t \to 0$. Now we need to estimate the instantaneous change in $\rho_1(t)$. This is difficult, as the instantaneous change is a function of which node is chosen and the distribution of states among that nodes neighbors. To simplify, we utilize the common *mean-field assumption* [Giordano, 1997, 207], which assumes that the distribution of states over an agents neighbors is the same as the distribution of states over the population. For low degree nodes this is not a very accurate approximation, however for higher degree nodes the assumption will be more accurate. While this is a significant assumption it is an approximation that is often used in these types of systems (more generally called "spin systems") see, [Sood et al., 2008, Delgado, 2002]. [Castellano and Pastor-Satorras, 2006] describes situations in which this does not hold.

We make one other assumption here. In the SMV the $f(x)$ is used to determine the probability of switching. In order to simplify our analysis we utilize a simple majority rule, where an agent changes to the majority state with probability 1.0. Note that as $\beta \to \infty$, $f(x)$ equals the simple majority rule. For $\beta = 10$, the value we use in the experiments below, there is a very slim probability of an agent not switching to the majority state. We believe this does not affect the analysis much.

Now, we can estimate the instantaneous change in $\rho_1(t)$ as the difference between the number of agents that are in state 0 and that change to state 1 (inflow) minus the number of agents in state 1 that change to 0 (outflow). The inflow will be the probability of picking as an active agent an agent in state 0: $1-\rho_1(t)$ times the probability that the majority state of the agents neighbors are 1.

The hypergeometric distribution can be used to calculate the probability of the majority state being 1 or 0. Consider each agent as a ball, and the state of the agent to be the color of the ball. In this case, state 1 indicates a red ball. There are $|N_i| = N$ balls of which, by the mean-field assumption above, there are $\rho_1(t)|N_i|$ red balls. From these agents a sample of size $\max\{1, \lfloor\theta * |N_i|\rfloor\} = m$ is taken. We want to find the probability that the number of agents with state 1 in the sample set exceeds $\lfloor \frac{m}{2} \rfloor$, which is equal to: $Q_H(N, N\rho_1(t), m, \lfloor \frac{m}{2} \rfloor)$.

The equation for the instantaneous change in $\rho_1(t)$ then becomes:

$$\begin{aligned}\frac{\partial}{\partial t}\rho_1(t) =& (1-\rho_1(t))Q_H(N, N\rho_1(t), m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1(t)(1 - Q_H(N, N\rho_1(t), m, \left\lfloor \frac{m}{2} \right\rfloor))\\ =& Q_H(N, N\rho_1(t), m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1(t)\end{aligned}$$

where the first term on the right hand side is the inflow, and the second term is the outflow.

Reorganizing Equation 4.7 and setting $\delta t \to 0$ we see that:

$$\frac{\partial}{\partial t}\rho_1(t) = Q_H(N, N\rho_1(t), m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1(t) \tag{4.8}$$

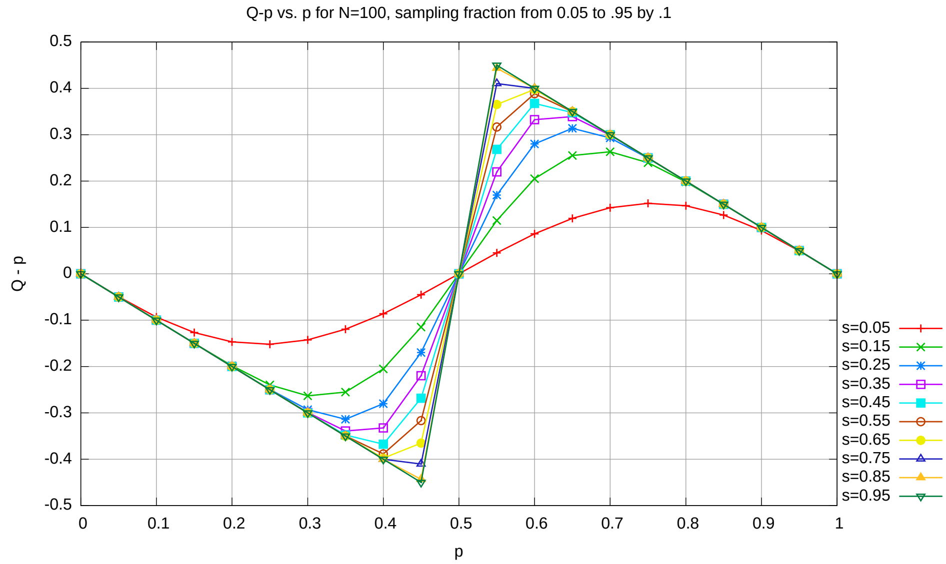
We want to study the stable fixed points of Equation 4.8, that is where $\frac{\partial}{\partial t}\rho_1(t) = 0$ since that will give us information on when the system will stop. In Figure 4.10 we numerically evaluate Equation (4.8) for different values of m.

Based on the graphs of instantaneous change we can see that there are two stable fixed points, at $\rho_1 = 0.0$ and 1.0, and one unstable fixed point which varies slightly but is around $\rho_1 = 0.5$. Based on this, we argue that the SMV process will result in agreement.

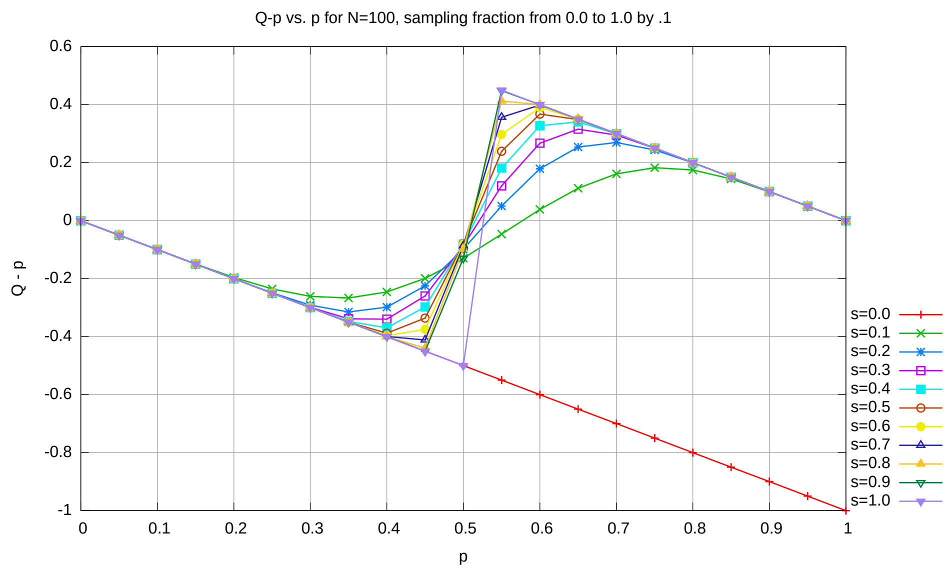
In the next sections we show why the unstable fixed points vary between the odd and even cases and also why in the even case the function is not symmetric.

Difference in the unstable fixed point.

Why is there a difference in the location of the unstable fixed point based on whether the size of the sample set is even or odd? The reason is that in the even case with $\rho_1(=)0.5$ there is the possibility of the sample set being split between the two states – in which case the active agent does not change. Thus, the probability of a majority of state 0 or 1 is skewed. In the odd case an even split cannot occur, so the probability of having a majority of state 0 or state 1 is exactly the same. Figures 4.11 4.12 illustrate this difference between the even and odd case. Theorems 12 prove that the probabilities are equal in the odd case and skewed in the



(a) Odd



(b) Even

Figure 4.10: ρ_1 vs. Equation 4.8 for odd and even sample sizes (represented as sampling fractions) with $N = 100$ Note that the unstable fixed point is not at .5 for the even case

even case. The proofs of these theorems are in Section 4.13.

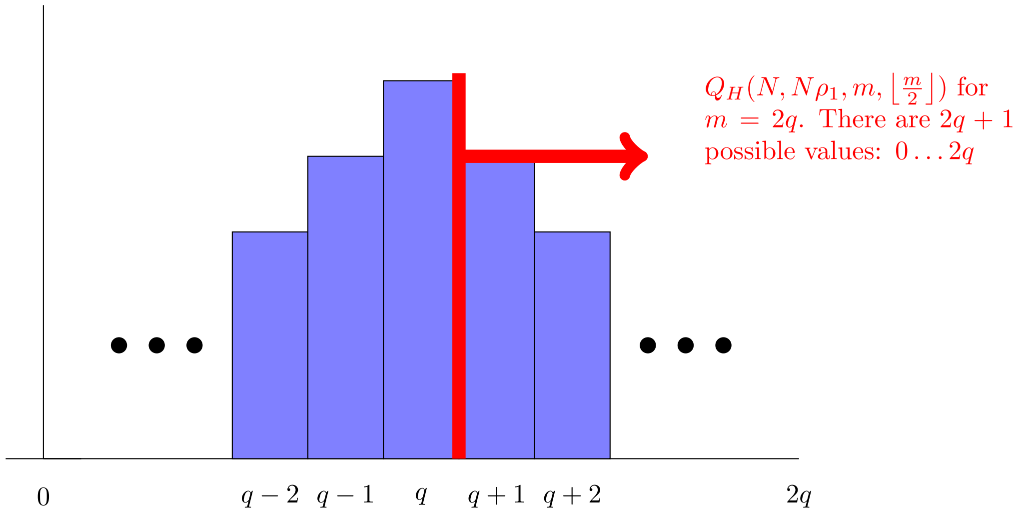


Figure 4.11: Diagram of the probability distribution for even m. The red arrow indicates the probabilities that are summed to calculate $Q_H(N, N\rho_1, m, \lfloor \frac{m}{2} \rfloor)$. The key difference between odd and even m is that in the even case $x = m/2$ is not included. See Figure 4.12 for the odd case.

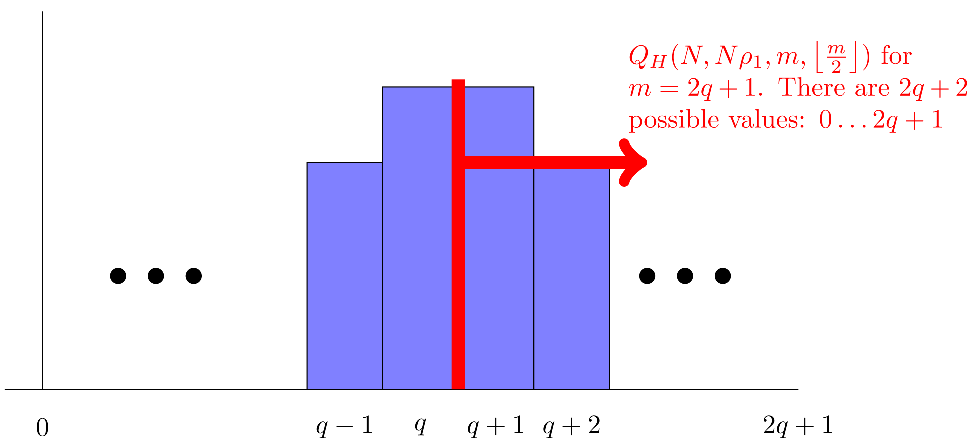


Figure 4.12: Diagram of the probability distribution for odd m. The red arrow indicates the probabilities that are summed to calculate $Q_H(N, N\rho_1, m, \lfloor \frac{m}{2} \rfloor)$. See Figure 4.11 for the even case.

Theorem 1. *Let N be any even number and let $\rho_1 = .5$. Then:*

$$h(N, 0.5N, m, x) = h(N, 0.5N, m, m - x) \tag{4.9}$$

for all $N, m, x \in [0, \ldots, m]$

Theorem 2. *Let N be any even number, $\rho_1 = .5$ and let $m = 2q + 1$, for some positive integer q, be an odd* *number. Then :*

$$Q_H(N, 0.5N, m, \lfloor m/2 \rfloor) = 0.5 \tag{4.10}$$

So we can see why in the odd cases (where $m = 2q+1$), when $\rho_1 = 0.5$, $Q_H(N, 0.5N, m, \lfloor m/2 \rfloor) - \rho_1 = 0.0$. In the even case this does not occur.

Theorem 3. *Let N be any even number, $\rho_1 = .5$ and let $m = 2q$, an even number. Then :*

$$Q_H(N, 0.5N, m, m/2) = \frac{1}{2}(1 - h(N, 0.5N, m, m/2)) \tag{4.11}$$

Symmetry

Another interesting quality of the instantaneous rate of change is its symmetry. In some cases there is symmetry around $\rho_1 = 0.5$, in other cases not. Let $f(x) = Q_H(N, Nx, m, \lfloor \frac{m}{2} \rfloor) - x$ for $x \in [0, 1]$. Then we can see that Figure 4.10(a) is symmetric, in the sense that: $f(x) = -f(1 - x)$. Why is this true of Figure 4.10(a) but not of Figure 4.10(b)? Theorem 4 proves that for even N and odd m there will be symmetry. Theorem 5 proves that for even N and even m there will *not* be symmetry.

Theorem 4. *For even N and odd m it is true that:*

$$Q_H(N, N\rho_1, m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1 = 1 - Q_H(N, N(1-\rho_1), m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1$$

Theorem 5. *For even N and even m it is true that:*

$$Q_H(N, N\rho_1, m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1 = 1 - Q_H(N, N(1-\rho_1), m, \left\lfloor \frac{m}{2} \right\rfloor) - h(N, N(1-\rho_1), m, \frac{m}{2}) - \rho_1$$

4.4.4 Related work

The Sampled Majority Vote protocol we have described in this section is similar to several other models of agreement problems. In Chapter 2 we surveyed some of these models. In this section we place the Sampled Majority Vote protocol in context with results from a variety of these other models.

Several of the models are meant to capture the dynamics of epidemics, and thus use epidemic terminology which we will follow as well. We will use the terms "infected" (I) which means an agent has a particular disease/infection, and "susceptible" (S) which means that the agent has the possibility of being infected, although the agent is healthy right now. A "recovered" (R) agent is one that has been infected and cannot be infected again. An epidemic is a situation where an infection spreads throughout a population.

We can view an epidemic as a type of agreement, where the state space is $\{I, S, R\}$. There has been much work on creating abstract models of epidemics.

[Moore and Newman, 2000, Newman and Watts, 1999] study epidemics through percolation theory. The basic idea is that there is a graph where the vertices and edges can be occupied or not occupied. If a vertex is occupied the agent is susceptible to being infected by a neighboring, infected vertex. If an edge is occupied, it means an infection can spread between the vertices. Suppose there is one infected individual in the population, and at each time step the infection spreads to the individuals neighbors, depending upon whether the edges and vertices are occupied. The main question Newman et. al. want to answer is, "For what fraction of occupied vertices/bonds will a giant occupied component occur?" They call this the *percolation threshold* and develop analytical solutions to find this for a variety of graphs. The size of this component indicates the reach of this infection into the population.

[Pastor-Satorras and Vespignani, 2001b, Pastor-Satorras and Vespignani, 2001a] also study epidemics but in a significantly different manner. Imagine now that an agent becomes infected with some probability v if their neighbor is infected, and that an infected individual recovers with probability δ. In this model there is no concept of occupied/unoccupied. Let $\lambda = v/\delta$ be the *effective spreading rate*. The question Pastor-Satorras et. al want to answer is, "For what value of λ will the number of infected individuals be comparable to the size of the entire population?" This value is called the *epidemic threshold*. [Pastor-Satorras and Vespignani, 2001b, Pastor-Satorras and Vespignani, 2001a] show severeal interesting things, such as the fact that there is no epidemic threshold for scale free graphs – infections with very small effective spreading rates can cause an epidemic.

A similar model is considered in the innovation diffusion literature. Suppose there are two behaviors, A and B. All agents start with behavior A, except for a few who are using behavior B. An agent changes from behavior A to behavior B depending upon how many of its neighbors are utilizing behavior B. Let q be the fraction of neighbors that must be in state B in order for an agent to change to B. Note that once an agent uses behavior B it cannot change back to using behavior A (this is called the "progressive assumption" [Kleinbergn, 2007]). Work in this area studies two different questions: (1) "What is the maximum value of q that allows for an innovation to spread through a population?" and (2) "If only k nodes can initially start with behavior B, which k nodes should we choose to maximize the spread of behavior B?". The answer to question (1) is called the *contagion threshold* and it was shown in [Morris, 2000] to be 0.5. Question (2) is difficult to answer (technically it is NP-Hard), but approximate solutions are known for certain simpler situations [Kleinbergn, 2007].

These three modeld differ in significant respects with the Sampled Majority Vote process. First, in the

Sampled Majority Vote model the probability of an agent changing state is proportional to the number of neighbors in the opposite state. In both studies of epidemics the probability of state change was fixed. Secondly, in the work on epidemic thresholds agents can individually change state, from I to R, even when their neighbors are not recovered; this type of state change is not allowed in the Sampled Majority Vote model.

Thirdly, the state space in all models are restricted. In the first model agents cannot recover from infection. In the third model agents cannot switch back from behavior B. The Sampled Majority Vote model does not constrain which states an agent can be in.

Apart from these differences in the underlying problem setting, there are differences in the objective of the work. With the Sampled Majority Vote protocol we are interested, primarily, in understanding how time to agreement is affected by the lack of accurate information; the models described above are primarily interested in whether agreement is reached.

Gossip protocols, studied in distributed systems, are protocols that use peer-to-peer, distributed message passing to propagate information between a large set of agents. A lot of work has been done on studying gossip protocols, [Kermarrec et al., 2003, Kempe et al., 2006, Patel et al., 2006, Demers et al., 1987]. In these cases agents can be classified in two states, having the information or not having the information. The progressiveness assumption holds in this scenario – once an agent knows a piece of information it does not forget that piece of information. The Sampled Majority Vote protocol does not hold to this assumption as agents can switch between the two states.

Cellular automata (CA) and graph-dynamical systems (an extension of cellular automata to general graphs) [Laubenbacher et al., 2008] are very similar to the Sampled Majority Vote model. Agents can be in a variety of states and change state based on the states of their neighbors. However, most of the results in CA and graph-dynamical systems assume some sort of synchronous updating where all agents update per time step. In contrast, in the Sampled Majority Vote model only one agent is active per timestep. The latter method of updating is often called "Random asynchronous updating" (RAS) [Cornforth et al., 2005].

As [Cornforth et al., 2005] describes, RAS updating has been shown to generate "edge-of-choas" patterns in 1-D CAs that are not present when updating synchronously. In addition, in synchronous updating cycles can be present, where a set of nodes might flip between two states forever [Kleinbergn, 2007]. There needs to be more work in studying the impact of asynchronous updating, but it is fair to say that results from synchronous updating settings do not necessarily hold in the asynchronous case.

To summarize, the Sampled Majority Vote model differs from the above models in these three characteristics:

1. *Non-progressive:* Agents can switch from state 0 to state 1 or from 1 to 0.

2. *Random Asynchronous updating:* A single agent is activated per timestep.

3. *Probabilistic update:* Agents change state based on the fraction of neighbors in a particular state.

4.5 Question 1: How much effort does an IGI expend?

Effort correlates to some type of resource expenditure which is based on the domain in which agreement takes place. We have listed several examples of effort when describing IGIs in section 4.3.2.

4.6 Question 2: IGIs and their impact on accuracy

In the previous section and in the chapter up till now we have established that IGIs require effort. Our concern now is with the accuracy of the information and how that is affected by the number of IGIs. In the SMV process the active agent calculates aggregate information about the distribution of states of its neighbors. Because of the limitation in the number of IGIs the active agent must sample its neighbors. The sampled distribution of states might not correlate with the actual distribution of states. There are two errors that can occur, the Mistaken Majority (MM) error and the Difference in Strength error. We define both below.

Suppose that α_i is the current active agent. Without loss of generality let us assume that α_i is in state 1. Let $|N_i| = k_i$ be the number of neighbors of agent α_i. Let k_1 be the number of neighbors of α_i that are in state 1 and let k_0 be the number of neighbors in state 0. Let $f_1 = \frac{k_1}{k_0+k_1}$ be the fraction of α_is neighbors in state 1. Let π_1 be the number of agents in Π that are in state 1 and then $\widehat{f_1} = \frac{\pi_1}{|\Pi|}$. The state that the majority of agents are in is called the *majority state*; the opposite is called the *minority state*.

We can now define more precisely the two types of errors:

Mistaken Majority The majority opinion of the agents in Π differs from the actual majority opinion of the neighbors. Without loss of generality assume that state 0 is the majority state, then:

1. $f_0 \geq 0.5$ but $\widehat{f_0} \leq 0.5$; or

2. $f_0 \leq 0.5$ but $\widehat{f_0} \geq 0.5$.

Difference in Strength The majority is preserved, but $\widehat{f_1}$ differs significantly from f_1

Even if an agent does correctly detect the majority state it still may misjudge the *strength* of that majority. Referring to the decision rule in Figure 4.8, strength misjudgments can significantly alter the probability of state change (by positioning an agent on the correct side of 50% but incorrectly far along the x-axis). The shape of the curve in Figure 4.8 is determined by β; the higher β is the smaller the effect of this positioning error will be. We leave a complete study of the impact of difference-in-strength errors for future work, and focus here on the more salient mistaken majority errors.

We say the sample is a *Success* when a mistaken majority error does not occur, otherwise it is a *Failure*. Clearly mistaken majority errors can cause non-convergence. Suppose a mistaken majority error *always* occurs – then agents switch to the minority state with high probability. This increases the fraction of agents with the minority state, eventually turning that state into the majority one and reversing the process. The result will be oscillation around $\rho_1 = 0.5$.

We can use the hypergeometric distribution to calculate the probability of a sample being successful. Let $P(\text{Success})$ be the probability that a sample is successful. In Section 4.4.3 we defined the probability of a majority state occurring. That is exactly the probability that we are concerned with here.

What is the probability of a sample being successful, $P(\text{Success})$? Without loss of generality suppose state 0 is the majority state. Then $P(\text{Success}) = P(\pi_0 > \pi_1)$ where $m = \pi_0 + \pi_1$ is the size of Π . This can be calculated easily by enumerating the number of different ways of choosing a π_0 size subset of k_0 times the possible ways of choosing an π_1 size subset of k_1. This leads to:

$$P(\text{Success}) = P(\pi_0 > \pi_1) = \frac{\sum_{\pi_0 > \pi_1} \binom{k_0}{\pi_0} \times \binom{k_1}{\pi_1}}{\binom{k}{m}} = Q_H(k_0 + k_1, k_1, m, \left\lfloor \frac{m}{2} \right\rfloor) \tag{4.12}$$

Figure 4.13 shows the probability of success for $k = 999$ for $\theta = [0.1, 1.0]$ and $f_0 = (.5, 1.0]$. As θ increases $P(\text{Success})$ increases for values of f_0 close to 0.5. In numerical simulations agents' states are randomly initialized, and stochastically one state will have a slight majority. Figure 4.13 shows that even under those conditions and with low θ the probability of committing a mistaken majority error is slim. As the fraction of majority state agents increases, this probability reduces significantly.

As the number of neighbors decreases $P(\text{Success})$ decreases as well. Figure 4.14 displays $P(\text{Success})$ for $k = 16$. As can be seen $P(\text{Success})$ does not increase vs. f_0 as it did when $k = 999$.

Figure 4.14 shows an interesting pattern where every other band has a higher probability. This is due to the fact that we require $\pi_0 > \pi_1$ and $\pi_0 + \pi_1 = m$.The intuition is that when we have an odd number we get a sample "for free" that does not change the minimum value of π_0 as compared to the even number one before. For example, for $m = 6$, $\pi_0 = 4$; however for $m = 7$, it is still true that the minimum value for π_0 is 4. The addition of the extra sample only increased the number of ways to choose a majority sample. Thus, the probability of success for an odd m is greater than the probability of success for $m - 1$.

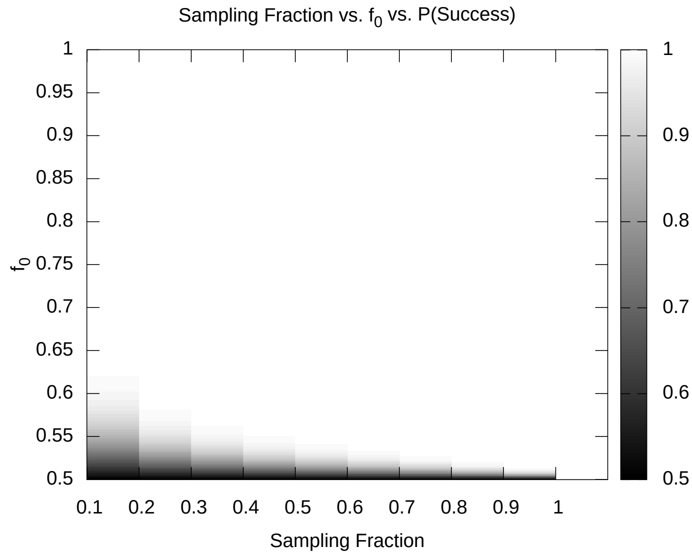


Figure 4.13: m (shown as the sampling fraction) vs. f_0 vs. P(Success)(color) for $k = 999$.

4.6.1 Discussion

Limitations on IGIs has a direct impact on the accuracy of the information an agent gets. We can see that the probability of a mistaken majority error increases significantly as the sampling fraction decreases. The next issue is whether the inaccurate information has an impact on agreement time.

4.7 Question 3: Information and agreement time

We performed extensive numerical simulations in order to explore the impact of inaccurate information on agreement time. We studied the SMV process on two types of interaction networks, complete and scale-free networks.

The complete network is the simplest interaction relation and represents a situation where everyone can talk to everyone else. Agreement occurs very quickly in the complete case. Scale-free networks are described in Section 3.6.1 and are a very important network that occur often in "real-world" data sets.

Figures 4.16 and 4.15 display sampling fraction versus time till agreement (TTA) for the complete graph case with $N = 1000, 5000, 10000$. We consider agreement to have taken place when 90% of the population is in the same state – this is a measure that is used in [Delgado, 2002] as well.

For each value of θ we performed 10 runs and averaged the results. Each run is comprised of a finite

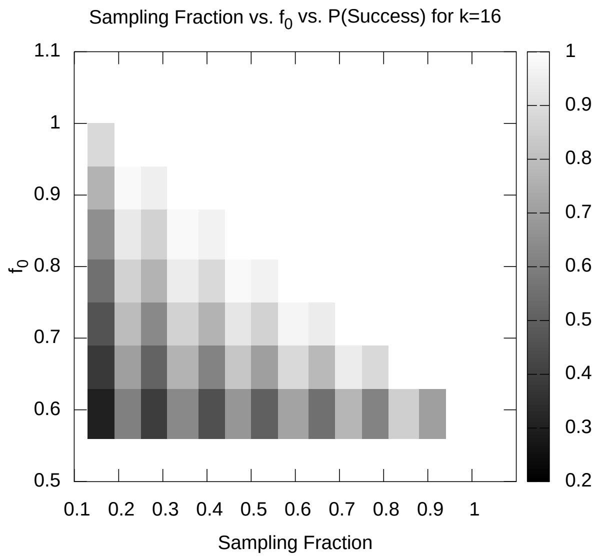


Figure 4.14: m (shown as the sampling fraction) vs. f_0 vs. $P(\text{Success})$(color) for $k = 16$.

number of time steps. In the complete graph case there were $50,000$ time steps and in the scale free graph case there were $1,000,000$ time steps. Each time step (or iteration) corresponds to one execution of the SMV process described in Section 4.4.2. At the start of each run a new graph was generated and all agents in the population were uniformly randomly initialized to 0 or 1.

For each run we calculated the first time step at which 90% of the population was in the same state – this time step is considered the time till agreement for this run. We averaged over 10 runs to get the results shown in the figures. The error bars indicate one standard deviation.

We used the Extended Barabasi-Albert scale free generation algorithm (described in Section A.1) to generate the scale free network. The parameters used were $m_0 = 4, m = 2, p = q = 0.4$. The coefficient of the degree distribution in this case will be approximately 2.5 (see [Delgado, 2002]). This exponent is close to the exponent of several real world networks, such as the World-Wide Web and the telephone call graph [Strogatz, 2001].

In the figures we have excluded the results for $\theta = 0.0$ because it corresponds to the voter model situation, for which numerous results are known.

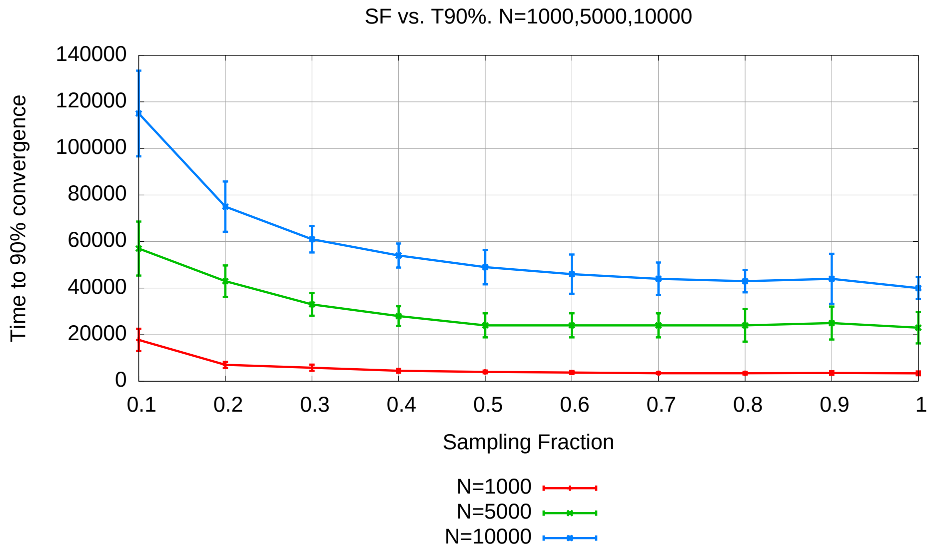


Figure 4.15: Log-Linear plot of Convergence time for Scale free graphs for a variety of N. Averaged over 10 runs, error bars indicate one standard deviation.

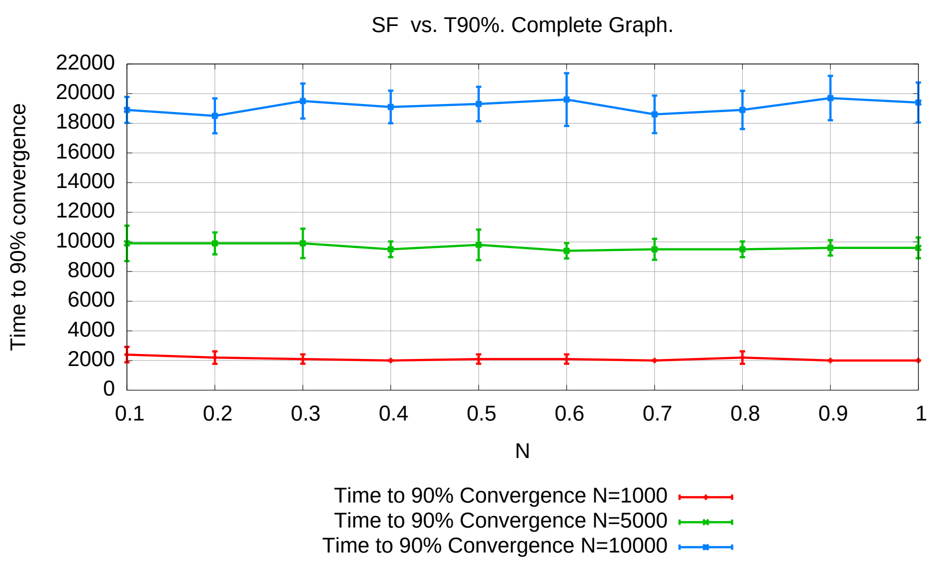


Figure 4.16: Plot of Convergence time for complete graphs for a variety of N. Averaged over 10 runs, error bars indicate one standard deviation.

4.7.1 Discussion

There are some surprising conclusions that can be drawn from these experiments. First, note that, in both the complete and scale-free case, as sampling fraction increases time till agreement decreases. Given that as the sampling fraction increases the probability of making a mistaken majority error significantly decreases, i.e. the information an agent gets is more accurate, we can conclude that these experiments provide support for the claim that information accuracy and agreement time are inversely related which is hypothesized in Figure 4.9.

The more important question is how accuracy and agreement time are related. We find some surprising results here. As can be seen, in the complete graph case the value of the sampling fraction seems to make very little difference in terms of time till agreement. For $N = 1000$ as long as $\theta \geq 0.1$ the system reaches agreement quickly. The same is true for $N = 5000$ and 10000. Even though every agent is receiving potentially incorrect information, agreement still occurs quickly. This is somewhat surprising, but seems correct based on the exponential decline in incidence of mistaken majority errors as the sampling fraction increases.

Figure 4.15 displays sampling fraction versus time to agreement for $N = 1000, 5000, 10000$. Immediately we can see a significant difference from the complete graph case. First, we can clearly see that time to agreement is significantly less in complete than scale-free networks, even with the same number of agents. For example, at $\theta = 0.1$ it takes approximately 2000 timesteps till agreement is reached in the complete graph case with $N = 1000$. However, in the scale free graph case it takes approximately $20,000$ iterations for the same number of agents. This holds true for all values of θ.

In the Scale-Free case we find that as we increase sampling fraction and increase the accuracy of the information, time to agreement decreases significantly. However, at some point more accurate information does not decrease agreement time significantly. This is similar to the complete graph case, except here there is a more gradual decrease in agreement time.

Intuitively, one would expect more accurate information to decrease agreement time in a linear fashion – the better information an agent has, the quicker the agent reaches agreement. However, this intuition is actually false, as there is a point at which more accurate information does not decrease agreement time. The experiments in this section have explored this point through extensive numerical simulations.

So far, we have shown how the fundamental agreement tradeoff operates in a specific binary state, complex static graph situation. Section 4.5 discussed how interactions require effort by surveying a range of IGIs from different domains. Section 4.6 showed how a reduction in the number of IGIs can lead to two types of errors, the mistaken majority error and the Difference in Strength error. Finally, in this section we showed how inaccurate information can affect agreement time.

In the next section we explore how to leverage the non-linear relationship between agreement time and accuracy of information in determining the optimal point in the tradeoff.

4.8 Question 4: Finding the optimal tradeoff

The first step to finding an optimal tradeoff is to provide a quantification of the tradeoff. We do this by creating the Information-Centric Convergence Cost Metric which captures the cost of effort and time to agreement. The ICCC metric can be used with any protocol.

Using the ICCC we can find the point, i.e. sampling fraction, at which the system is optimally trading off inaccurate information for agreement time.

4.8.1 The information centric convergence cost metric

Let T be the number of time steps till agreement in discrete units of time. We assume that the systems we are studying can be viewed as discrete-time systems. Then $T \in \mathbb{N}$.

Previously we have used the term "effort" to denote the expenditure of resources by an agent. We assume here that effort is a positive real number.

Let $E(t)$ be the total amount of effort expended by the system at time t. $E(t)$ is the sum of the effort for interaction and effort for information use:

$$E(t) = \text{Effort spent on interaction at timestep } t + \text{Effort spent on information use at timestep } t$$

.

Each active agent α_i at time t will execute at least $|I_{i,t}|$ interactions per timestep (one interaction per agent in the actual interaction set), thus the total number of interactions at time t is:

$$I_t = \sum_{\alpha_i \in C_t} |I_{i,t}|$$

.

Let $EI(t)$ be the effort spent on interactions at time step t, then:

$$EI(t) = I_t \times E_I$$

where E_I is the amount of effort expended per interaction.

Each active agent will use the information it gathered to possibly change state. Thus the effort spent at time t on information use, $EU(t)$ is:

$$EU(t) = |C_t| \times E_U$$

where E_U is the amount of effort expended for information use by a single agent.

In calculating $EU(t)$ and $EI(t)$ we have assumed a homogeneous agent population. This is not generally true, as in open, heterogeneous systems new agents may enter that are significantly different from others in terms of their capabilities. However, the scope of this work is limited and we leave to future work the study of heterogeneous agent situations.

Finally:

$$E(t) = EI(t) + EU(t) \tag{4.13}$$

$$= E_I \sum_{\alpha_i \in C_t} |I_{i,t}| + |C_t| \times E_U \tag{4.14}$$

.

Let T be the number of timesteps for agreement to be achieved in the system. Then the total effort expended by the system, E, to reach agreement is:

$$E = \sum_{0}^{T-1} E(t)$$

.

We are now ready to define the ICCC metric. The purpose of the ICCC is to capture the fundamental agreement tradeoff, that is the interaction between effort, information and time to agreement. We will not represent information directly, but rather focus on effort, which we know impacts the accuracy of information (see Section 4.6 for an exploration of this in the context of SMV). Then the ICCC will be a function of E and T. We define the ICCC, $C(T, E)$ as a linear combination of these two factors:

$$C(T, E) = c_t T + c_e E \tag{4.15}$$

where c_t and c_e are a measure of the cost per unit time and unit effort (respectively). We normalize c_t and c_e so that they sum to 1.0.

Intuitively, the relation between c_t and c_e indicates the relative cost of time and effort. If $c_t = 1.0$ it means that all the cost is on time to agreement – it doesn't matter how much effort a system expends, as long as the time to agreement is minimized.

On the other hand, when $c_e = 1.0$ then every unit of effort is expensive, so one would want to minimize the amount of effort expended at the cost of time to agreement.

By varying c_e and c_t we can vary the importance of time to agreement versus effort.

4.8.2 ICCC and SMV

We performed extensive numerical simulations in order to find the optimal tradeoff between effort and agreement time in the SMV system. As before, we studied the SMV process on two types of interaction networks, the complete and scale-free networks.

We assume in this work that communication cost dominates processor cost; thus we set $E_U = 0.0$. The effort values shown in the graph below are a count of the number of interactions that took place – where in every interaction one active agent gets information about the state of another agent. This is a justified assumption as in many cases (for instance energy constrained wireless sensor networks) communication cost dominates the cost of processing.

The experimental setup was exactly the same as in Section 4.7, in fact, the data for these graphs were collected from the same runs.

Figures 4.17 and 4.18 shows the system effort averaged over 10 runs for each setting of sampling fraction for the complete and scale-free graph cases.

In the complete graph case there is a significant increase in effort as sampling fraction increases – by nearly an order of magnitude from 1.5×10^5 to 1.5×10^6. In contrast, in the scale-free graph case the increase in effort is significant, but not an order of magnitude.

Figure 4.19 and 4.21 shows the ICCC for $c_t = 1 - c_e$ going from 0.0 to 1.0 for the complete and scale free (respectively) cases. As c_t increases, the ICCC value changes to reflect the fact that interactions cost more with respect to time to agreement.

Figure 4.19 displays the ICCC as a 3d surface. The x-axis is c_t, the y-axis is θ, and the z-axis is the ICCC value. The z-axis is in logarithmic scale, while the color represents the ICCC value in linear scale (legend to the right).

Let us look at the complete case first. Looking from right to left we can see an increase in the ICCC value (as the height increase). At the far right, $c_t = 1.0$ which means that the cost of an interaction is 0.0. Thus, the ICCC value is based solely on the time till agreement. As Figure 4.16 shows, there is very little change in time to agreement as θ varies – thus the nearly uniformly colored column at $c_t = 1.0$.

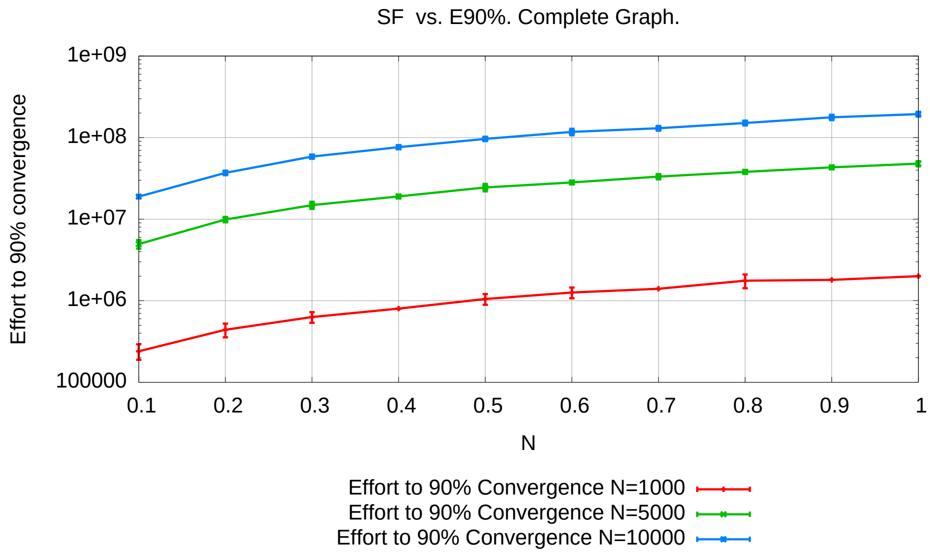


Figure 4.17: Log-Linear plot of effort to agreement for complete graphs for a variety of N. Averaged over 10 runs, error bars indicate one standard deviation.

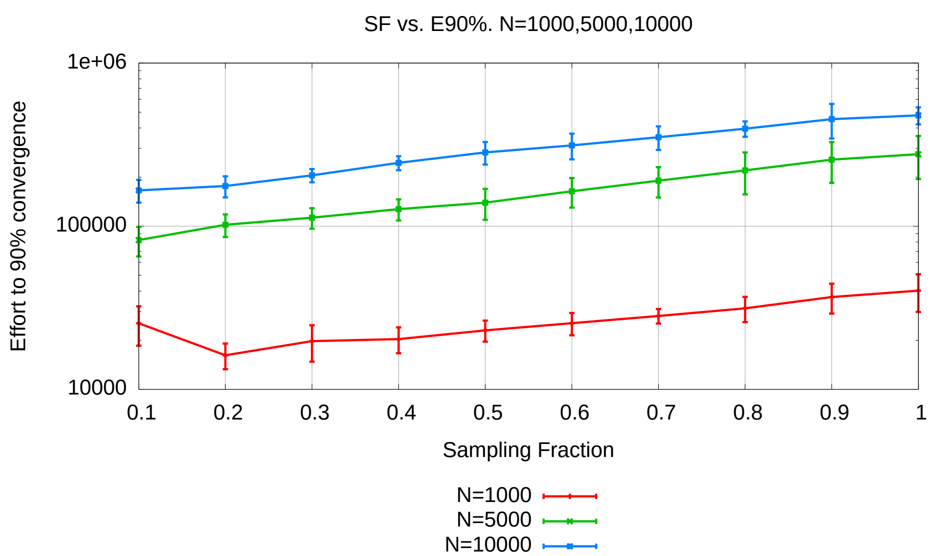


Figure 4.18: Log-Linear plot of effort to agreement for scale-free graphs for a variety of N. Averaged over 10 runs, error bars indicate one standard deviation.

As c_t is decreased we see more heterogeneity in terms of the ICCC value. consider $c_t = 0.5$ (not labelled). In this case, we consider the time to agreement and the number of interactions to have the same cost. Thus, when the sampling fraction is high, the ICCC is high as well. As we decrease c_t the cost of effort increases as well and thus the ICCC increases for more sampling fraction values.

Figure 4.20 shows the ICCC for $N = 10000$ with $c_t = 0.0, 0.5, 1.0$. We can see that as sampling fraction increases, the ICCC value increases, due to an increase in the number of interactions.

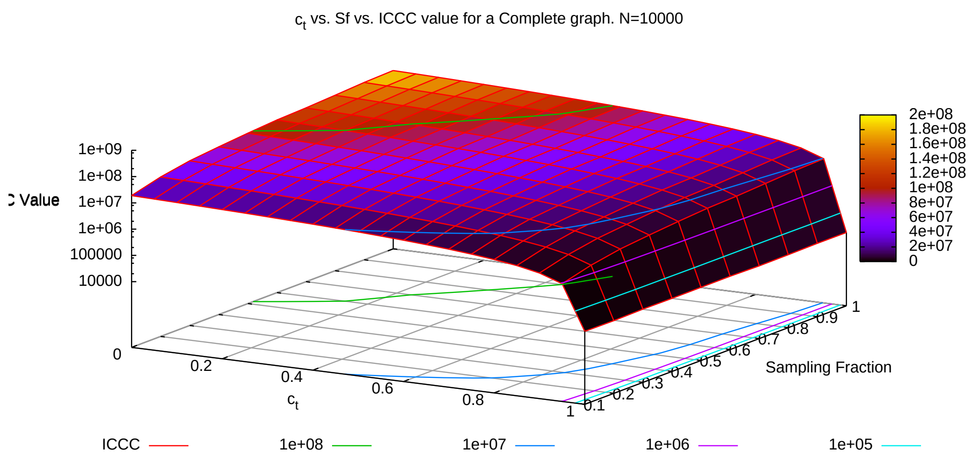


Figure 4.19: The x-axis represents c_t the y-axis is θ, and the color of a square represents the ICCC value of that particular combination of c_t and θ. As c_t decreases (from right to left) the ICCC value increases significantly as the cost of an interaction goes up.

Figure 4.21,4.22 show the ICCC value for the scale-free graph case. We see similar behavior as in the complete graph, but with some small differences. Starting from the right and moving left the ICCC increases significantly as effort is counted more than time.

Given a specific setting of c_t we can find the value for θ that minimizes the ICCC. Figure 4.23 shows the minimal sampling fraction over the range of c_t values. We can see that for the majority of values of c_t the optimal sampling fraction is 0.1 (note we only evaluated the system at $\theta = 0.0$ and 0.1 – there is a possibility that the system reaches agreement quickly at an even lower value of θ.

When $c_t = 1.0$ Figure 4.23 indicates that that optimal sampling fraction is 0.2. As Figure 4.20 shows there is a slight decrease in the ICCC value at $\theta = 0.2$ – which most likely was due to a statistical fluctuation. Thus the specific value of 0.2 is not as important as the concept of there existing some optimal sampling fraction which is significantly less than 1.0.

For the scale-free graph there is greater heterogeneity in terms of which value of the sampling fraction is optimal. Figure 4.24 shows the optimal sampling fraction for different values of c_t. We can see that as

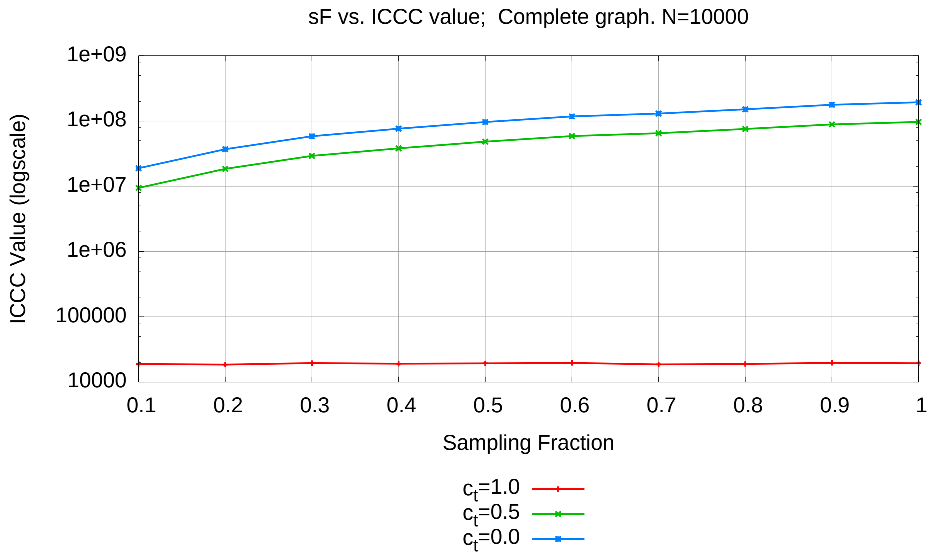


Figure 4.20: Log-Linear plot of sampling fraction versus ICCC value for a variety of c_t values. As θ increase the ICCC increases as more interactions are executed. Complete graph case.

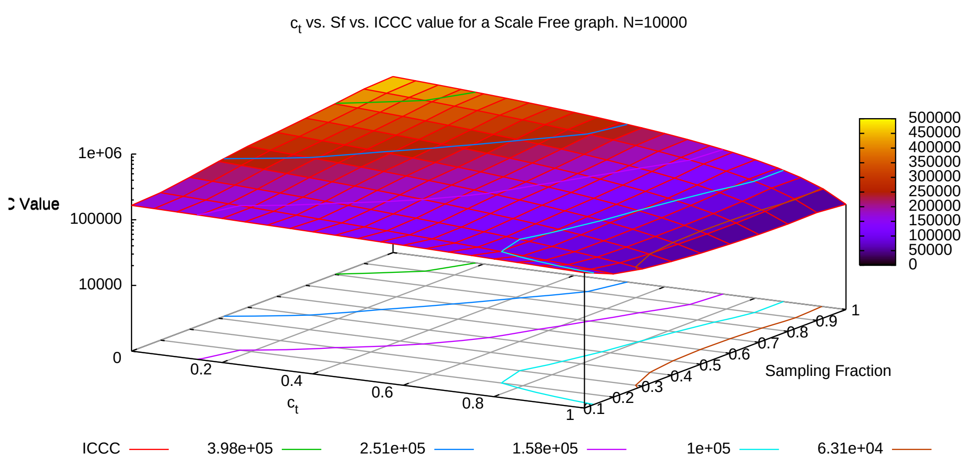


Figure 4.21: The x-axis represents c_t the y-axis is θ, and the color of a square represents the ICCC value of that particular combination of c_t and θ. As c_t decreases (from right to left) the ICCC value increases significantly as the cost of an interaction goes up.

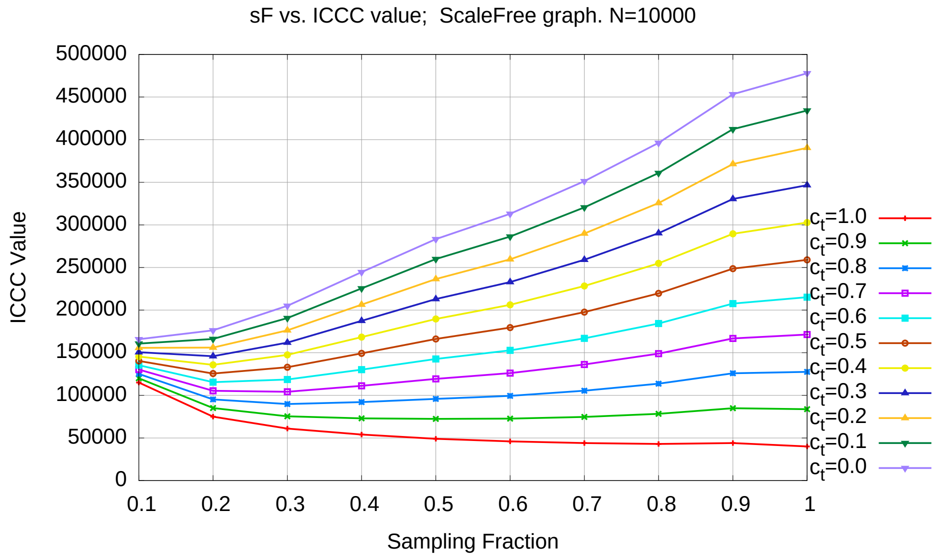


Figure 4.22: Plot of sampling fraction versus ICCC value for a variety of c_t values. As θ increases the ICCC increases as more interactions are executed.

c_t increases the optimal sampling fraction increases as well, but that for some intervals of c_t the optimal sampling fraction does not change.

One interesting point, for the scale free graph as we increase the number of agents the optimal sampling fraction changes. Figure 4.25 shows the optimal sampling fraction for all three values of N. We can see that as N decreases, the point at which a higher sampling fraction is optimal occurs for much higher values of c_t. We are investigating this phenomena.

4.9 Discussion

Through the ICCC metric we can determine the optimal sampling fraction in terms of trading off the cost of longer time to agreement and effort to agreement. We find, surprisingly, that the optimal sampling fraction can be quite low. For complete graphs, a sampling fraction of 0.1 seems to work quite well. For scale-free graphs the optimal sampling fraction varies significantly. When $c_t \leq c_e$, the optimal sampling fraction varies from 0.0 to 0.2. When $c_t > c_e$, the optimal sampling fraction varies from 0.2 to 1.0. Except at the extreme values (when $c_t = 0.0$ or 1.0, the optimal sampling fraction is quite low.

When studying innovation diffusion on social networks it is often assumed that an agent interacts with all of its neighbors concurrently [Gibson, 2005]. However, real interactions take place *non-concurrently* – one might interact with a subset of friends more often than others.

One option is to limit the links in a social network to only those pairs of individuals that communicate

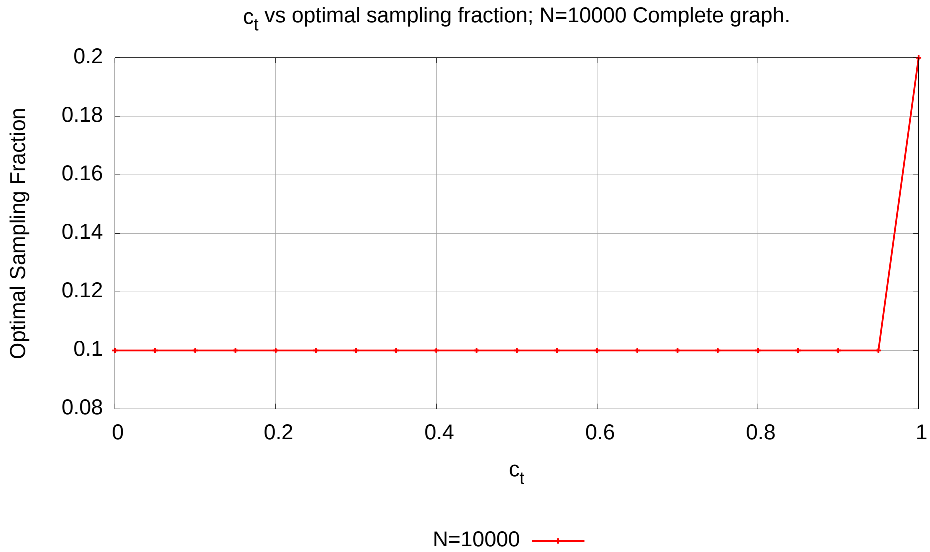


Figure 4.23: The x-axis is c_t the y-axis is the value of θ that minimizes ICCC.

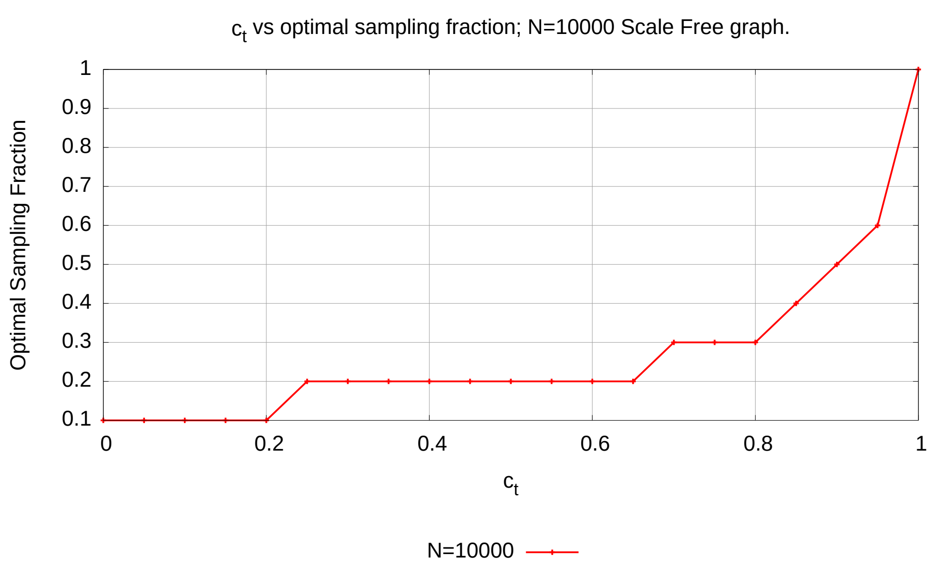


Figure 4.24: The x-axis is c_t the y-axis is the value of θ that minimizes ICCC.

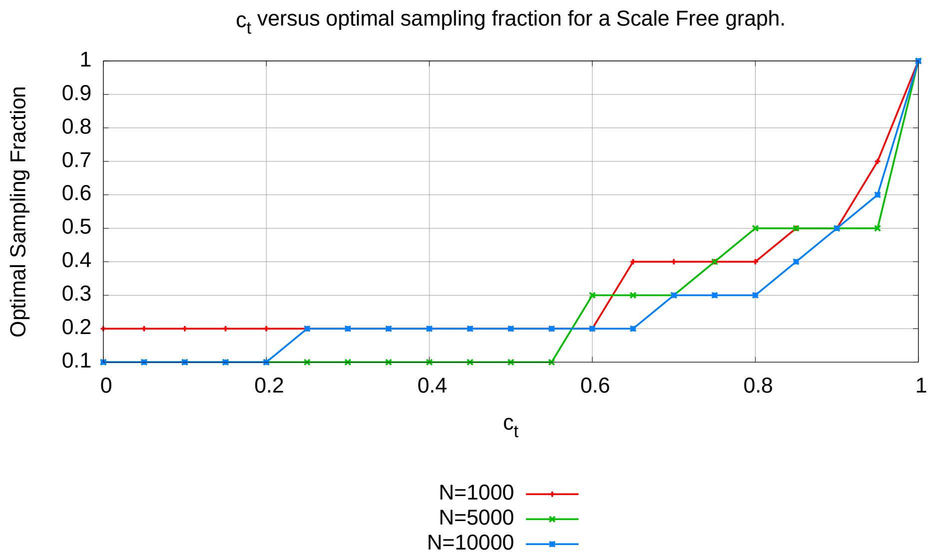


Figure 4.25: The x-axis is c_t the y-axis is the value of θ that minimizes ICCC. For all three values of N

above a high threshold. However, this does not capture the fact that individuals can be influenced by limited contact.

There has started to be some work on understanding innovation diffusion with non-concurrent interaction. [Gibson, 2005] looks at how time scheduling can influence innovation diffusion. [Habiba et al., 2008] studies disease spreading on dynamics networks – we can view the existence of an edge as communication. . [Kossinets et al., 2008] investigated real-world email-communication data from this perspective.

We can view the SMV model as a model of non-concurrent interactions between agents. Agents randomly choose a subset of other agents to interact with when they are active.

The simplicity of the SMV model does not capture the complexities of why there is non-concurrent interaction; however, we can view the SMV as an approximation that is more amenable to formal analysis.

4.10 An exploration of fractional versus fixed size sampling

In this section we explore the differences between two sampling regimes, fractional and fixed sampling. Fractional sampling is the default option in the Sampled Majority Vote process which is described above.

A fixed sampling regime is slightly different in that a fixed number of nodes are sampled rather than a fraction of nodes. The number of nodes that should be sampled is called the *sample size*. Our objective is to understand how fixed sampling affects agreement time.

Let θ be the sampling fraction used in fractional sampling and let m be a fixed number of samples that is used in fixed sampling. Let k refer to the degree of a node. Let $m(\theta, k)$ be the number of neighbors sampled for a node with degree k in the fractional sampling regime when using sampling fraction θ. That is:

$$m(\theta, k) = \max\{1, \lfloor \theta * |k| \rfloor\}$$

Let $mf(m, k)$ be the number of neighbors sampled for a node with degree k in the fixed sampling regime:

$$mf(m, k) = \begin{cases} m & \text{if } m \leq k \\ k & \text{if } m > k \\ 1 & \text{if } m == 0 \end{cases}$$

We assume that m and θ are global and constant – they apply to every node and they do not change in an experiment.

Let us start by examining fixed and fractional behavior on k-regular graphs. For every $m <= k$, there is a range of values of θ that would make $m(\theta, k) = mf(m, k)$. For instance, with $k = 4$ and $m = 2$ we can use $\theta = 0.5$. The reverse applies as well, for every θ there exists an m that will make $m(\theta, k) = mf(m, k)$. This means that there is no difference in behavior between fixed and fractional sampling regimes on k-regular graphs. Figure 4.26 shows time to 90% agreement for a fixed sampling regime; we can see that the results are very similar to the results for the fractional sampling case shown in Figure 4.16.

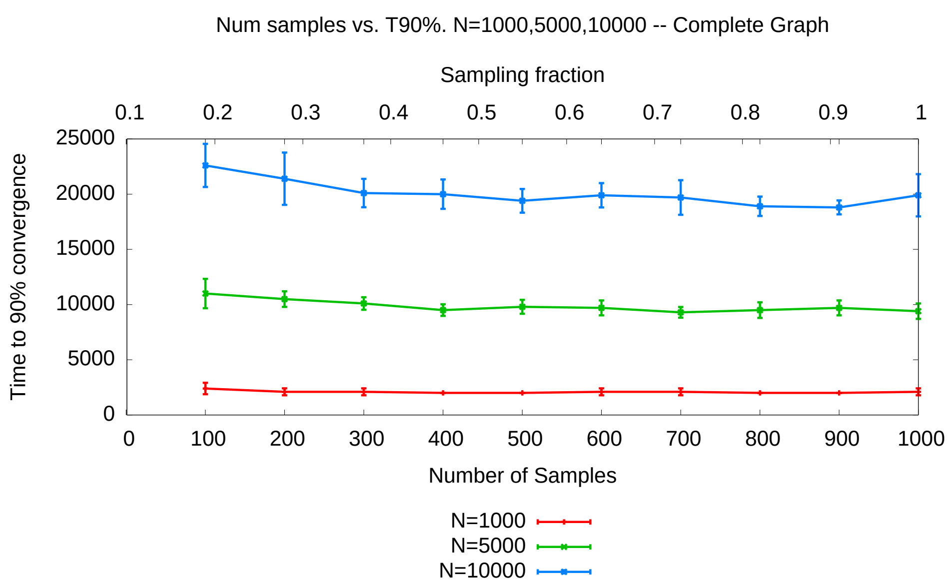


Figure 4.26: Time to 90% Agreement for a comple graph using the fixed sampling regime.

There are more interesting differences when considering degree-heterogeneous graphs. Our objective is to understand the role of limited, inaccurate information on time to agreement. As such, we want to compare time to agreement with varying sample sizes with time to agreement when an agent has maximum information. The maximum information case occurs when an agent can sample all of its neighbors. As we pointed out in Section 4.6, there are two types of errors that can occur when an agent samples its neighbors, the mistaken majority and difference in strength errors. In the following we only consider the more salient mistaken majority error and leave the difference in strength error for future work.

Our goal, then, is to understand the incidence of mistaken majority errors and how that is affected by fixed and fractional sampling. In Equation 4.12 we defined the probability of committing a mistaken majority error which is based on:

1. The degree of the active agent;
2. The distribution of states in the active agents neighborhood;
3. The number of neighbors sampled.

Figure 4.13 and Figure 4.14 indicate how the probability of success changes as the distribution of states and the number of neighbors sampled varies. We can clearly see that the number of neighbors sampled has a dramatic affect on the probability of success. Thus, in order to explore how time to agreement is impacted by fixed and fractional sampling we will need to understand how the number of samples in both cases is affected.

Clearly when all neighbors have the same state the probability of success is 1.0 no matter how many neighbors there are and how many samples are taken. Similarly, when there is only 1 neighbor the probability of success is 1.0 as well – at a minimum the active agent samples at least 1 neighbor.

Figure 4.10 illustrates the impact of fixed and fractional sampling on different degree nodes. The number of neighbors sampled does not depend upon the distribution of states over the neighbors, so we do not consider that variable here.

In the figure we have $\theta = 0.5$ and $m = 3$. The blue line indicates $mf(m, k)$ and the dashed line is $m(\theta, k)$. We can see that for low degree nodes fixed sampling provides more samples than fractional sampling. If we increased θ, the dashed line would gradually move up to become the $x = y$ line, since when $\theta = 1.0$ all neighbors are sampled. At $k = 6$ we see that the fractional and fixed sampling regimes are equivalent – both sample 3 neighbors. For $k > 7$, the fractional regime samples more than the fixed regime.

What does this mean? The basic idea is that fixed sampling is “better” in the sense of resulting in more samples, for lower degree nodes, whereas fractional sampling is “better” for higher degree nodes. This makes

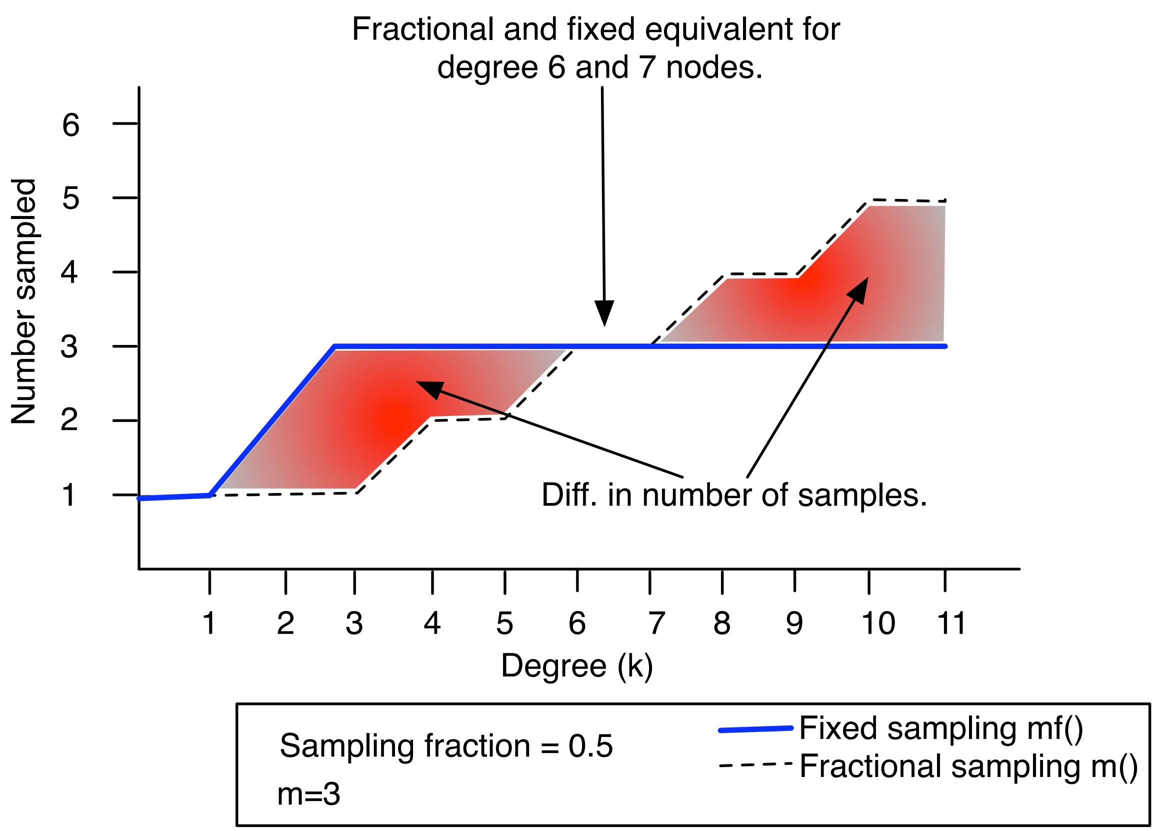


Figure 4.27: Illustration of the impact of fixed and fractional sampling.

sense, since for low degree nodes fractional sampling will usually result in choosing a single neighbor as a sample; whereas with fixed sampling one chooses all the neighbors.

Given this, we should understand the distribution of node degrees in a scale free graph. The Sampled Majority Vote process utilizes a random asynchronous update process [Cornforth et al., 2005], that is, one uniformly randomly chosen node is picked as the active agent per timestep. Knowing the distribution will indicate which nodes will be chosen as active agents. Figure 4.28 shows the degree count for a single scale-free graph generated using the algorithm and parameters described in the results section. Figure 4.29 shows the cumulative count, the number of nodes with degree $\leq x$. We can see that nearly 80% of the nodes have degree less than 16.

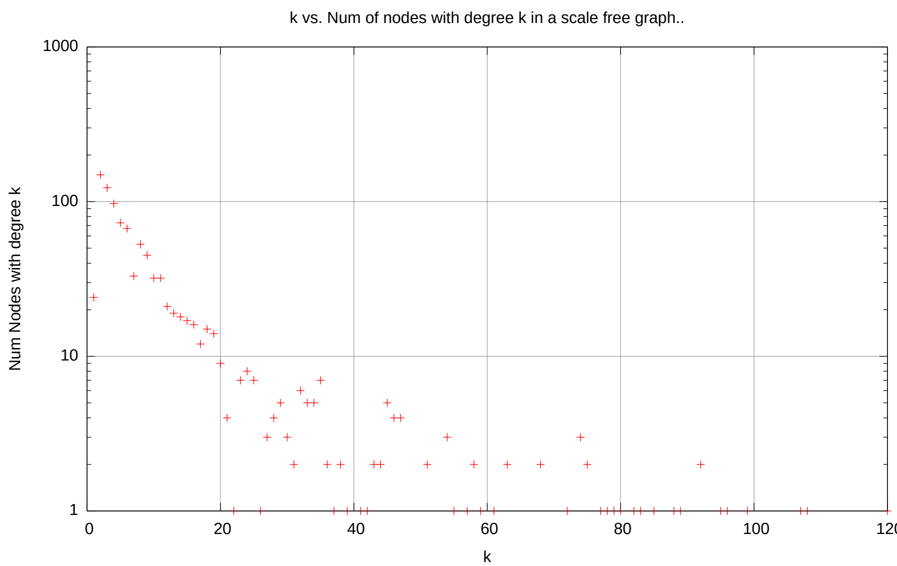


Figure 4.28: Degree count for a randomly generated scale-free graph. The x-axis is degree and the y axis indicates how many nodes have that degree.

Given this, we expect that fixed sampling should reach agreement quicker than fractional sampling, as the majority of nodes chosen will be small degree nodes. Figure 4.30 is a log-linear plot of time to 90% agreement under fixed and fractional sampling regimes for $N = 1000$. The sampling fraction is on the top x axis, and the number of samples is on the bottom x axis. We can see that the time to agreement in both regimes quickly reaches a similar value for $m = 25$ and $\theta = 0.4$.

However, a fixed sampling regime provides for very fast agreement time for very small values of m, because most of the nodes chosen are low degree nodes.

The reason fixed works better is that more samples are being taken, which means more communicative effort. Figure 4.10 shows the effort required to reach agreement. We can see that the fixed sampling regime case executes more communicative effort than the fractional case.

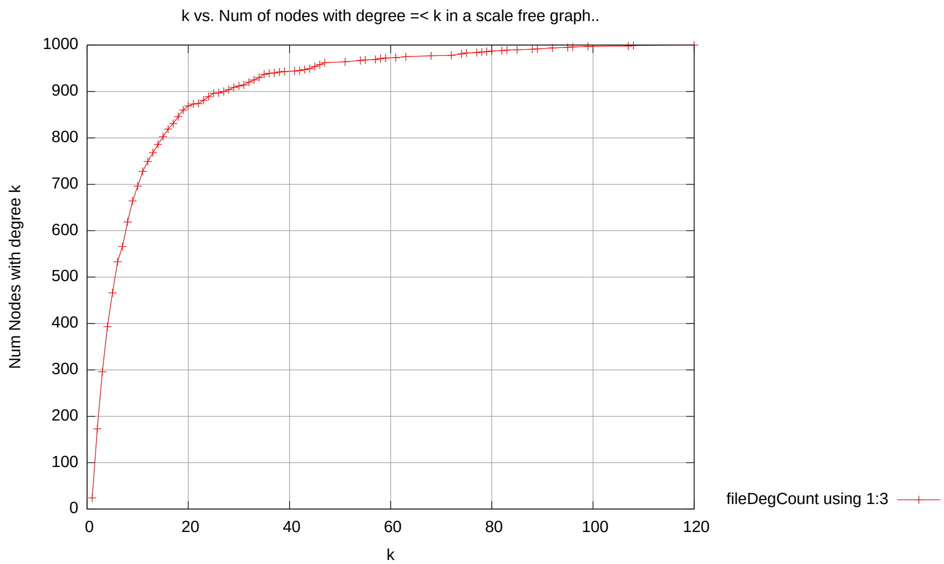


Figure 4.29: Cumulative degree count for a randomly generated scale-free graph. The x axis is degree and the y axis indicates how many nodes have degree less than or equal to the x value.

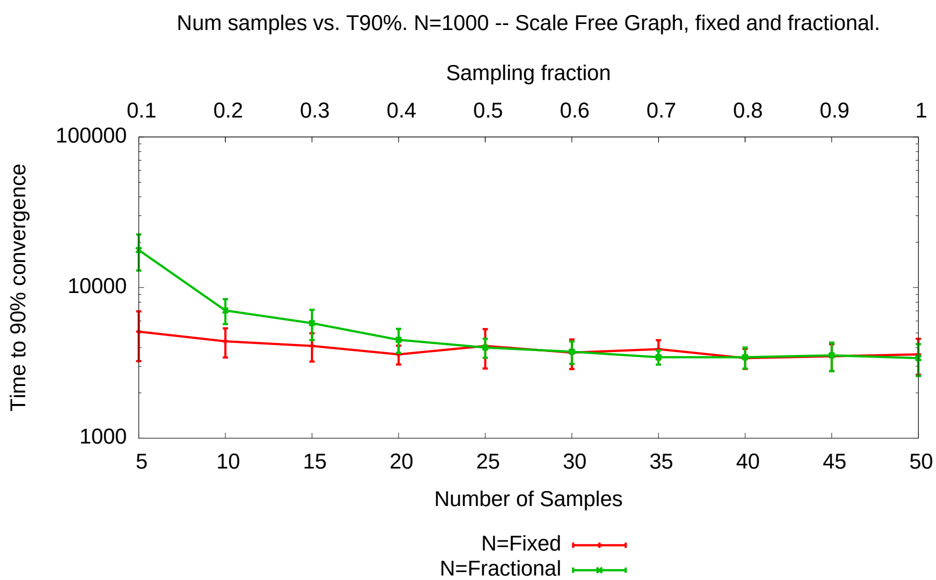


Figure 4.30: Time to 90% agreement for fixed and fractional sampling regimes.

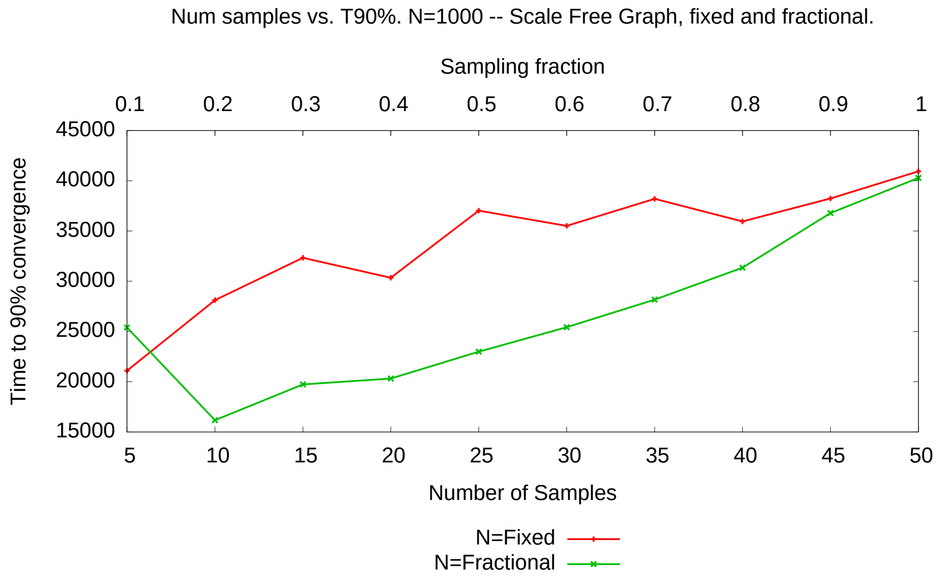


Figure 4.31: Effort to 90% agreement for fixed and fractional sampling regimes.

4.10.1 Summary

Fixed and fractional sampling regimes are both instances of sampling and both can be used to explore the role of information. We have shown in this section that a fixed sampling regime provides more samples when the active agent has few neighbors, however when an agent has many neighbors the fractional approach results in more samples.

In a scale-free graph the majority of nodes are low degree, thus a fixed regime works well. It would be interesting to understand how the fixed vs. fractional issue impacts agreement in other graphs, such as small world graphs, where the range of degrees of nodes is much smaller.

While fixed sampling does perform slightly better, this performance comes at the cost of increased sampling, i.e. effort. This reinforces the fundamental agreement tradeoff – better performance requires more effort.

4.11 Visualization of agreement dynamics

Visualization can provide insight into the dynamics of agreement. We used Cytoscape [Shannon et al., 2003] a program for visualizing large graphs to create visualizations of the Sampled Majority Vote process at several timepoints. Since scale-free networks can be difficult to visualize due to the large number of edges, we visualized a single run in which there were only 300 agents.

Fraction of agents in the majority state at the time steps vizualized above:

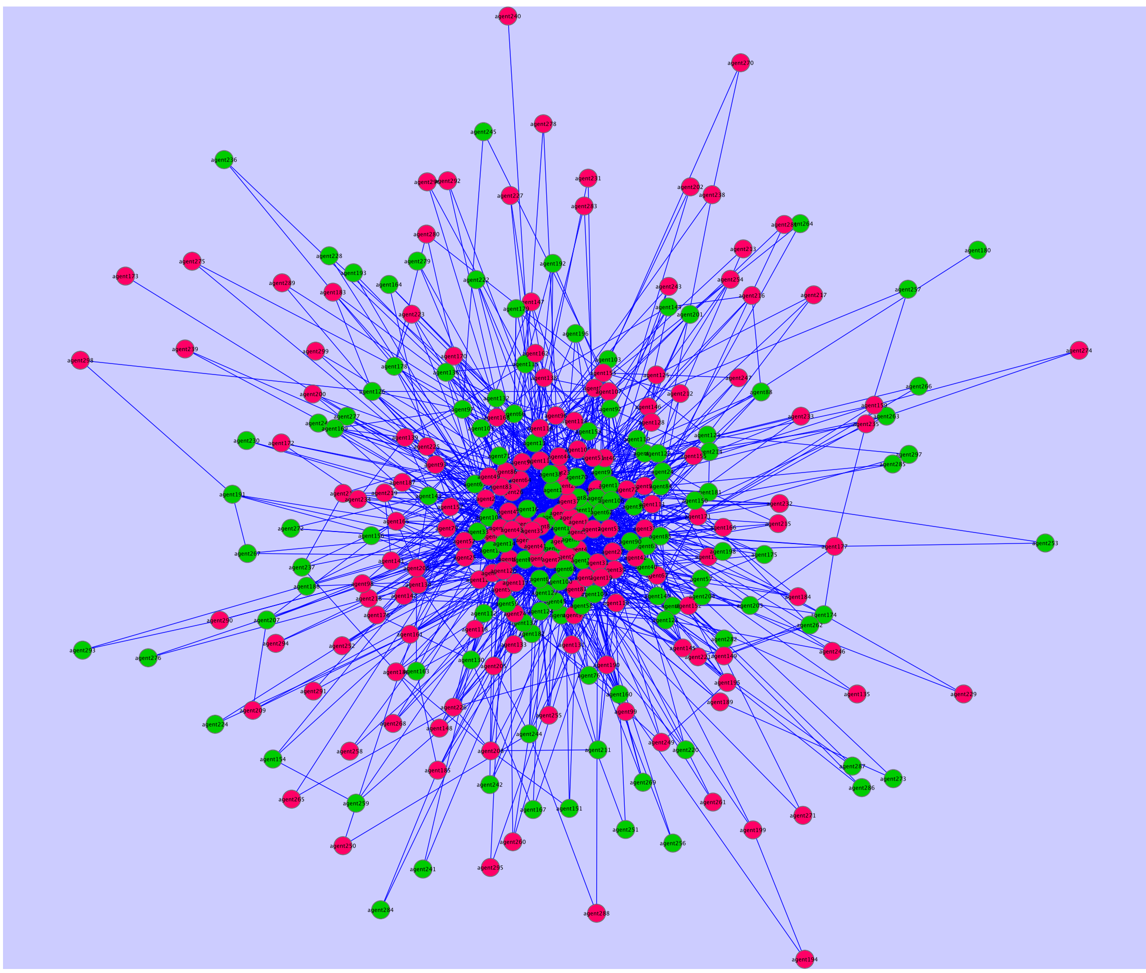


Figure 4.32: The system at $t = 0$.

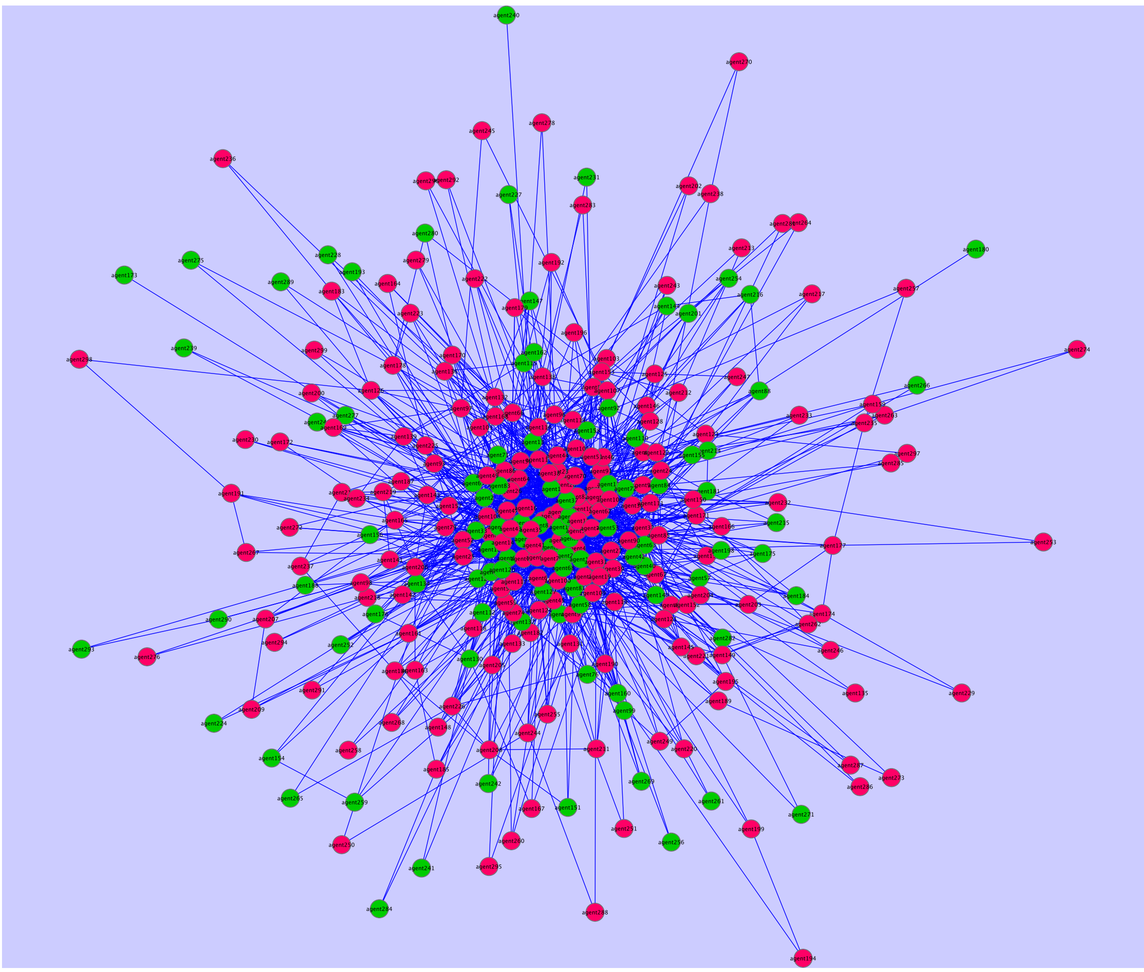


Figure 4.33: The system at $t = 500$.

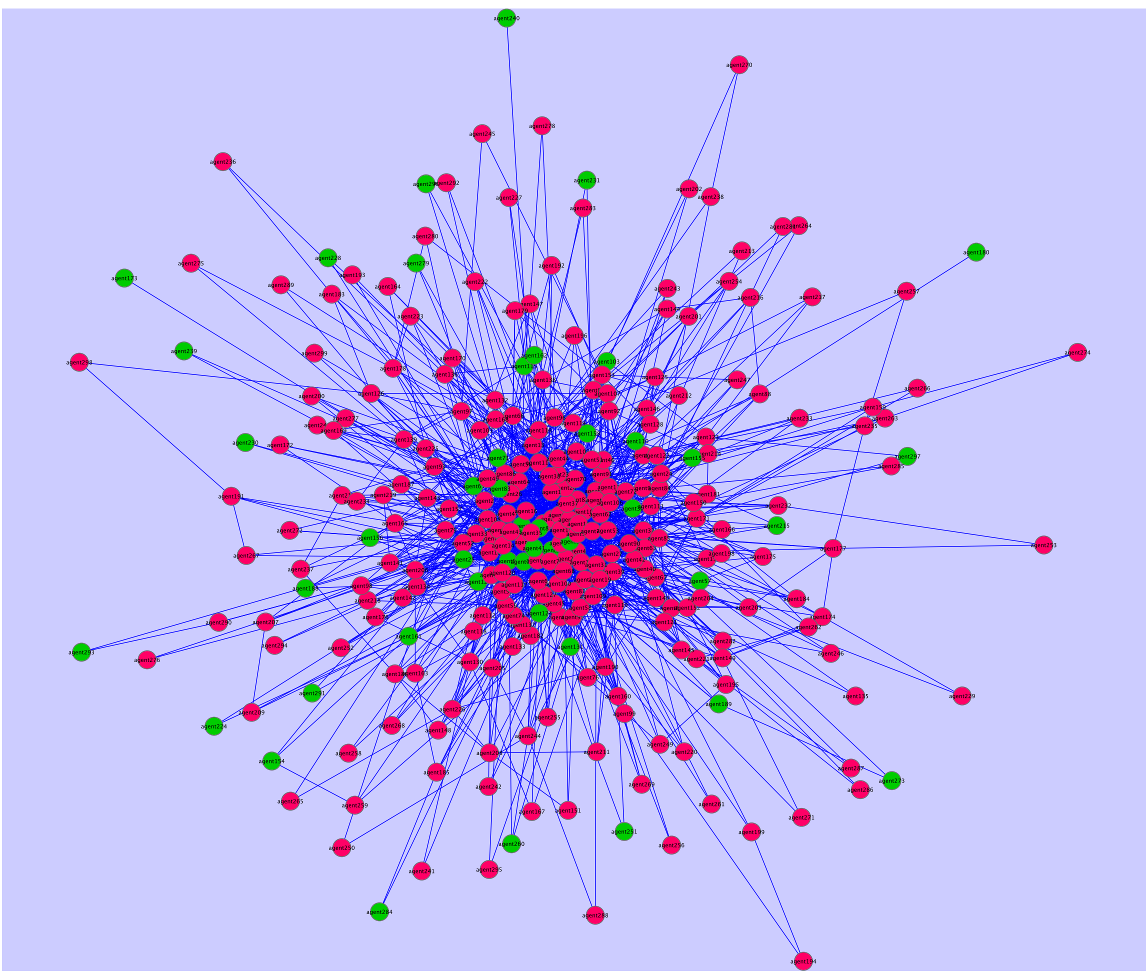


Figure 4.34: The system at $t = 1000$.

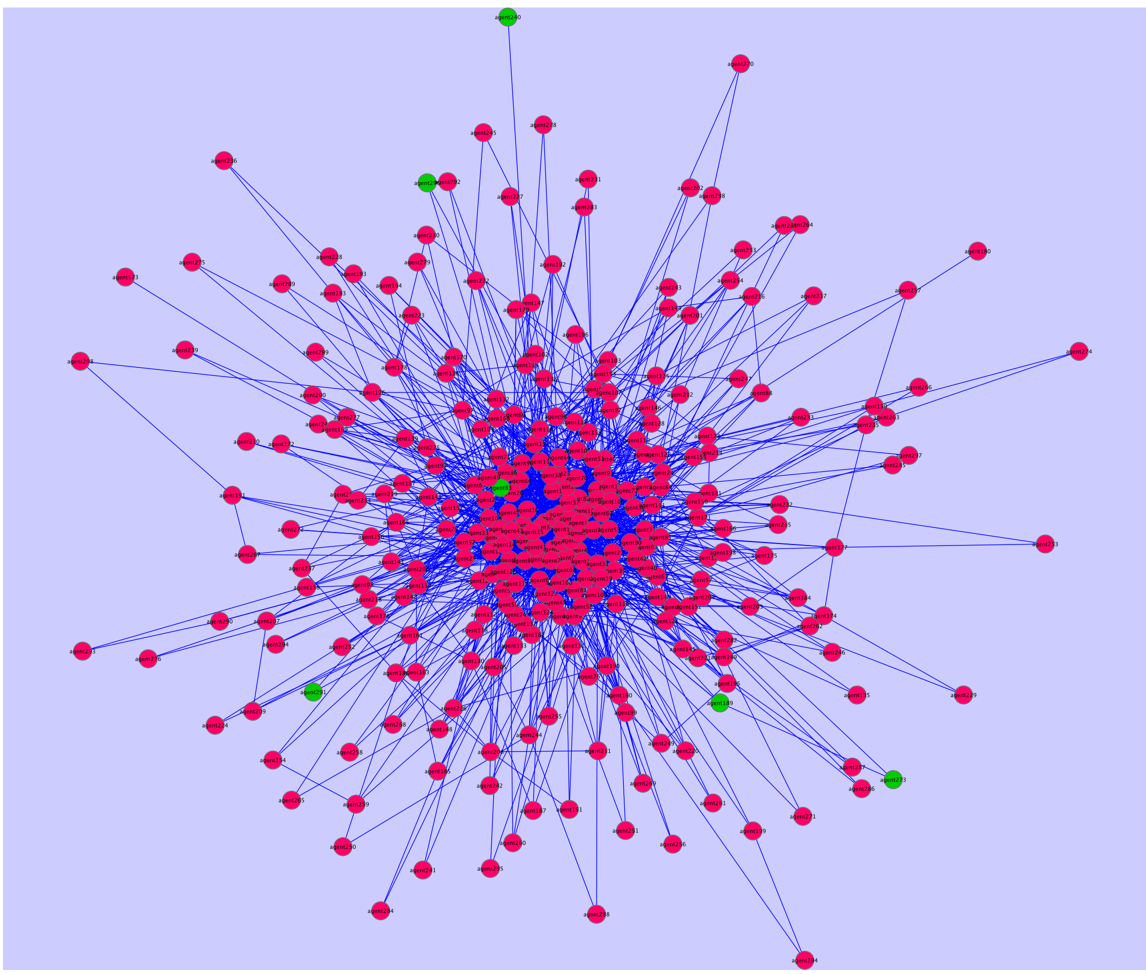


Figure 4.35: The system at $t = 1500$.

t	Frac in Agreement
0	.5667
500	0.6833
1000	0.85
1500	0.98

4.12 Conclusion

The aim of this chapter was to investigate, in detail, the process of Information Gathering which is one of the three key components of the Generalized Agreement Process. We articulated the *Fundamental Agreement Tradeoff* which captured the interaction between effort, information and agreement time. Investigation of this tradeoff is a critical part to developing a general theory of agreement.

To greater understand the tradeoff we need to understand how each element interacts with each other; that is, to execute an interaction requires some resource expenditure, interactions impact information, and information impacts agreement. To say that we understand the fundamental agreement tradeoff means that we must understand each causal link. In addition, we should be able to utilize this understanding in order to find the optimal point in the tradeoff between agreement time and effort. This leads to four questions which we have tried to address in this chapter:

1. How much effort (resource expenditure) does an IGI require/expend? Effort directly correlates to cost.

2. How does the number of IGIs affect the accuracy of the information gathered?

3. How does accuracy of information gathered impact time to agreement?

4. What is the best point in the tradeoff?

The answers to these questions will help us explore the fundamental agreement tradeoff. To explore questions 2-4 in a more quantitative manner we focused on studying the dynamics of a novel agreement protocol that allows for the direct modulation of the effort of the agent by controlling the frequency of IGIs that can be executed. We called this the Sampled Majority Vote (SMV) protocol.

A summary of the main contributions of this chapter are:

1. Developing a greater understanding of the information gathering process.

2. Developing the concept of an IGI and its two main characteristics, noise and accuracy.

3. Articulation of the fundamental agreement tradeoff.

4. Developing the novel SMV protocol that allows us to explore the fundamental agreement tradeoff.

5. Understanding the types of inaccurate information that can occur with limitations on IGIs: i.e Mistaken Majority error and Difference in Strength error.

6. Understanding the impact of inaccurate information on agreement time.

7. Developing the ICCC metric to quantify the fundamental agreement tradeoff.

8. Finding the optimal sampling fraction for different ratios of cost.

4.13 Appendix: Proofs of theorems

Theorem 1 is:

Theorem. *Let N be any even number and let $\rho_1 = .5$. Then:*

$$h(N, 0.5N, m, x) = h(N, 0.5N, m, m - x) \tag{4.16}$$

for all $N, m, x \in [0, \ldots, m]$

Proof. Since $N - 0.5N = 0.5N$ we have:

$$h(N, 0.5N, m, x) = \left(1/\binom{N}{m}\right)\binom{0.5N}{x}\binom{0.5N}{m-x} = \left(1/\binom{N}{m}\right)\binom{0.5N}{m-x}\binom{0.5N}{x} = h(N, 0.5N, m, m-x) \tag{4.17}$$

□

Theorem 2 is:

Theorem. *Let N be any even number, $\rho_1 = .5$ and let $m = 2q + 1$, an odd number. Then :*

$$Q_H(N, 0.5N, m, \lfloor m/2 \rfloor) = 0.5 \tag{4.18}$$

Proof. We know that:

$$\sum_{x=0}^{m} h(N, 0.5N, m, x) = 1.0$$

Since $m = 2q + 1$, there are $2q + 2$ possible values for x. By Theorem 1 half of these elements are the same as the other half. That is the probability for $x = 0 \ldots \frac{m-1}{2}$ is the same as $x = \frac{m+1}{2} \ldots m$. Note that

$\lfloor m/2 \rfloor = \frac{m-1}{2}$ for odd m. Thus we get:

$$2 \cdot \sum_{x=\frac{m+1}{2}}^{m} h(N, 0.5N, m, x) = 1.0$$

However, $\sum_{x=(m+1)/2}^{m} h(N, 0.5, m, x) = Q_H(N, 0.5N, m, \lfloor m/2 \rfloor)$ since $\lfloor m/2 \rfloor + 1 = \frac{m+1}{2}$; thus we get the desired result. □

Theorem 3 is:

Theorem. *Let N be any even number, $\rho_1 = .5$ and let $m = 2q$, an even number. Then :*

$$Q_H(N, 0.5N, m, m/2) = \frac{1}{2}(1 - h(N, 0.5N, m, m/2)) \tag{4.19}$$

Proof. We know that:

$$\sum_{x=0}^{m} h(N, 0.5N, m, x) = 1.0$$

Since $m = 2q$, there are $2q + 1$ possible values for x. By Theorem 1 nearly half of these elements are the same as the other half. That is the probability for $x = 0 \ldots \frac{m}{2} - 1$ is the same as $x = \frac{m}{2} + 1 \ldots m$. However there is the extra term for $x = \frac{m}{2}$. Thus we get:

$$h(N, 0.5N, m, m/2) + 2 \cdot \sum_{x=\frac{m}{2}+1}^{m} h(N, 0.5N, m, x) = 1.0$$

And noting that $\sum_{x=\frac{m}{2}+1}^{m} h(N, 0.5, m, x) = Q_H(N, 0.5N, m, m/2)$ we get the desired result. □

Theorem 4

Theorem. *For even N and odd m it is true that:*

$$Q_H(N, N\rho_1, m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1 = 1 - Q_H(N, N(1-\rho_1), m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1$$

Proof. We know that for odd m, $\left\lfloor \frac{m}{2} \right\rfloor = \frac{m-1}{2}$. Also, note that:

x	$\cdots$	$\lfloor \frac{m}{2} \rfloor - 2$	$\lfloor \frac{m}{2} \rfloor - 1$	$\lfloor \frac{m}{2} \rfloor$	$\lfloor \frac{m}{2} \rfloor + 1$	$\lfloor \frac{m}{2} \rfloor + 2$	$\cdots$
	$\cdots$	$\frac{m-5}{2}$	$\frac{m-3}{2}$	$\frac{m-1}{2}$	$\frac{m+1}{2}$	$\frac{m+3}{2}$	$\cdots$
$m - x$	$\cdots$	$\frac{m+5}{2}$	$\frac{m+3}{2}$	$\frac{m+1}{2}$	$\frac{m-1}{2}$	$\frac{m-3}{2}$	$\cdots$

$$
\begin{aligned}
Q_H(N, N\rho_1, m, \left\lfloor \frac{m}{2} \right\rfloor) &= \sum_{x>(m-1)/2}^{m} h(N, N\rho_1, m, x) \\
&= \sum_{x>(m-1)/2}^{m} h(N, N - N\rho_1, m, m - x) && \text{by Equation (4.6)} \\
&= \sum_{x\leq(m-1)/2}^{0} h(N, N - N\rho_1, m, x) && \text{This the definition of the CDF.} \\
&= 1 - Q_H(N, N(1 - \rho_1), m, x) && \text{By Equation 4.4.3}
\end{aligned}
$$

Now subtracting $\rho_1$from each side gives the required result. □

Theorem 5 is:

Theorem. *For even N and even m it is true that:*

$$
Q_H(N, N\rho_1, m, \left\lfloor \frac{m}{2} \right\rfloor) - \rho_1 = 1 - Q_H(N, N(1 - \rho_1), m, \left\lfloor \frac{m}{2} \right\rfloor) - h(N, N(1 - \rho_1), m, \frac{m}{2}) - \rho_1
$$

Proof. We know that for even m, $\left\lfloor \frac{m}{2} \right\rfloor = \frac{m}{2}$. Also, note that:

$$
\begin{aligned}
Q_H(N, N\rho_1, m, \frac{m}{2}) &= \sum_{x>m/2}^{m} h(N, N\rho_1, m, x) \\
&= \sum_{x>m/2}^{m} h(N, N - N\rho_1, m, m - x) \\
&= \sum_{x<m/2}^{0} h(N, N - N\rho_1, m, x) \\
&= H_{cdf}(m/2 - 1) \\
&= 1 - Q_H(N, N(1 - \rho_1), m, m/2 - 1) \\
&= 1 - Q_H(N, N(1 - \rho_1), m, m/2) - h(N, N - N\rho_1, m, m/2)
\end{aligned}
$$

Now subtracting $\rho_1$from each side gives the required result. □

Chapter 5

Information use: Agreement in complex state spaces

5.1 Introduction

In Chapter 3 we identified two major axes that can be used to define and classify agreement problems, that of *Information Gathering* and *Information Use*. In Chapter 4 we focused on the information gathering phase. We articulated the components of the information gathering process and developed the Fundamental Agreement Tradeoff between the effort required to gather information and its impact on agreement time. We explored this tradeoff via the novel Sampled Majority Vote protocol.

In this chapter we focus on *Information Use* – how an agent uses the information from the information gathering phase. This process is also called a "decision rule" or "decision process". *Can we call this a "decision rule" – the process by which an agent decides on which state to converge upon?'*

One cannot really split these two processes apart as cleanly as presented here; the information an agent gathers will be influenced by how the agent can use the information, and clearly an agent will make use of whatever information there is. We will try to identify these links as we proceed.

We start by outlining the process of information use through an idealized schema. The essence of the information use phase is the choice of a state for the active agent to move to; this is influenced by the states of other agents (information about which was gathered in the previous phase), which states an agent can move to (are accessible), how "far" an agent can move, and the reward for moving to the other state. We describe the impact of each of these components.

In determining how far an agent may move in the state space it is important to consider the organization of the state space. We discuss a general class of state spaces, which we call *Vector state spaces* based on n-dimensional real vectors, that can capture a wide array of phenomena.

The size of a vector (each element is called a *feature*), the values each feature can take, and the interdependence between the values of a feature are the main properties that distinguish the difficulty of agreement in a vector state space. Low complexity vector state spaces are ones where there are few features which can take on very limited values, and the features are independent. On the other hand, in high complexity vector state spaces there are multiple features that can take on a large number of values and, critically, the values of the features are dependent upon each other.

We describe varying complexity vector state spaces and provide examples from various domains. We are interested in agreement in high complexity vector state spaces for the reason that we can capture properties of many more complicated phenomena via this state space.

One of the key questions when studying high complexity vector state spaces is *which* state is converged upon. The interdependence of features and the impact of limitations on how an agent can move in the state space can greatly impact the state that is finally agreed upon.

In order to start to understand agrement in high-complexity vector state spaces we developed the Distributed Constraint Agreement (DCA) framework[1] which captures and parameterizes the issues of:

- Bounded rationality and accessibilty relation via the effort limitation parameter ϵ.
- Interaction limitiations, both in the choice of the interaction set (via an interaction relation specified as a graph) and in the amount of information that can be communicated per IGI which is limited by the communication limitation parameter κ.
- Complexity of the state space through the definition of multiple integer valued variables representing features, $\mathcal{X} = \{x_0, x_1, \ldots, x_{m-1}\}$, that influence each other through a set of constraint functions, $\mathcal{C} = \{f_0, f_1, \ldots, f_{q-1}\}$.

The DCA framework specifies a constraint network for each agent and limits the communication and effort of each agent. A solution to a DCA problem is a shared setting of the variables of each agent – i.e. the agents have agreed upon a setting of the variables. We can explore a wide variety of scenarios by varying ϵ, $\kappa$and $\mathcal{C}$. Since a DCA problem is related to a constraint optimization problem we are able to leverage some of the solution concepts from DisCOPs. A DCA problem can also be viewed as a mix of constraint satisfaction (finding a solution for the constraint problem) along with dynamical systems (coming to an agreement on a shared solution state).

Via the DCA we can begin to capture processes and phenomena that occur in more complex social influence settings. Where previously cultural dynamics were modeled as a set of independent features (see for instance, [Axelrod, 1997]) the DCA framework allows us to define influences between features and vary the complexity of interaction between features.

The DCA framework explicitily uses the notation of constraint optimization problems (COPs) because COPs were designed with the goal of capturing interactions between different elements. The added benefit is

[1]We will show that the DCA framework can be modeled via the DOA framework)

that we can leverage the tremendous amount of work in COPs to help us understand how the interdependence of features can impact finding a solution. Our unique contribution is to study this in the context of multiple agents solving the same COP in parallel but with severely limited communication and effort restrictions. Such settings capture important properties of social systems thus allowing us to utilize DCA in building an Agent Based Social Simulation.

In the final part of this chapter we show how DCA can capture some important aspects of linguistic phenomena. We explore the use of the DCA framework in capturing properties of chain shifts in language, in particular we provide evidence for how a simple iterative improvement algorithm can result in chain shifting in a population of phonological agents. The results of these numerical simulations provides some support for the theory that population migration caused a chain shift.

To summarize, in Section 5.3 we describe an idealized information use process that allows us to describe a wide variety of information use processes. In Section 5.2 we describe how the state space can vary in complexity and identify three particularly interesting state spaces.

Section 5.4 begins by identifying the need for the exploration of agreement in more complicated state spaces. We then describe the DCA framework, show how it relates to a Distributed Constraint Optimization Problem, and show how DCA is related to the DOA framework.

In Section 5.6 we describe how we can use the DCA framework to explore issues in vowel chain shifting. This provides insight into agreement under constraints as well as shedding light onto the usefullness of varying theories of the GEVS.

Portions of this chapter were previously published in [Lakkaraju et al., 2009] with coauthors Les Gasser and Samarth Swarup.

5.2 State space complexity

Up until now we have only considered very simple state spaces, for instance in Chapter 4 we focused on the binary state space case. However, there are many problems in which a binary state space cannot adequately capture the complexities of real situations. In order to form a general theory of agreement, we must be able to describe more complex state spaces.

In the following, we describe several settings in which a binary state space does not capture the complexities of the problem. We then describe the “Vector state space model” that can be used to capture many of the characteristics of complex state spaces.

5.2.1 Examples of complex state spaces

The first change we can make to the binary state space is to increase the number of values. This can range from having a finite number of values to a continuous space. These kinds of situations occur in many domains such as distributed function calculation, or in the Potts model, which is similar to the Ising model except with multiple states that a variable can be in.

A single variable, even with a multitude of values, does not eloquently capture situations in which there are a multitude of behaviors, each of which can take a different value. For instance, consider the "principles-and-parameters" framework [Gibson and Wexler, 1994] for language. In this linguistic framework language is characterized by a set of principles called the "Universal Grammar" (UG) and a set of parameters that can take on the values 0 or 1. The universal grammar is shared by all humans . The setting of the parameters determine the specific language, for instance English would have one setting of parameters and Hindi another.

In the principles and parameters approach the space of all languages can be defined as binary vectors of size n, where n is the number of parameters. In Section 5.3 we describe a popular language learning algorithm called the "trigger learning algorithm". To capture these situations the state space should have multiple variables that can take on a multitude of values.

Oftentimes behaviors are linked with each other. An example from [Mason et al., 2007] is apt:

> If social influence leads the target to change a given attitude (e.g., about the relative merits of Israeli versus Palestinian positions in their ongoing conflict), over time other related attitudes (e.g., views of the trustworthiness of media sources that take different perspectives on the conflict) may shift accordingly. In other words, because people's cognitions are linked and interdependent, often a change in one will lead to corresponding adjustments in others . . .

The basic idea here is that cognitions (in this context, thoughts, beliefs, etc) impact each other. Thus, one cannot just modify one cognition and not affect the others in some manner. This is a situation in which there are constraints between different elements. Cognitive dissonance theory argues that humans try to reduce the "dissonance" between their cognitions – i.e. find the set of beliefs that are most consistent with each other [Wilson and Keil, 1999, Dissonance]. Based on this idea, humans are actively searching a state space of possible settings of beliefs in order to find the most consistent option.

There are numerous computational models of this search for a coherent or consistent set of beliefs, see [Thagard and Verbeurgt, 1998, Shultz and Lepper, 1996].

Language is also a domain that has a complex state space in which multiple features interact with each other. In Section 5.6 we will describe a simple computational model of phonological change that incorporates the use of constraints.

5.2.2 Vector state space model

In this section we develop the *Vector state space model* – a simple model of states that can capture many of the elements from above.

At the heart of the matter, a state is a representation of the behavior or cognitive state of an agent. It can represent some action an agent is going to take – like whether to abort or to commit a transaction; or some sensor readings(s) of an agent, such as the temperature; or a representation of the agents language. These three examples vary significantly in several features.

In this work we view a state space as some kind of Cartesian space. Every state is a m-tuple in a subset of $\mathbb{R}^m$. We call each dimension a *feature*, thus an individual state is some setting of the m features comprising a state. There are three important properties which characterize different types of state spaces: the number of features, the values a feature can take and the interaction between different features. Different options for each of these properties are described below.

Number of features The options are *single* when there is only one feature, and *multiple* when there are more than one.

Value of the features The number of values each feature can take on. First there is a difference between *discrete* and *continuous*; in the former there is a discrete set of possible values, in the latter there is a continuous range of possible values. The number of discrete options and the range in the continuous case are differentiators.

Interaction When the value of each feature does not depend upon any other feature we call the features *independent*; when the value of a feature depends upon the values of any of the other features we call the features *dependent*.

There are four settings of the properties that capture many of the settings described above.

Single feature, Multiple discrete value This is the "simplest" case where a single feature is agreed upon. As we have seen in the previous chapter there is a lot of work in the binary discrete value case. There is some work on situations in which there are more than 2 options.

Single feature, Continuous value This setting of parameters captures problems in control theory like distributed function calculation [Olfati-Saber and Murray, 2004, Olfati-Saber and Murray, 2003, Saber and Murray, 2003, Sundaram and Hadjicostis, 2008a, Sundaram and Hadjicostis, 2008b]

Multiple independent features, either discrete or continuous In this case there are multiple features where each feature can be either discrete or continuous, and each feature may have a different number of values or range. This setting captures the state space in models like the principles and parameters framework, or the cultural dissemination model in [Axelrod, 1997].

Multiple dependent features, either discrete or continous In this setting the value of each feature is influenced by the values of some of the other features of the state. Since the features are dependent this problem does not decompose into a set of independent agreement problems. This setting captures elements of situations in which agents are trying to find a consistent set of beliefs etc.

5.2.3 Bounded rationality and communication

Bounded rationality is the idea that agents must operate under conditions of limited resources and knowledge. In these settings it is difficult (or impossible) for agents to make optimal decisions, instead they must make the best decision they can under the resource limitations that they have [Wilson and Keil, 1999, Bounded Rationality].

The concept of bounded rationality plays an important role in the study of human behavior in economics (for a review see [Conlisk, 1996]) and artificial intelligence (see [Russell and Norvig, 2003]). To capture the dynamics of sociological agreement processes we must understand the role of bounded rationality on the process of agreement.

From a technical point of view, there are several reasons why communication may be limited. In chapter 4 we talked about the effort to communicate in terms of power usage in energy constrained sensor networks.

Communication restrictions are restrictions on the ability for agents to communicate information about themselves to others. We can define two categories of communication restrictions:

- Restrictions on how many times one can communicate.
- Restrictions on what can be communicated.

The first type of restrictions was described extensively in chapter 4

When discussing complex agreement spaces the second issue become increasingly important. As a representation of a complex set of cognitions, the idea that there is perfect communication of beliefs between humans is implausible. Language barriers can inhibit accurate communication of cognitions as well.

Linguistic knowledge comes implicitly via interaction, rarely from explicitly setting out to learn a language. Knowledge of another language comes through interaction and is limited to the tasks in which the language is spoken. Thus we can expect that there are significant communication limitations when passing linguistic knowledge. While we are focused on limited information about a language per interactions, there are many studies on the limited *cumulative* knowledge of a language a child gets. This is usually referred to as the *Poverty of Stimulus* argument [Wilson and Keil, 1999, Poverty of Stimulus] and has been used to justify the universal grammar position.

There are numerous reasons why limitations on communications can exists in technical systems. First, in open systems with heterogeneous entities information must be translated into an exchangeable format which might not exist, or if it exists, be a format that is too expensive to convert to.

Secondly, communication might be an expensive action. This was discussed in chapter 4, however we provide some examples of communication restrictions in MAS here.

There has recently been a stream of work on studying communication costs in the the distributed sensor interpretation (DSI) task. [Shen et al., 2006,Shen et al., 2003,Shen et al., 2002] model a DSI task as inference on a two level Distributed Bayesian Network. They study the interaction and communication between agents via a Dec-POMDP model. Through this model they show the impact of increased communication on solution quality. Their analysis is focused on achieving a good solution and not on the time to reach agreement. In addition, they have focused on situations with a small number of agents (only 2).

[Koren, 2005] extended the models from [Shen et al., 2002] to multiple agents and provided a higher resolution understanding of the communication cost than the simple communication cost model in [Shen et al., 2002]. However, a complete communication topology was still assumed between agents (every agent could communicate with every other agent) although the cost of communication varied with the distance between agents.

[Blumrosen et al., 2007] studied auctions under severe communication constraints; where agents were only allowed to send messages of length t bits. [Blumrosen and Feldman, 2006] study mechanism design in situations where the action space cannot fully represent the private information (the types) of the players.

Agents might intentionally restrict communication in order to confer privacy – that is agents do not want to reveal information about their state. One example of this is distributed meeting scheduling – agents on the one hand want to quickly schedule a meeting time that is acceptable to everyone else, but on the other hand do not want to reveal information about their full schedule, which would make scheduling quicker. Meeting scheduling has been extensively studied as an instance of a Distributed Constraint Satisfaction Problem (DisCSP) problem. [Wallace and Freuder, 2005] have laid out the tradeoff between privacy and efficiency in the meeting scheduling problem.

5.3 The information use process: state update algorithm

Figure 5.1 is a highly idealized diagram of the process by which an agent will choose which state to move to. In reality, many of the steps are combined together. However, this diagram is useful in laying out the space of agreement protocols and identifying the key differentiators between agreement protocols.

Figure 5.1 should be read from left to right. In this schema we view the choice of state by an agent as, literally, the active agent choosing a particular state to move to from the set of all possible states. This choice is impacted by several factors that we describe below. As we progress to the right in the figure, we identify the factor and the resulting state set that occurs after incorporating that factor.

The bottom of Figure 5.1 describes the most complex part of the choice, where the active agent integrates information from the information gathering process and decides how to tradeoff intrinsic vs. frequency dependent reward.

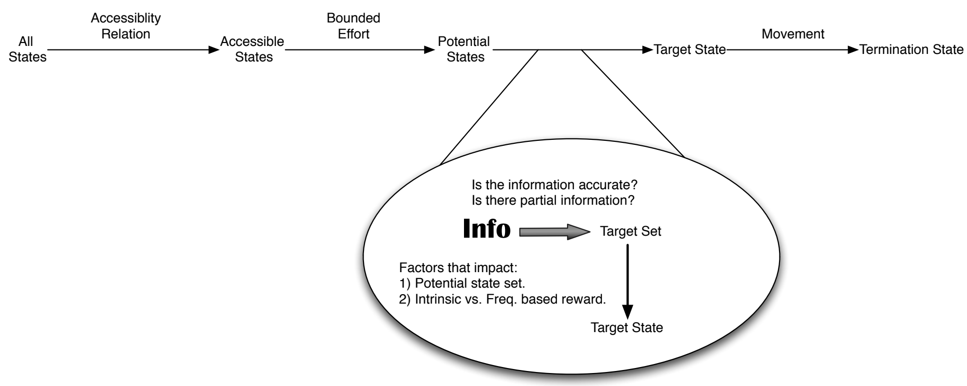


Figure 5.1: An abstracted and idealized schematic of the information use process. In reality, many of these steps are combined together. However, for our purposes this schematic provides a way of organizing and understanding the realm of information use processes.

5.3.1 Accessiblity relation

The accessibility relation specifies the effort required to move between states for each individual agent. Some states may be impossible to move to, i.e. inaccessible.

For instance, suppose we have a state space that is comprised of all currently existing languages. For an agent to change state means an agent learns the new language. In this case the accessibility relation will define the effort required to learn a new language. Thus, if a human (our agent in this setting) is currently in state “American English” then state “British English” require little effort to learn because of the similarity in vocabulary etc. Essentially, the accessibility relation, in this case, captures the differences between states that affect the ability for an agent to change states.

On the other hand, moving from "American English" to "Mandarin"' would require a great deal of effort, since Mandarin is quite different from English (for one thing, it is a tonal language).

Suppose we classify the dancing of honeybees as a language in our space[2]. The physical requirements of the honeybee language (being able to dance on the hive and emit a certain type of odor) make it impossible for a human to learn, thus it is an inaccessible state – which we denote by setting the cost to infinity.

The first step in the information use process is to identify the states that are impossible to reach and remove them from consideration. We define the set of *accessible states* for an agent α_i from state j as the set of states that have cost less than infinity.

5.3.2 Bounded effort

Bounded rationality is the idea that an agent has only limited resources to solve a problem, and thus does not make the best decisions, but only decisions that are "good enough" [Russell and Norvig, 2003, 973]. In our context, bounded rationality is seen as a restriction on the amount of effort an agent can exert to change state. We thus denote this as "Bounded Effort". Every agent has some level of effort that they can exert every instance that they want to change state. We define the *potential state set* as the set of states that satisfy the bounded effort limit for an agent at the current timestep.

In section 4.8.1 we defined the ICCC metric which defined effort as a quantity that relates to the expenditure of resources by an agent. The concept of bounded effort defined here extends that.

The combination of the accessibility relation and the bounded effort of an agent greatly influences the dynamics of state change by limiting the possible states an agent can move to. When an agent has unbounded effort, i.e. an agent can consume as much effort as needed, the entire state space is available for an agent to move to (with the exception of the inaccessible states). In this case any organization of the state space is lost and the problem becomes, essentially, settling on a value of a single feature with multiple possible values.

As soon as there is a limit on effort, the problem cannot be simplified to the single feature, multiple possible values setting.

As an example of bounded effort, consider the "trigger learning algorithm" [Gibson and Wexler, 1994] that works with the principle-and-parameters model of language described in Section 5.2. In the trigger learning algorithm humans change the value of one of their parameters when a sentence arrives that is not compatible with their current set of parameters – this sentence is called a "trigger".

[Matsen and Nowak, 2004] study the situation where a population of agents are coming to agreement on

[2] I think it would be safe to say that honeybee dancing is more of a signaling system than an actual language.

a particular language. Under the assumption that flipping a parameter value requires one unit of effort, the effort bound is then 1. This leads to a hypercube state space, where each node is a language, characterized as a set of binary variables, and the edges connect languages that are separated by a single parameter difference. In this case there are no inaccessible states, however there are pairs of states that are quite different and will require a significant amount of effort to change between. s

This example shows that the notion of bounded effort is an important one that occurs in at least a linguistic setting.

5.3.3 Choice of a target state

The penultimate, and the most complicated, step is to choose a *target state* – a state that the active agent wants to move to. The final state that an agent actually moves to is called the *termination state* because it is the state which the active agent wants to change to, i.e. terminate at. The termination state might differ from the target state because the process of changing state might be noisy and an agent might end up in a different state than intended.

There are numerous factors that influence the choice of a target state. Figure 5.1 illustrates this process in the ellipse at the bottom of the Figure.

We do not depict this in the diagram, but in some cases agents are endowed with memories that can store information or reward. The HCR decision rule explored in [Shoham and Tennenholtz, 1997, Delgado, 2002] utilizes a finite size memory.

From information to a target set and the impact of partial information

The first step is to use the information from the information gathering phase to identify the state or states that the other agents have; we call this set of states the *target set*.

We identified two types of information in the previous chapter, single agent state information and aggregate agent state information. Single agent state information provides a clear target state. In the case of aggregate agent state information a decision must be made on how to define the target set. In some cases, like the majority-rule, the target state is calculated by finding the majority state based on the distribution of states as learned from the aggregate agent state information. On the other hand, if the agent were to use the probabilistic protocol described in Section 4.2 the target state would be randomly chosen from the distribution of states identified from the aggregate information.

For both types of information the question of accuracy comes up. This is in part determined by the IGI executed by the agent and how accurate the IGI is. If the information is inaccurate the agent could generate a target state that it thinks is the majority state, but really is not. We explored this phenomena in detail in chapter 4. Additional inaccuracies can occur, which were mentioned but not explored in the previous chapter.

One important type of inaccuracy that we consider in more detail here is of *partial state information*. Partial state information is information that only partially describes a state – there are several possible states that are consistent with the information known about a state. For instance, you might be trying to find a house on a block and you know that the house is red, however there are several red houses. Partial state information issues occur more often in situations where there is a complex state with numerous features.

As an example, consider a setting where the state of an agent is a simple compositional language. Figure 5.2 illustrates a *description game* between two agents with compositional languages. A part of the speakers language is shown in the upper left of the figure. Every concept (represented as logical predicates) is represented by one word (and vice-versa, each word represents only one concept). For instance, the speakers language (top left) encodes the concept of chasing ($Chase(\cdot)$) with the word **gavagai**.

The grammar indicates the order words should be put together; the speakers language has the event first, the patient second and the agent third. Thus"**gavagai mischal wooziar**" is how the speaker describes a scene in which a dog is chasing a cat. A state in this example is a complex entity consisting of the mappings from concepts to words as well as the grammar of an agent.

Suppose the agents are using an IGI in which they play a simple description game. The speaker and hearer jointly observe a scene (here the scene is a dog chasing a cat), and the speaker encodes the scene via their language. The only information the hearer gets is the shared scene and the speakers sentence. In order to reach agreement the hearer will try to change its language to match that of the speakers; however the information in this IGI is partial information and thus does not unambiguously identify the speakers language. In particular there are 6 languages that are consistent with the given information, shown at the bottom of the figure.

Thus, in this situation the hearer is left with a target set of size 6 rather than a target state. Without any other information the hearer has no way of knowing which of the consistent languages is the actual state of the speaker – the hearer agent might choose to modify its language towards the "wrong" (in the sense of not the same as the speakers) state. This can have a drastic impact on time to agreement (see [Lakkaraju and Gasser, 2008a] for further exploration of this issue and a technique to help resolve the uncertainty of the speakers language).

In the context of vector spaces an agent could receive partial information about a state by only getting the values of a subset of variables in the state. In this case, the number of potentially consistent states is

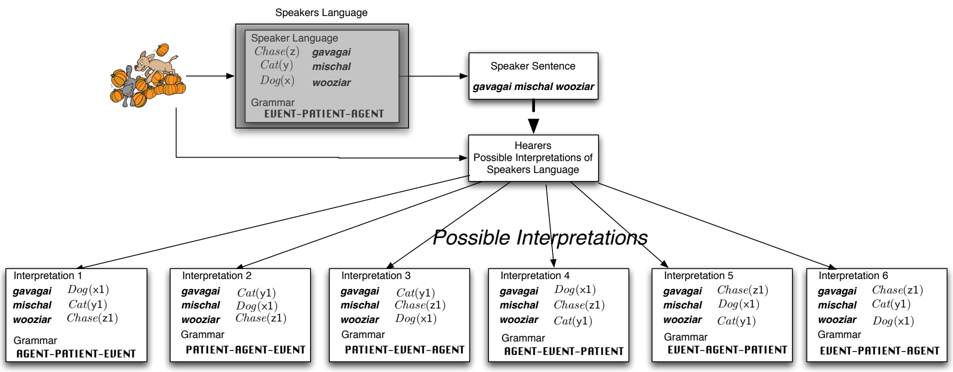


Figure 5.2: At the top left is a shared scene of a dog chasing a cat that is viewed by both the hearer and speaker. The speakers language encodes the scene as "**gavagai mischal wooziar**". The hearer can interpret the sentence with reference to the shared scene in 6 different ways, or strategies. Only one of the interpretations matches the speakers language (shown in the shaded area). Each interpretation is a hypothesis about the speakers language based on an information gathering interaction. For example, interpretation 1 hypothesizes that the speakers language encodes $Dog(\cdot)$ as gavagai, $Cat(\cdot)$ as mischal and $Chase(\cdot)$ as wooziar; and that the speakers grammar is of the form Agent-Patient-Event

easily quantified.

Choosing a target state

Once a target set has been identified, the next question is to pick a particular state to move to. This choice is dependent upon two factors:

1. The potential state set – i.e. where an agent can move to.
2. The tradeoff between intrinsic and frequency based reward.

First, the target state should be a state that it is possible to move towards. Suppose we take the simple situation where an agent only gets single agent state information, like the voter model. In this situation the target state is a single state. However, if the state of the other agent is not in the active agents potential state set there is no way to move to that state. In this case, there are two options. The first option is that the active agent might not change state. The second option is for the active agent to change to a state that is the "closest" to the target state.

The second option is complicated. First, we must define what it means to be the closest state. There are two options, the first is to consider closeness to be a function of how similar the states are. In the vector state space domain, this could be the Manhattan or Euclidean distance between two states. Using this method, the active agent will try to match as many features of the target state as it can.

The second option is to consider closeness in terms of effort. Two states are close if the effort to reach one state from the other is small. For two states p and q and an agent α_i, the closest state s that is in the potential state set of α_i is the state that minimizes the effort to subsequently get to q. That is, for s in the potential state set of α_i,

$$s = \arg\min_{x \in \Sigma} \Delta(\alpha_i, x, q)$$

.

Note that s depends upon the state from which one starts – s might not be the closest state to q if the active agent starts from a different state.

As an example where this definition of closeness might be useful, consider a set of UAVs that need to agree upon a location to meet in a square grid. The state space will be the possible locations and the agents are the UAVs. We can use Euclidean distance to measure the similarity of states. If the environment has obstacles (such as walls) then the closest point in terms of similarity might be separated from the target state by an impassable obstacle. However, when effort is measured as the amount of movement to get to a state, we can calculate the best state and eventually reach the target state.

Oftentimes the two closeness measures coincide, and we will assume that they do for the most part.

To compute these measures, the agent has to have access to the accessibility function in order to identify which of the states in the potential state set has the minimum value. This might not be possible, in which case the target state might just be randomly chosen from the potential state set.

Intrinsic vs. frequency dependent reward The other important factor is the reward an agent will receive.

In section 3.3.1 we described the intrinsic reward function, ρ that determined the intrinsic reward an agent receives from being in a state. We noted in describing the information use process in section 3.3.2 that two rewards are possibly given to an agent. One is the reward from the intrinsic value function. The other is a frequency dependent reward [Swarup et al., 2006] that depends on the number of other agents in the active agents current state (after possibly moving).

The active agent must decide how to tradeoff between the reward from staying in its current state to moving to the target state. This could be a situation where an agent might move from a high intrinsic reward state to a lower one, sacrificing intrinsic reward for being in agreement with others.

5.3.4 Movement

The final step, once a target state has been chosen, is for an agent to actually move towards the target set. Depending upon the domain and the capabilities of the agent the actual state an agent ends up in might be different from the one it *intended* to end up in. For instance, consider the case of UAVs coordinating to a single direction. A UAV might intend to turn left by, say, 45 degrees, but due to unpredictable fluctuations in the terrain, etc, finds itself having turned only 35 degrees. The actual termination state of the agent is different from the target state.

5.4 Distributed constraint agreement problems

In this section we describe the *Distributed Constraint Agreement* (DCA) problem – a way of capturing state spaces with several variables, multiple values and constraints between the variables. The basic idea of DCA is to bring together constraint optimization problems with dynamical systems.

5.4.1 Motivation

Our goal in this chapter is to understand how information use processes can impact agreement. To investigate this, we need a setting which captures:

1. Large state spaces with multiple features;
2. Constraints between features;
3. Bounded effort;
4. Bounded communication.

Constraint satisfaction problems are a well studied area in AI and multiagent systems and would, on the face of it, be a natural way to model interacting features. A constraint satisfaction problem (CSP) specifies a set of variables, the possible values those variables can take on, and a set of constraints that indicate valid combinations of values for the values. For instance, a CSP might have two variables, x_0 and x_1 which can take on any value in $\mathbb{N}$. A constraint might be that $x_0 > x_1$. A CSP is solved by setting the variables to values that satisfy the constraints. For instance, $x_0 = 5, x_1 = 3$ is a solution, whereas $x_0 = 10, x_1 = 11$ is *not* a solution since it violates the constraint.

CSPs have been used to represent a large number of problems from many different areas, including graph coloring, scene labelling and resource allocation [Tsang, 1993]. Constraint Optimization Problems (COP) are a generalization of CSPs where the set of constraints is replaced with a set of cost functions from the possible values of the variables to a cost. A solution to a COP minimizes the cost function.

In a distributed constraint satisfaction problem (DisCSP) a set of agents *own* different subsets of variables. That is, only certain agents can read or change the value of a particular variable. Agents can communicate with each other to solve the Distributed CSP. A distributed constraint optimization problem (DisCOP) is the distributed extension of the COP.

The main problem with DisCOPs is that they do not, in general, allow us to model bounded effort and communication. As [Yokoo, 2001] points out, there are several assumptions made when talking about DisCOPs:

- Agents communicate by sending messages.
- An agent can send messages to other agents iff the agent knows the addresses/identifiers of the agents.
- The delay in delivering a message is finite, though random.
- For the transmission between any pair of agents, messages are received in the order in which they were sent.

from [Yokoo, 2001, 48].

These assumptions do not fit with what we would like to capture.

First, we want to capture the idea of bounded communication – that agents cannot communicate an arbitrary number of arbitrary length messages. In the literature on DisCOPs messages have no limitations on length or on what can be sent.

Second, we want to model bounded rationality – severe limitations on the resources of an agent. We want to be able to parameterize this quantity in order to study its affect on agreement.

Thirdly, we want to study situations in which agents have extremely limited communication in the sense that they cannot communicate with all other agents. Much of the work in DisCOPs assume that agents have the potential to communicate with all other agents, even though in acuality only a small set are communicated with.

We are also interested in open systems where agents can enter and leave at will. We cannot assume that there is a total ordering of agents and variables as the number and type of agents might change. For instance, the well known Asynchronous Backtracking Algorithm [Yokoo, 2001, Yokoo and Hirayama, 2000] assumes there exists a globally known priority order that agents are given access to.

Because of the above reasons, we have developed the Distributed Constraint Agreement problem (DCA) – a type of DisCOP that allows us to study agreement in complex state spaces with significant communication and effort bounds. We describe the DCA problem below.

5.4.2 Formal model

A *Distributed Constraint Agreement* problem is a tuple $< A, \mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{G}, \epsilon, \kappa >$ where:

- $A = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ The set of agents.
- $\mathcal{X} = \{x_0, x_1, \ldots, x_{m-1}\}$ are a set of m variables. Each agent owns one copy of each variable. Let $x_{i,j}$ denote variable i owned by agent α_j.
- $\mathcal{D} \subset \mathbb{N}$ is a finite subset that defines the values any variable can take. $\mathcal{D}^m$ is the m dimensional space of all possible assignments to all m variables.
- $\mathcal{C} = \{f_0, f_1, \ldots, f_{q-1}\}$ is a set of q constraint functions where each $f_i : \mathcal{D}^m \rightarrow \mathbb{R}$ represents a constraint by associating a cost to a setting of the variables.
- $\mathcal{G} =< A, \mathcal{E} >$ is an undirected, connected graph where each node is an agent and $\mathcal{E}$ is a set of tuples that specify the edges in the graph.
- ϵ is a formalization of bounded rationality. See below.
- κ is a formalization of the limited communication constraint. See below.

We can also view the DCA as n instances of the same constraint optimization problem – each agent owns one instance of the problem.

A solution to a DCA is a shared setting of variables over all agents. The generic protocol followed in a DCA problem is called a *(ϵ-κ) protocol* and is specified below. We are interested in two questions:

1. What implementations of the (ϵ-κ) protocol lead to solving a DCA problem?
2. What is the form of the solution?

$\vec{x_i} \in \mathcal{D}^m$ is an m dimensional vector formed by the values of each variable owned by agent i. $\vec{x_i}(t)$ is the state of an agent at time t.

The set of cost functions for an agent, $\mathcal{C}_i$, represents the constraints between the variables of an agent. The set of constraint functions are shared by all agents.

Let the *cost* of a state $x \in \mathcal{D}^m$ be the summation of the set of constraint functions applied to x:

$$C(x) = \sum_{i=0}^{q-1} f_i\,(x) \tag{5.1}$$

.

The cost of a state represents how well the state satisfies the constraints.

Let

$$F_i(t) = C(\vec{x_i}(t)) \tag{5.2}$$

be the *local cost* at time t for an agent i. The local cost measures the extent to which the state of an agent satisfies its constraints. Our goal is to find states for every agent that minimize the local cost.

Let

$$F(t) = \sum_{z=0}^{n-1} F_z(t) \tag{5.3}$$

be the *global* cost at time t. The global cost function is a measure of how well the entire population of agents satisfy their constraints. This is simply the sum of local costs over all agents.

We say the system is at a *coordinated and minimized* configuration at time t when the system meets the following two conditions:

Coordination Condition $\forall_{i,j}\ \vec{x_i}(t) = \vec{x_j}(t)$

Minimization Condition $F(t)$ is minimized.

We also use the term *fully solved* when both conditions are met. When one or both of the conditions are not met we say the system is *coordinated and unminimized*, *uncoordinated and minimized*, and finally *uncoordinated and unminimized.* We also refer to these three situations as the system being *partially solved.*

We define the *distance* between two agents to be the Manhattan distance between their values:

$$d_1(\vec{x_i}, \vec{x_j})\ \sum_{z=0}^{m-1} |\vec{x_i}[z] - \vec{x_j}[z]|$$

A fully solved system is in agreement on a state that is minimal. Our goal is to define a solution protocol that will lead to agreement. In this, we are limited by bounded effort and communication.

In a DCA problem every agent follows a *(ϵ-κ) protocol*, which is a version of the Generalized Agreement Process:

Active Agent Choice Some subset of agents can be active at every time step. This is domain dependent.

Information Gathering Interaction Choice $\mathcal{G}$ specifies the allowed interaction between agents. Any agents that are neighbors of each other can interact with each other.

Interaction Agents in the DCA framework use an IGI in which they can only communicate a certain number of values. In an interaction, the agents can communicate the values of only κ number of variables to each other.

Information Use The active agents are limited in how much they can change their state. They can only move to states that are within ϵ distance of their current state.

By varying ϵ and κ we can vary the difficulty of the agreement problem. This space, however, is under explored. Figure 5.3 is a visual depiction of the interaction between different settings of communication limit and effort limit. The x axis represents the communication limit, going from communicating 0 variable values to communicating all variables values. On the y axis is the effort limit, going from 0 effort to "all" which means infinity. To the left of the y-axis is a depiction of the state space for the different values of effort limit and for a communication limit of 1.

When the communication limit is 0, no information is known, and thus there can be no agreement (except under a random process).

On the right hand side are situations in which an agent can communicate all variable values. When the effort limit is 0, a high communication limit provides nothing, since an agent cannot move to any other state. On the other hand, when the effort limit is high, the situation is effectively equivalent to a 1 feature, multiple value scenario. Each setting of the variables can be uniquely labelled, and agents can communicate and move to any of these states.

Finally, we know of some work when the communication limit is 1. The work in [Axelrod, 1997] utilizes a limited communication situation with a high effort – although in this case there were independent features. The work in [Lakkaraju et al., 2009] is a situation where there are dependent features.

For a vast majority of communication and effort limitation combinations there has been no work to our knowledge, as signified by the question mark.

5.4.3 DCA: Examples

Collective graph coloring

As an illustration we will study a simple graph coloring problem. Suppose we have the graph G_1, shown in Figure 5.4. A coloring of a graph is a labelling of each vertex with a color, which we represent as an integer. In the graph coloring problem we are trying to find a coloring of the vertices such that the color of each vertex is different from the colors of its neighbors; so if vertex 0 is labelled red, then vertex 1 cannot be labelled red.

The goal of this DCA problem is for a set of agents to agree on a shared labelling of the graph, but each agent has their own local copy of the graph that they color. Agents exchange messages about the color of certain nodes of their graph, and can change the colors of some of their nodes.

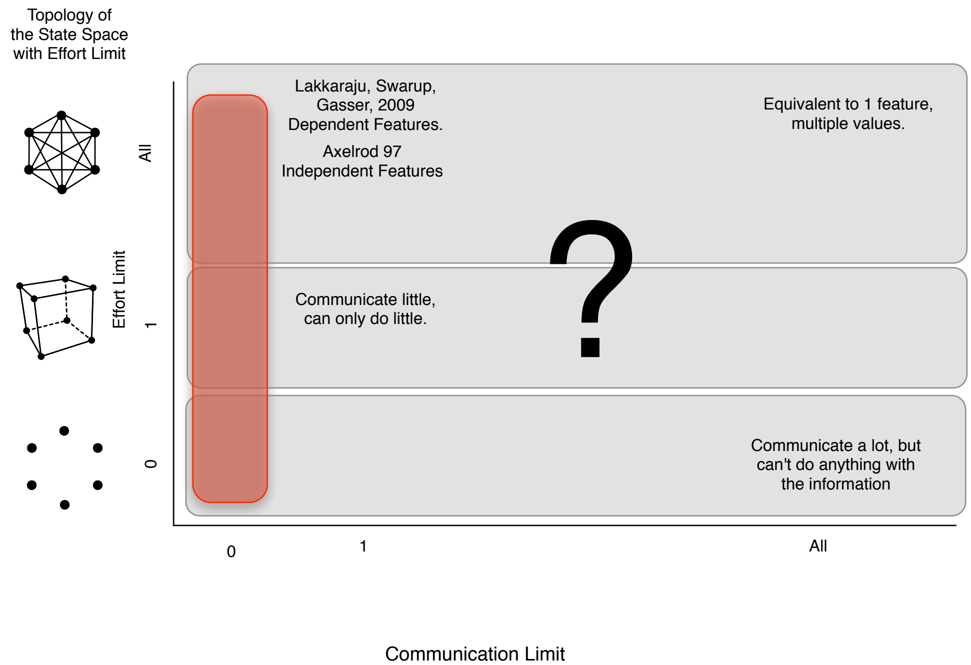


Figure 5.3: A visual depiction of the interaction between different settings of communication limit and effort limit. See text for details.

In terms of interaction, for this example we will assume that every agent can interact with every other agent – this is a complete communication topology.

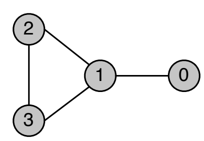


Figure 5.4: Example graph G_1

We will model each vertex in the graph as a variable, thus:

$$\mathcal{X} = \{x_0, x_1, x_2, x_3\}$$

where x_i refers to node i.

This graph can be colored with 3 colors, so we will set the domain to:

$$\mathcal{D} = \{0, 1, 2\}$$

.

In the figures we will use the color encoding of 0 = red, 1 = blue, 2 = green.

There will be a set of 12 cost functions, one function for each pair of nodes i and j. Each cost function will have the same form:

$$f_{i,j} = \begin{cases} 1 & \text{if } x_i = x_j \\ 0 & \text{if } x_i \neq x_j \end{cases},$$

where the cost will be 1 if the variables match and 0 if the variables match. If an agent has correctly colored its graph its local cost will be 0.

In a fully solved system all agents will have a setting of variables that has 0 cost and that is shared by all other agents. We can specify several different protocols that vary on: which agents are active, which agents interact and how they interact, and the way an agent changes its state. As an example here is one protocol (we do not actually know if this solves the system). We call this the *Random* protocol:

Active Agent Choice A single agent is uniformly randomly chosen to be the active agent.

Information Gathering Interaction Choice One randomly chosen neighbor of the active agent is chosen as part of the interaction set. We call this agent the "sender".

Interaction The sender picks a κ subset of their variables. The sender communicates the values of only these variables to the active agent.

Information Use The active agent changes its state by modifying the value of each of its variables to match the values the sender communicated. Changes are made until the ϵ has been reached or all the variables have been set.

For instance, suppose Agent 1 is chosen to be active and Agent 2 as its sender. Let the variables of both agents initially be:

Variable	Agent 1	Agent 2
x_0	0	2
x_1	1	0
x_2	2	1
x_3	0	2

Note that in this situation both agents have local cost 0 since there are no constraints violated. Suppose we set $\kappa = 2$ and $\epsilon = 2$. Let Agent 2 choose as a subset of variables to communicate the set $\{x_0, x_2\}$. The sender communicates to the active agent the values of its two variables: $x_0 = 2$ and $x_2 = 1$.

Under the random protocol Agent 1 will have to change $\{x_0, x_2\}$ to 2 and 1 respectively. However, this will pose a problem for Agent 1 since the effort required to make this change is 3, and Agent 1 can only expend $\epsilon = 2$ units of effort. The two possibilities for Agent 1 are to move to a state with $x_0 = 2$ or to a state with $x_0 = 1$ and $x_2 = 1$. Let's assume that in these cases Agent 1 starts by modifying the lowest numbered variable, thus Agent 1's state will be the former. After one timestep we find:

Variable	Agent 1	Agent 2
x_0	2	2
x_1	1	0
x_2	2	1
x_3	0	2

.

In this situation the local cost of Agent 1 did not change – luckily changing x_0 to 2 did not cause any constraint violations. However, if Agent 2 had chosen to communciate the value of x_1 Agent 1 would have violated a constraint when it set its variable and thus the local cost would have gone down.

As we modify ϵ and κ we can see different behavior. Suppose κ is infinite, then the constraint is the bound on effort – the sender can communicate all of its values; however the active agent will have to choose between the possible states.

In the opposite case, where ϵ is infinite, there are significant communication restrictions but the active agent can change to whichever state.

In the case where $\kappa \geq 4$ (all variables can be communicated) and $\epsilon \geq 8$ (the maximum distance between any two states is 8, thus this means an agent can change to any state) we recover multi-state voter model behavior, otherwise called the *Potts Model*. (these values of κ and ϵ are effectively infinite). There are $3^4 = 81$ different colorings of this graph (not all of which satisfy all the constraints), we can index each coloring by an integer. Since communication involves sending all the values of your coloring and the effort limit allows an agent to change to any state, the active agent will switch to the state of the sender. This is exactly the same behavior as in the voter model, except there are multiple states. Thus, the dynamics in this case should exactly match that of a multi-state voter model. Figure 5.5 illustrates example interactions with different effort and communication limitations.

The drawback to this situation is that new solutions will not be explored – one of the initial set of colorings in the population will end up being the one that is converged upon. For the random protocol, the communication and effort limitations actually induces the exploration of new states.

We can see, even from this simple example, that communication and effort limitations have a large impact on the dynamics of the system.

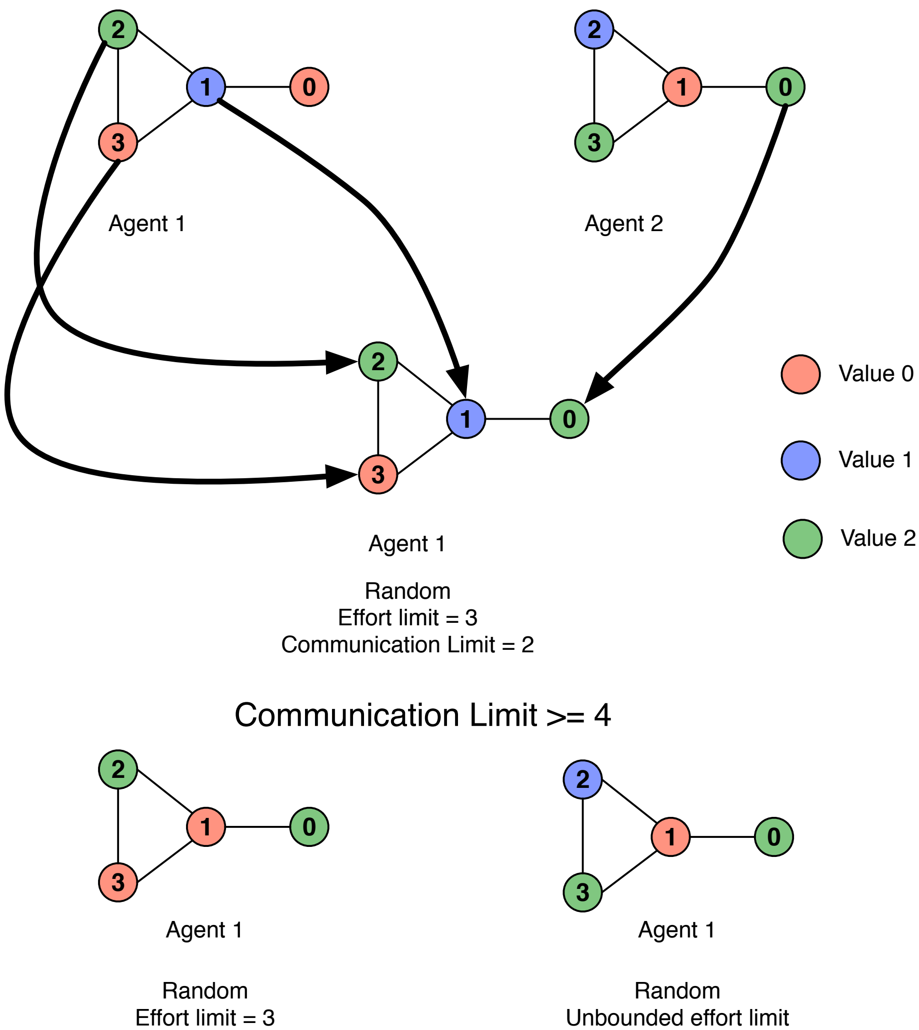


Figure 5.5: Example change in agents state.

Other examples of DCA problems

Coherence is the concept of fitting together a number of elements; these can be propositions or concepts for instance. In [Thagard and Verbeurgt, 1998] the authors propose modeling the process of coherence as a constraint satisfaction problem. The set of cognitions are represented as variables that can take on the values of 0 or 1 indicating whether an individual believes in the cognition or not.

Pairs of cognitions can have positive and negative links that reflect whether the two cognitions are coherent (make sense for both to be true) or incoherent (does not make sense for both to be true). These links can be represented as constraints.

[Thagard and Verbeurgt, 1998] focuses on the process of an individual reaching cohrerence. We can use the DCA model to understand processes of *collective coherence* where each agent is represented by a constraint network and through interaction a group of agents reach collective coherence. To our knowledge, these types of models have not been studied before.

5.4.4 DCA in the DOA

	DOA	DCA
Agents	A	$=A$
State space	Σ	$=\mathcal{D}^m$
Intrinsic value	$\rho(\alpha_i, z \in \Sigma)$	$=-F_i(z)$
Interaction Relation	Θ	$=\mathcal{G}$
Accessibility	$\Delta(i, x, y)$	$=d_1(x, y)$

Table 5.1: Summarizing the translation of DCA to DOA. For the interaction relation we must transform the graph $\mathcal{G}$ into an interaction relation.

Every DCA problem can be specified in the DOA framework. We identify how elements of the DCA problem correspond to the agents, state space, intrinsic value, interaction relation and accessibility relation in the DOA. The starting and termination states are defined in terms of the preceding relations. Table 5.1 summarizes how the elements of a DCA problem can be cast in the DOA light. We describe in more detail below.

The agents from a DCA problem are the same as agents in the DOA problem.

$\mathcal{D}^m$ describes the possible settings of variables for an agent. Since our goal in the DCA is for all agents to agree on a particular setting of the variables, and since no other element of the agent changes, $\mathcal{D}^m$ corresponds to the possible agreement space in the DOA framework.

The intrinsic value of a state defines the reward an agent gets from being in a state. A DCA problem does not explicitly provide reward, however we can view minimization of cost as the maximization of reward

– thus we can set the intrinsic reward of a state as the negative of its cost.

$\mathcal{G}$ the interaction graph corresponds to the interaction relation. See Section 3.6.1 for details on how to transform an interaction graph into an interaction relation.

The accessiblity relation in the DOA specifies the cost of changing states. We have used the Manhattan distance to define the effort to move between states, so this is a natural analog to accessibility between states. In a DCA problem there are no states that are inaccessible.

A $(\epsilon - \kappa)$ solution protocol is an instantiation of the GAP, as described above.

5.4.5 Related work

There has been some work in the DisCSP/DisCOP community that has tackled some of the issues we have presented.

Work in meeting scheduling has particular relevance to DCA. In the meeting scheduling problem a set of agents are trying to agree on when to schedule a meeting. Each agent has a private calendar that indicates their meetings. Through communication with other agents about their availability the agents come to an agreement on a time and place that is acceptable to all agents [Wallace and Freuder, 2005].

What makes the meeting scheduling problem interesting to us is the notion of privacy – agents do not want to share too much information about their schedule. [Wallace and Freuder, 2005] characterized the tradeoff between privacy and efficiency in solving the meeting scheduling problem. We can view communication limitations as a form of privacy for the agents. While relevant, distributed meeting scheduling problems usually do not capture bounded effort which is a critical component of a DCA problem.

The *Private Incremental MAP* model of [Modi and Veloso, 2005] is another model of privacy, however in this case agents preserve privacy by limiting communication of variable values to agents who are participating in a constraint with the variable.

5.5 Methods for solving a DCA problem

Due to the bounded communication and effort aspects of the DCA there are a restricted set of methods that can be applied. Protocols can differ on two axes:

- Which variable values to send.
- Picking a target state.

The activation set can have a significant impact on dynamics (as mentioned in Chapter 3), we leave a full exploration of that concept to future work. We assume that a uniformly random subset of agents are chosen as active every timestep.

While DisCOP/DisCSP algorithms cannot be "ported" directly, we can take many of the insights from them and apply them to DCA. In terms of DisCOP/DisCSP algorithms DCA only allows *Iterative Improvement* algorithms.

In this section we describe a few of the ways in which agents can decide which variable to communicate and how to change state. We intend to thoroughly explore the space of protocols in our future work (See Section 6.2).

5.5.1 Which variables to communicate

In Section 5.4.3 we described the random protocol which chooses a random set of variables to send. Clearly there are many other ways to choose a set of variables to communicate.

In CSPs there has been a significant amount of work on determining heuristics for variable ordering in order to minimize backtracking [Tsang, 1993]. These heuristics, such as minimum-width ordering (MWO), minimal bandwidth ordering (MBO), and maximum cardinality ordering (MCO) all work by exploiting the structure of the constraint graph and ordering variables according to how many constraints they take part in [Tsang, 1993]. The basic idea behind these heuristics is to identify and assign values to the variables that are more constrained first, since the values of these variables will have a large impact on the values of the other variables.

Agents in the DCA problem can exploit these heuristics by choosing to communicate variables based on this static ordering. Since these are static orderings (fixed based on the constraint graph), agents will still need to decide between the different variables – this could be done somewhat randomly where variables that are higher in the ordering are chosen more frequently. The goal would be for all agents to agree on the value of variables that are heavily constrained first, then move on to variables that are not as heavily constrained.

5.5.2 Picking a target state

Once again, we can bring to bear work from CSPs on this issue. The basic idea is to intelligently choose the target state based on the intrinsic reward of the state and the bound on effort. Value ordering heuristics from CSPs, such as the min-conflict heuristic ([Tsang, 1993]) can be used to determine which of the potential states are to be chosen. The min-conflict heuristic orders values for a variable based on the number of conflicts that value has with the values of other variables.

Similar to the min-conflict heuristic, an agent could pick its next state based on the itrinsic reward, i.e. the cost of the state. In Section 5.6 we describe an algorithm that only moves to a state that does not decrease its current intrinsic value.

5.6 Modeling phonetic change using the DCA

While it is clear that language does change, it is still unclear how and why languages change. There are numerous factors that impact how languages change, including cognitive, societal and physical factors. Because of the interaction between these many factors computer simulations have played an increasingly important role in modeling how languages change.

In this section we study the process of chain shifts – changes in the pronunciation of vowels that occur in a specific pattern, where one vowel moves creating space for another vowel to move. Chain shifts are interesting because they occur often in language and have a distintive pattern.

A chain shift is the result of interplay between two processes in language users:

- Increase communicability by aligning pronunciation with others;
- Maintain acoustic difference between the pronunciations of different vowels.

We can model these two processes in a DCA problem. Increasing communicability is the drive towards agreement; and maintaining acoustic differences between pronunciations of vowels will form the constraints.

Our interest in studying chain shifts is two fold:

- To shed light on how a particular chain shift, the *Great English Vowel Shift* took place;
- As a case study of agreement in a complex state space with bounded effort and communication.

5.6.1 Chain shifts and the Great English Vowel Shift (GEVS)

The Great English Vowel Shift (GEVS) took place between the middle of the fifteenth century to the end of the seventeenth century. It was a change in the pronunciation of certain vowels; "the systematic raising and fronting of the long, stressed monopthongs of Middle English" [Lerer, 2007]. For example, the pronunciation of the word "child" went from [čild] ("*cheeld*") in Middle English to [čəɪld] ("*choild*") to [čaɪld] ("*cha-ild*") in Present Day English.

The GEVS is often seen as an example of a *chain shift* – a situation where one vowel changes pronunciation, thus "creating space" for another vowel to "move up". This causes a chained shift for a set of vowels – starting from a change by one vowel [Hock and Joseph, 1996].

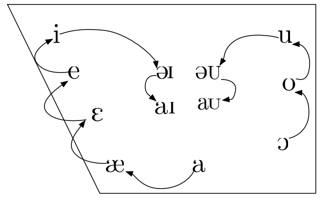


Figure 5.6: The space of vowels, and the shift that occurred during the GEVS.

The pronunciation of a vowel can be quantified by finding the *formants* of the sound. A formant is a high amplitude frequency of a sound. Different vowels have different formants. The space of pronunciations can be viewed as a two dimensional space of the frequencies of the first two formants of the sound. For instance, for the vowel [i], the first three formants are 280, 2250 and 2890 [Tserdanelis and Wong, 2004].

Figure 5.6 is a graphical depiction of the GEVS. The trapezoid is an abstract representation of the vowel space – the space of possible pronunciations of a vowel. The axes represent the first and second formants of the associated sounds.The symbols represent vowels (in the standard *International Phonetic Alphabet* notation). The arrows between vowels indicate how the vowels shifted during the GEVS. Note that the topology of the vowel shift is linear, and thus we can model it in a linear array.

There are numerous theories on why the GEVS took place. Some argue for the incorporation of numerous French loanwords into the English language [Diensberg, 1998], while others argue that the north and south of England had two different shifts due to social distancing.

One especially interesting theory is that the GEVS occured after massive migration due to the Black Death. This caused populations with highly different pronunciations of vowels to come into contact with each other [Lerer, 2007, Leith, 1997, Perkins, 1977]. We call this the "migration theory".

Many linguists argue that the reason vowel systems are organized the way they are is that they maximize acoustic distinctiveness. That is, vowels must "spread out" in pronunciation space in order to be unambiguously identified. We call this the "phonetic differentiation constraint". It has been found that vowel systems in human languages are often optimized for phonetic differentiation [Schwartz et al., 1997].

The idea behind the migration theory is a combination of two processes. First, language users in contact with others will align pronunciations, by imitation, in order to communicate better. This causes vowels to "move" in the vowel space. Because of this movement, the pronunciation of some vowels will be too similar, that is their phonetic differentiation is not large enough – this can cause vowel merger or the spreading of vowels.

Our goal is to model how vowels can change and shift due to the above two processes. Our preliminary analysis will set the stage for further exploration of this phenomenon. In general, many linguistic situations can be modeled as coming to consensus on a set of constrained variables. We can capture the interaction of the two processes through a DCA problem – in fact the DCA was developed with such goals in mind. The current work is a simple example that serves to illustrate concepts of agreement in complex spaces with bounded effort and limitations.

This work was published as [Lakkaraju et al., 2009].

5.6.2 Previous work on modeling sound change

As described in Chapter 2 in recent years there has been a huge burst of activity in modeling language change. While a large part of the work has been focused on the emergence of a shared lexicon, there has been some work in modeling vowel systems.

Vowel systems of human languages have certain patterns, one of the hypotheses is that a vowel system maximizes acoustic difference (what we call phonetic differentiation) [Schwartz et al., 1997]. While one can find these patterns, it is unclear how the vowel system might have become this way. Recently there has been work on computational model of simple acoustic agents – agents with articulatory synthesizers as well as acoustic models – interacting. The goal is to see if through interaction human language like vowel systems will appear.

[de Boer, 2000, de Boer, 1999] investigates the self organization of vowel systems through repeated language games (see Section 2.7). This work, however, focuses on the emergence of a vowel system, and not on how pronunciation could change upon interaction with other languages.

[Ettlinger, 2007] describes an exemplar based model of vowel change. Language sounds are characterized by a cloud of remembered "tokens" – as new tokens are perceived the cloud changes location in pronunciation space and thus the center of this cloud moves. [Ettlinger, 2007] showed how chain shifts can occur as agents exchange tokens. However, this work focuses on only two agents and two vowels.

5.6.3 Language model

Our focus in this paper is on vowel shifts, and thus we will create a simple phonological model of language – that is we are not modeling higher level features of language such as morphology (the creation of words), syntax, or semantics. Our model of language only deals with vowels and how they are pronounced.

A language in our system consists of some finite number of vowels. For the purposes of these experiments we fix the number of vowels in a language to 5, although we believe the results should be applicable for languages with more vowels.

A vowel has an integer value between $0 \ldots (q-1)$. This corresponds, abstractly, to some measure of the first formant of the vowel. Due to the low resolution of human perception the first two formants chiefly disambiguate a vowel [de Boer, 2000]; for simplicity we focus on a single formant model, following Ettlinger [Ettlinger, 2007]. Since the topology of a chain shift is linear, this simplifying assumption is reasonable.

The state space of the system is of the multiple features (each vowel is a feature, we use the terms interchangeably in the rest of the section), dependent type. The vowels influence each other through two constraints:

Phonetic Differentiation Constraint The vowels (in a vowel system) should have pronunciations that are different enough to allow reasonable differentiation between them. Intuitively, if two vowels are very similar they will be mistaken for each other. To increase communicability vowels should be different from each other.

Ordering Constraint There is a total ordering on the vowels. This means, e.g., that two vowels cannot swap their locations.

More formally we can model this via the DCA framework:

- $A = \{a_0, a_1, \ldots a_{n-1}\}$ are a set of n agents.
- $\mathcal{X} = \{v_0, v_1, v_2, v_3, v_4\}$ be a set of 5 variables where each variable represents a vowel.
- $\mathcal{D} = \{0, 1, \ldots (q-1)\}$. Each variable can take a value from $\mathcal{D}$ which represents the frequency of the first formant of the vowel. In this model we look at a set of discrete frequencies labeled 1 through 30.
- $f : D \times D \rightarrow \{0, 1\}$. f models the phonetic differentiation constraint. See below.

f is parametrized by d $(0 < d < q)$ which implements the phonetic differentiation constraint. We define f as:

$$f(x,y) = \begin{cases} 0 & \text{if } |x-y| - 1 \geq d \\ 1 & \text{otherwise} \end{cases} \tag{5.4}$$

$f(x,y) = 0$ when two vowels cannot be differentiated: their pronunciations are too similar. The definitions of the cost functions are as defined above.

Agents have the capability to change their vowels based on interaction with others. This models people interacting with each other and learning new words/phrases and even changing their pronunciation to increase communicability with each other. We simulate interactions between agents via *language games* [Wittgenstein, 1953, Steels, 1996] – interactions in which agents exchange knowledge of their language with each other. Since we are concerned only with vowels here, we abstract an interaction as follows.

A language game consists of two agents, a speaker (a_s) and a hearer (a_h). One of the five vowels is chosen as the *topic* of the game. The speaker then communicates to the hearer the value of this vowel. The hearer changes the value of its own vowel based on the hearers value of the vowel, and based on the constraints below. Note that this seemingly unrealistic interaction is an abstraction of a more natural interaction, where an agent utters a word that contains the vowel which we are calling the topic above. Further, it is generally possible to guess the word from context, even if the vowel pronunciations of the two agents disagree. Thus, we assume that the hearer knows both which vowel the speaker intended to utter, and which one (according to the hearer's vowel system) he actually uttered. The hearer then changes his own vowel system based on this information and the constraints below.

This interaction can be viewed in terms of the DCA as a (ϵ-1) solution protocol. That is, agents are limited to only communicating the value of one of their vowels.

In this work we focus on interactions where agents want to change – that is, whether intentionally or unintentionally hearer agents modify their vowels. In real interactions there are a whole host of social forces that could inhibit change – such as wanting to maintain a separate social identity. While this is an extremely important question, we do not attempt to address those issues in this simple model.

In terms of the GAP the dynamics of the system are:

At every time step a language game is played:

Active Agent Choice A uniformly random agent is chosen as the *speaker*, a_s.

Information Gathering Interaction Set Choice One neighbor of the speaker is chosen as the *hearer*, a_h.

Interaction a_s plays an IGI with a_h as described above. A random variable v_t is chosen as the *topic* of the game. Let $x_{s,t}$ and $x_{h,t}$ be the value of v_t for the speaker and hearer respectively.

Information Use The hearer runs the iterative update algorithm updateVariable$(v_t, x_{s,t}, \epsilon)$ described below.

Note that because of the ordering constraint this is true: $\forall_i \; x_i < x_{i+1}$.

While $\epsilon$is an important and interesting parameter to modify we leave it to future work to explore the impact of varying bounds on effort. We set effort to a large amount so that it does not affect the system dynamics.

The algorithm by which the hearer agent modifies its vowels is presented in Algorithm 1. The essential idea is that the hearer only modifies its vowel if the modification does not violate the phonetic differentiation constraint. For instance, suppose a_h has value $x_{h,t} = 5$ and $x_{h,t+1} = 10$ with $d = 4$ and plays a language game with a_s who has value $x_{s,t} = 6$. The value of v_t would not change for a_h, as it would violate the phonetic differentiation constraint. In addition, if there is not enough effort the change does not occur as well.

The algorithm presented here is at one extreme of the continuum between valuing intrinsic reward versus frequency dependent reward. In this algorithm the hearer agent only changes state if the new state does not reduce the reward the agent is currently receiving.

Algorithm 1 Hearer update algorithm.

procedure UPDATEVARIABLE($v_t, x_{s,t}, \epsilon$)
 $dir \leftarrow \text{sign}\,(x_{s,t} - x_{h,t})$
 $z \leftarrow t + dir$
 $x'_{h,t} \leftarrow$ CHANGEVARIABLE($v_t, x_{s,t}, \epsilon, dir$)
 if $|x'_{h,t} - x_{h,z}| - 1 \geq d$ **then**
  $x_{h,t} \leftarrow x'_{h,t}$
 end if
end procedure
function CHANGEVARIABLE(v_t, to, ϵ, dir)
 return $min\{to, x_{h,t+dir} - dir, x_{h,t} + dir \cdot \epsilon\}$
end function

Figure 5.7 is an illustration of transforming the vowel space into the DCA framework.

5.6.4 Number of minimal states

The optimal solution is for all agents to converge upon a minimal state, this increases the communicability since the vowels will be phonetically differentiated. A clear indication of the complexity of this is the number of minimal states which depends upon the values of $n, q,$ and d. Let $N(n, q, d)$ be the number of minimal states, then

$$N(n, q, d) = \binom{n+z}{n} \tag{5.5}$$

where $z = q - (n-1)(d+1) - 1$.

We derive $N(n, q, d)$ by first counting the amount of space taken by the *left justified minimally compact state*. Imagine the variables positioned on a 1-D array labeled from 1 to q.Then the left justified minimally compact state is where all the variables are as close as possible to the left side of the domain. The extra space to the right can be allocated to each variable without violating the phonetic differentiation constraint. $z = q - (n-1)(d+1) - 1$ is the amount of remaining space, and $\binom{n+z}{n}$ is the number of ways to distribute

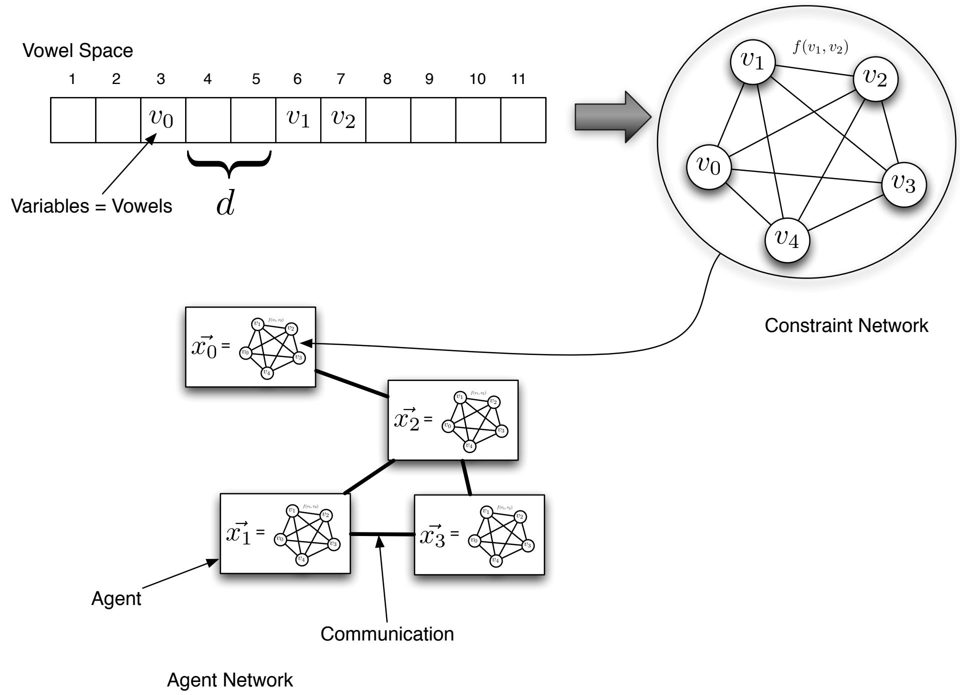


Figure 5.7: From vowel space to a DCA problem.

that space over n variables. Figure 5.8 shows an example of a left justified minimally compact state and two minimal states.

Figure 5.9 shows the number of minimal states for $q = 30$, $n = 2 \ldots 5$, $d = 1 \ldots 10$. For all values of n there is an exponential decline as d increases. The rate of decline increases as n increases, because more variables have to "fit into" the vowel space.

5.7 Simulation results

We want to study how a chain shift can occur in a population. To do this, we will empirically evaluate two situations:

1. From an initially random condition, where the agents have randomly chosen assignments of vowel positions (that respect the ordering constraint but not the phonetic differentiation constraint), we run the simulation to show that they can attain consensus on a fully solved configuration.

2. From a fully solved configuration, where all agents have the same state that minimizes the local cost, we modify a subset of the agents to have a slightly different, but minimal, state. This models sudden immigration of a new population. Does this cause a chain shift in the pronunciation of vowels?

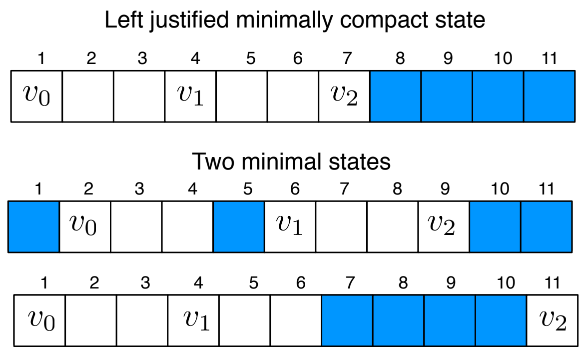


Figure 5.8: Example construction of minimal states with $n = 3$, $q = 11$ and $d = 2$. The blue squares are "extra space".

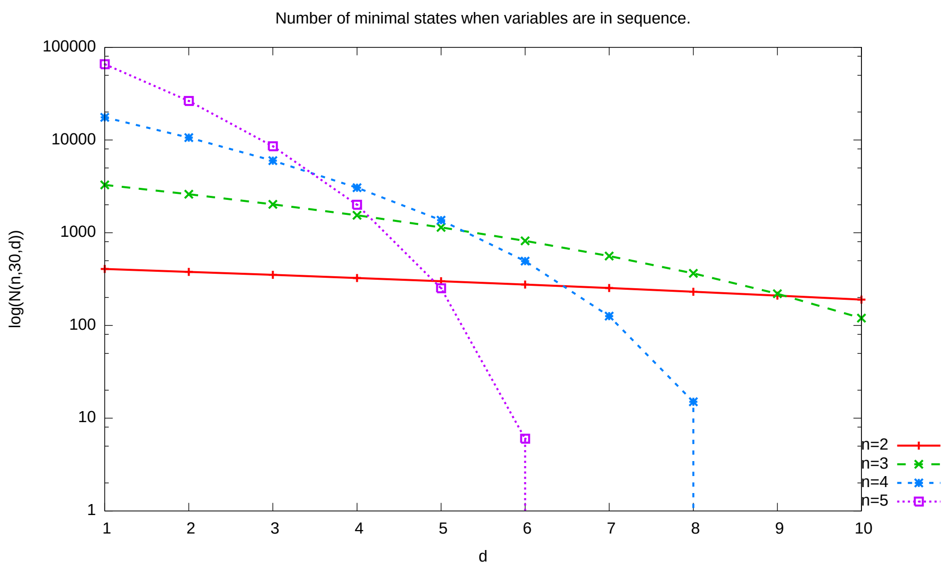


Figure 5.9: Log-Linear plot of the number of minimal states for $q = 30$ and $n = 2 \ldots 5$ and $d = 1 \ldots 10$.

Simulation parameters

The simulations below had the following settings: $n = 1000$, $m = 5$,$q = 30$, $d = 4$. To simulate social hierarchies we array the agents on a scale-free graph. We use scale-free graphs because they have been shown to model many other real-world phenomena, such as actor-collaboration graph [Barabasi and Albert, 1999]. We used the extended Barabasi-Albert scale free network generation process [Albert and Barabàsi, 2000] with parameters $m_0 = 4, m = 2, p = q = 0.4$.

5.7.1 Consensus to a fully solved configuration

In the first simulation, we initialize the population randomly, as described above, and follow algorithm 1 for the hearer in each language game. Figure 5.10 shows time on the x-axis, and the average value of each vowel

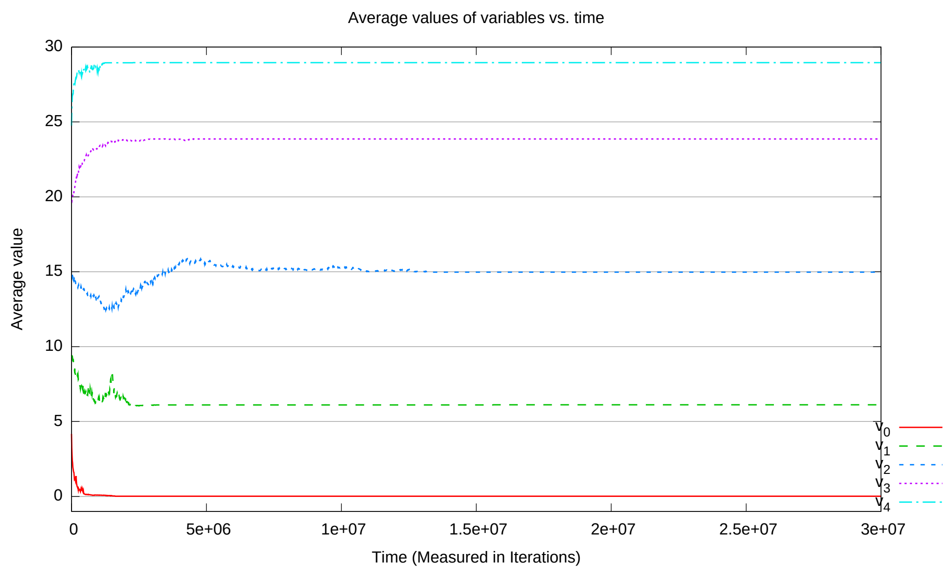


Figure 5.10: Emergence of a consensus fully-solved configuration.

on the y-axis. We see that the lines corresponding to the vowels become completely flat as the simulation progresses, and then stay that way. This demonstrates the emergence of a stable state. Further, the vowel positions are widely separated, which shows that the cost function is being minimized.

5.7.2 A new subpopulation

What happens when a new population of individuals is introduced? In this experiment we replaced 30% of the population with a new population with a slightly different state. Initially, all agents had state $[0, 5, 10, 15, 20]$, which for simplicity we call state A. The introduced population had state $[5, 10, 15, 20, 25]$, which we call state F which is an overlapping but different minimal state. The entire vowel system is shifted by five positions with respect to the existing vowel system.

The new population replaced the lowest degree 30% of nodes in the graph. The new population is introduced on iteration 2000 of the simulation. On each iteration the language game described above was executed by first picking a random agent as a speaker and one of the agents neighbors as a hearer.

Figure 5.11 shows the average values of the vowel over the entire population for a single run of the experiment. Notice the jump in average value of the variables at timestep 2000 due to the incorporation of the new population.

Interactions between the new population and existing agents immediately start to cause a shift in the vowel positions. However, the vowels do not start shifting all together – the first vowels to shift are v_0 and v_4, which move in opposite directions. These are followed in turn by v_3 and v_2 (in the direction of v_4), while

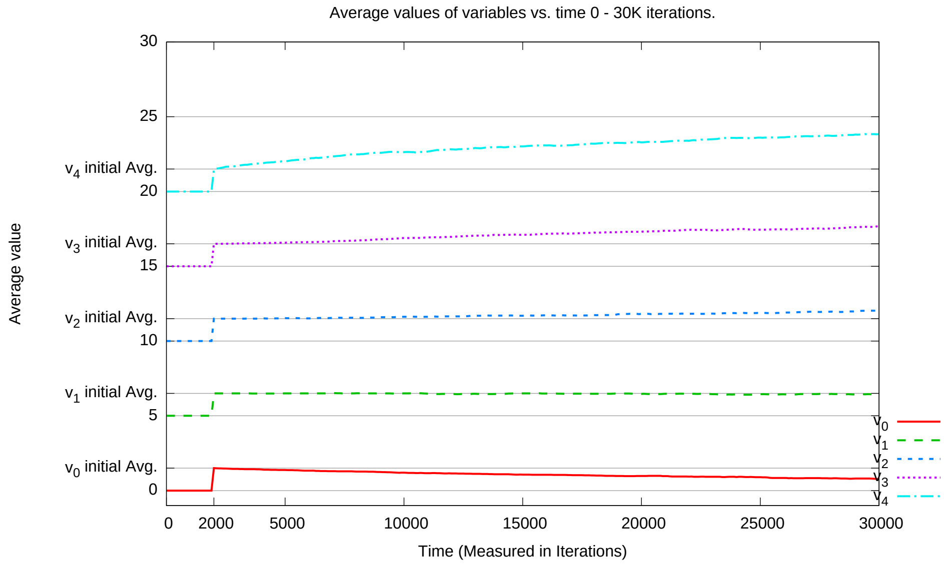


Figure 5.11: Initiation of the vowel shift.

v_1 ends up staying more or less stable. Figure 5.12 shows the long run behavior of the same experiment. We see that eventually the entire population converges on a new stable state $[0, 5, 15, 20, 25]$, which is a combination of the vowel systems of the two populations. Further, the emergence of this new vowel system occurs through a chain shift, in two directions – v_0 moves down, while $v_4, v_3,$ andv_2 move up, in that order.

Figure 5.13 shows the convergence curve for the system – what fraction of the population is in the same state. We can see that very quickly the entire population arrives to the same state. Agreement takes only $10,000,000$ iterations.

Figures 5.15 and 5.14 show one standard deviation of the variable values for the first $30,000$ iterations and the long term behavior. We can see that the standard deviation drops to 0 very quickly.

5.8 Discussion

As discussed earlier, the vowel space in figure 5.6 can be "unfolded" into a linear array, in which the a → æ→ ɛ→ e → i → əɪ→ aɪ shift is a movement to the left, and the ɔ → o → u → əʊ→ aʊ shift is a movement to the right. This matches, qualitatively, the movements observed in the experiments above, where some of the vowel positions shift up (to the left), and some shift down (to the right).

These results indicate that a chain shift could have occured due to the sudden influx of a linguistically different population.

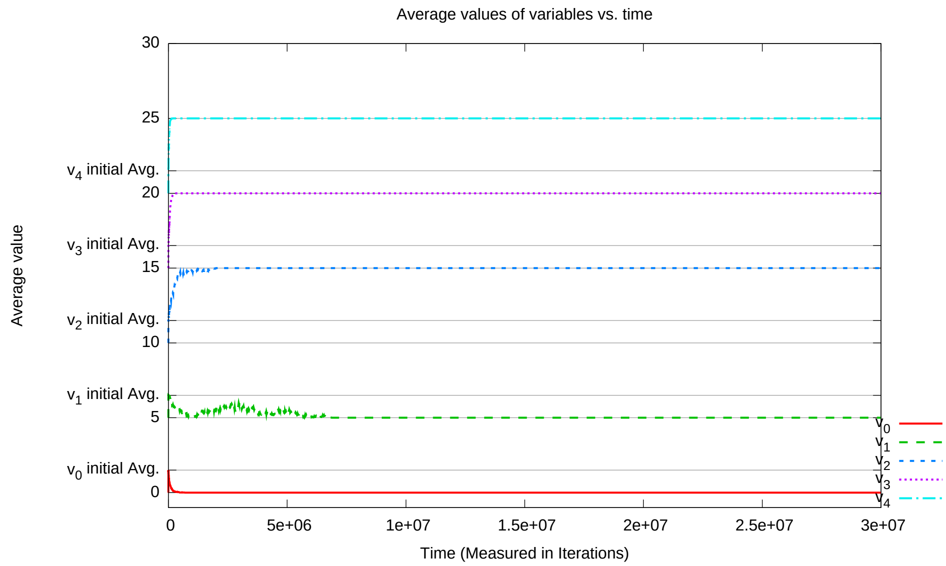


Figure 5.12: Coordination to a new state with a new introduced population.

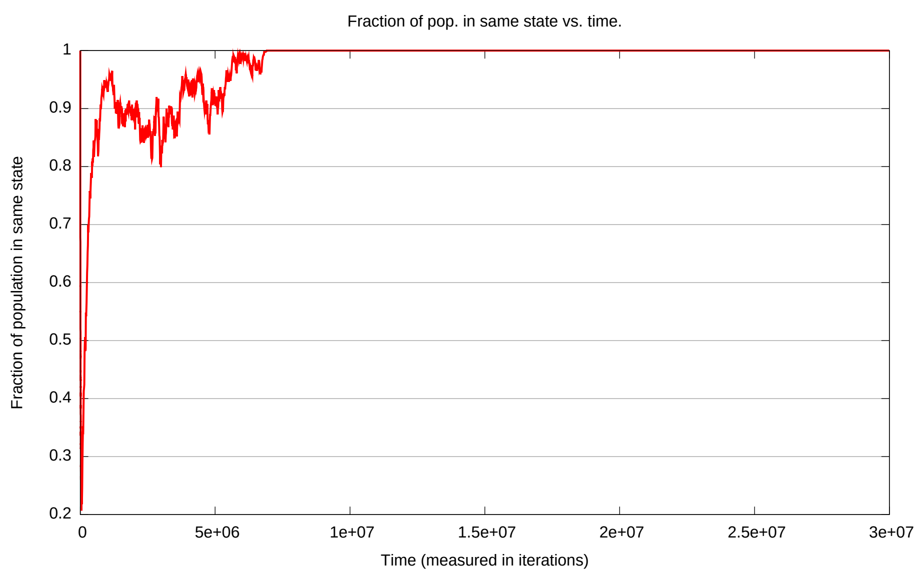


Figure 5.13: Convergence Curve

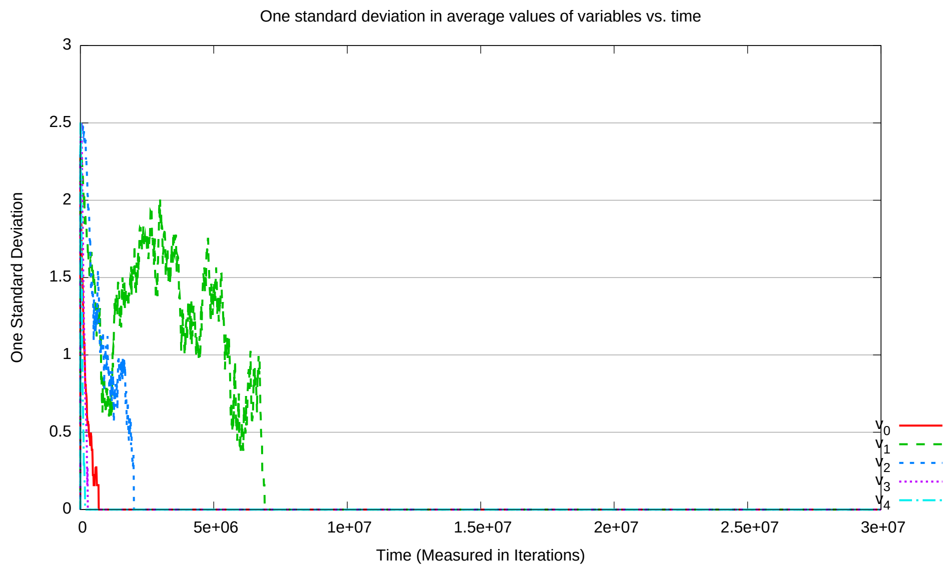


Figure 5.14: One standard deviation of the variable for a single run.

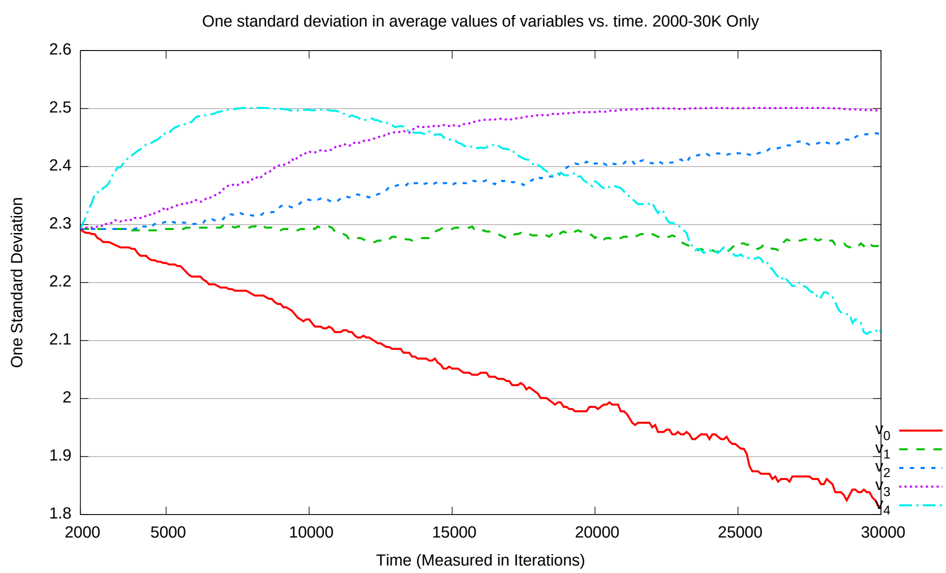


Figure 5.15: One standard deviation of the variable for a single run – first 30,000 iterations.

5.9 Conclusion

In this chapter we have explored the information use process. First we developed an idealized schemata that captures some of the key differentiators between different information use processes. One of the important issues is that of bounded communication (which impacts the information an agent receives) and bounded effort (which impacts the ability of an agent to change state).

In order to capture agreement problems in more realistic settings it is necessary to capture the concepts of a large state space with multiple features that could be dependent upon each other. We introduced the vector state space to capture these elements.

A Distributed Constraint Agreement (DCA) problem is a way of capturing agreement problems with several variables, multiple values and constraints between variables. A (ϵ-κ) protocol is a version of the Generalized Agreement Process that captures bounded communication and bounded effort.

Finally, we captured some key aspects of the linguistic phenomena of vowel chain shifting via the DCA problem. Through extensive numerical simulations we provided support for the migration theory of the Great English Vowel Shift.

5.10 Appendix: Preliminary experiments on changing the introduced subpopulation size

The work described in the section is preliminary (and thus not part of the main argument above). In this case we are trying to identify how subpopulation size affects the agreement state, and also how the topology of the system affects agreement.

In the experiment settings we set $\epsilon = 10$, effectively providing no bounds on effort, since every variable differs only by a maximum of 5 between the two populations. Because of this, we can identify all possible outcome states. Table 5.2 lists all the possible outcome states.

Label	v_0	v_1	v_2	v_3	v_4
A	0	5	10	15	20
B	0	5	10	15	25
C	0	5	10	20	25
D	0	5	15	20	25
E	0	10	15	20	25
F	5	10	15	20	25

Table 5.2: All possible states agents can take in Experiment 2.

State A is the state of the initial population and F is the state of the introduced population.

Given the potential number of states, it is somewhat surprising that in the cases above only a single state was inevitably chosen (for the parameters above, 10 out of 10 runs settled on state D).

Even more interestingly, if we were to vary the fraction of agents using the new state, we see a difference in the agreement state. Table 5.3 shows how the state that is agreed upon changes as the size of the introduced population increases.

	Primary		*Secondary*		
Fraction	State	Perc.	State	Perc	Not Converged
0.1	B	77% (7)	C	23% (2)	1
0.2	C	80% (8)	D	20% (2)	0
0.3	D	100% (10)	–	–	0
0.4	E	88% (8)	D	12% (1)	1
0.5	E	80% (8)	F	20% (2)	0
0.6	F	85% (6)	E	15% (1)	3
0.7	F	77% (7)	E	23% (2)	1
0.8	F	88% (8)	E	12% (1)	1
0.9	F	80% (8)	E	20% (2)	0

Table 5.3: Final agreement states for different sizes of introduced populations. The lowest degree agents were replaced on a scale free topology. For each experiment each run settled on one of only two states; or else the run did not converge. The Primary column denotes the state that was converged to in the majority of runs; the secondary column denotes the other state that was converged to. The numbers in parenthesis are the number of runs that settled on that state. The not converged column indicates how many runs did not converge.

	Primary		*Secondary*		
Fraction	State	Perc.	State	Perc	Not Converged
0.1	B	70% (7)	A	30% (3)	0
0.2	B	90% (9)	C	10% (1)	0
0.3	C	70% (7)	B	30% (3)	0
0.4	C	100% (10)	–	–	0
0.5	D	60% (6)	C	40% (4)	0
0.6	D	80% (8)	E	20% (2)	0
0.7	E	50% (5)	D	50% (5)	0
0.8	E	80% (8)	F	10% (1)[3]	0
0.9	E	60% (6)	E	40% (4)	0

Table 5.4: Final agreement states for different sizes of introduced populations. A randomly chosen subset of the agents were replaced with agents using state F. For each experiment each run settled on one of only two states; or else the run did not converge. The Primary column denotes the state that was converged to in the majority of runs; the secondary column denotes the other state that was converged to. The numbers in parenthesis are the number of runs that settled on that state. The not converged column indicates how many runs did not converge.

We can see from the table that as the fraction of agents in state F increase, the impact on the finally agreed upon state increases as well.

Consider the complete graph case. We can view the situation as a competition between A's and F's. A's are pulling the vowels of the population down, while F's are pulling the vowels of the population up. When

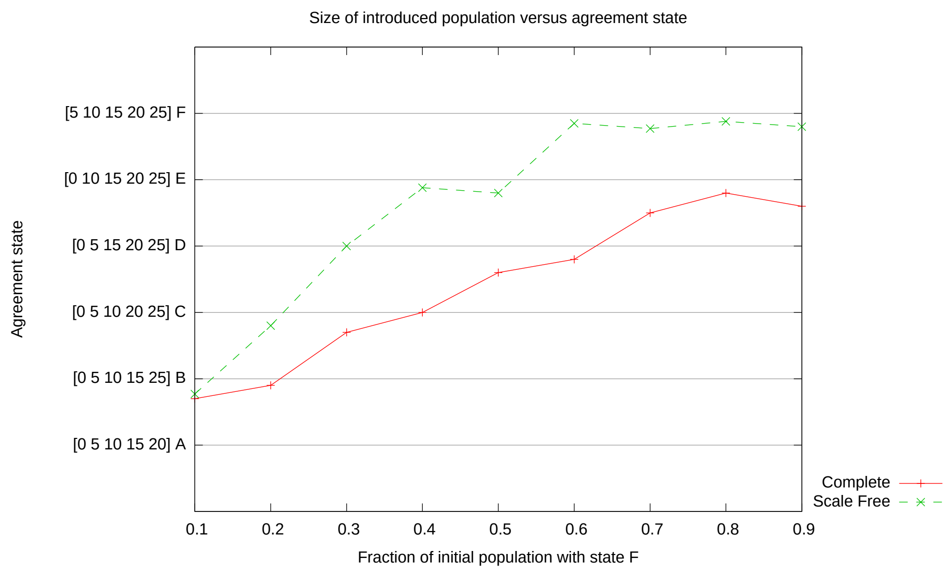


Figure 5.16: Graph of the agreement states versus fraction of population set to state F for Complete and Scale-Free graphs. The range between two states on the y-axis represents the fraction of runs that converged on the state at the top of the range. For instance, a point halfway between B and C indicates that half the runs converged on state C.

the rates of picking pairs of A's and F's are the same, that is the complete graph case with the fraction being introduced $= 50\%$, the two forces are approximately equal and thus we get the middle state D, where 2 vowels go up, and 2 vowels go down and the middle variable switches between 10 and 15.

As we increase the fraction of F's, the influence of agents with state F increases, as they are more likely to be chosen as speakers. Conversely, when there is a larger population of agents with A, the final state is biased towards A.

The same dynamics should hold for the scale free case. However, Figure 5.16 shows significant differences between the converged state on the complete graph versus the converged state on the scale free graph. We do not yet know why this occurs, but we hypothesize that it occurs because agents with state F are chosen as speakers with higher frequency that agents with state A. This could be due to the fact that agents with state F replace the lowest degree 30% of populations. Further work is necessary to understand this process.

Chapter 6

Conclusion

Agreement problems arise in a large number of domains, spanning from models of basic physical processes (i.e. Ising models), to models of complex linguistic processing. Our overall research aim is to develop a general theory of agreement that is applicable to many domains. To achieve this research aim we must understand the fundamental processes that underlie agreement problems. The goal of this dissertation has been to begin to develop an understanding of these basic processes so that we may start to formulate a general theory of agreement.

The thesis of this dissertation is that the ability for agents to optimally allocate resources towards 1) gaining information from which to infer the agreeing population's global agreement state ("information gathering") and 2) effectively using that information to make convergence decisions that move towards agreement ("information use"), are the fundamental factors that explain the performance of a distributed agreement-seeking collective, and that variations on these processes capture all prevalent styles of agreement problems.

Before we summarize our contributions, we lay out a few general messages from this work.

1. Agreement problems from a wide variety of disciplines are fundamentally similar, and there is much to be learned by studying agreement problems from a variety of disciplines.

2. When developing an agreement protocol we must consider the cost of communication. Communication is often very expensive and it must be limited. While some areas (such as distributed systems) have studied the message complexity of algorithms, most work on agreement protocols have not. In Chapter 4, via the ICCC, we explored the interaction between communication cost and time to agreement.

3. For better computational models of agreement problems in linguistic domains we must develop techniques that handle large and complex agreement spaces where there are significant limitations in the accuracy of information gathered. The DCA was developed for this purpose, to capture these three elements and also the issue of bounded effort and communication.

We had four goals for this disseration. We discuss each below.

6.1 Summary of contributions

6.1.1 Organizing agreement problems

In Chapter 3 we developed the *Distributed Optimal Agreement* (DOA) taxanomic framework as a way of organizing agreement problems under a common conceptual framework. We described the Generalized Agreement Process, which we argue captures the three fundamental processes common to all agreement problems, that is:

Agent Activation Some agents must change – we call these the active agents.

Information Gathering The active agents must gather information from other agents.

Information Use Agents must decide how to change their state.

By varying which and how many agents are active; what information can be gathered; and how agents use this information we can model many different agreement problems. The DOA framework provided a formal model of these processes by modeling agreement as distributed search through a state space. We showed how information gathering is modulated by the interaction relation that determines the cost for agents to communicate. Information use is modulated by the accessibility relation – the cost for agents to move between states, and the intrinsic and frequency dependent reward functions.

The DOA framework laid the foundation for further, in depth exploration of the information gathering and information use processes in the subsequent chapters.

This work was published in [Lakkaraju and Gasser, 2007]

6.1.2 Language as an agreement problem

We described agreement in language and argued that to tackle linguistic agreement problems we need to capture these three aspects:

Large Agreement Space The number of possible languages to agree upon is extremely large.

Complex Agreement Space Elements of language (i.e. grammar, lexicon etc) interact with and constrain each other.

Incomplete Information Agents do not get complete information about the languages of others.

In chapter 5 we explored agreement in phonological spaces which captures all three of these aspects – a large and complex agreement space (with 5 vowels and the phonetic differentiation constraint between them) and incomplete information through bounded communication (the linguistic games only uncovered the value of one of the variables of an agent).

This work was published in [Lakkaraju and Gasser, 2008b].

6.1.3 Exploring information gathering

The aim of chapter 4 was to investigate, in detail, the process of Information Gathering which is one of the three key components of the Generalized Agreement Process. We articulated the *Fundamental Agreement Tradeoff* which captured the interaction between effort, information and agreement time. Investigation of this tradeoff is a critical part to developing a general theory of agreement.

To greater understand this tradeoff we need to understand how the elements interact with each other; that is, to execute an interaction requires some resource expenditure, interactions impact information, and information impacts agreement.

To explore the fundamental agreement tradeoff we developed the Sampled Majority Vote(SMV) model which allowed us to directly modulate the amount of information an agent could gather in a binary state, static complex graph setting via the *sampling fraction* parameter.

We showed that the number of IGIs an agent executes has a direct impact on the accuracy of an agents information. When information gathering was limited there was a significant increase in the incidence of Mistaken Majority (MM) errors. We calculated the probability of a mistaken majority error to occur based on the sampling fraction and distribution of states among an agents neighbors. Through extensive numerical simulations we showed the impact of MM errors on agreement time.

Finally, we developed the Information Centric Convergence Cost Metric that quantifies the fundamental agreement tradeoff in terms of the cost of effort (both communication and cognition) and the cost of not being in agreement (missed opportunity cost). Using the ICCC we found that the optimal value for the sampling fraction was surprisingly low for a wide range of cost ratios.

This work was published in [Lakkaraju and Gasser, 2009b].

6.1.4 Exploring information use

In Chapter 5 we focused on the process by which agents decides on their target state – the state to which an agent will change. We outlined an idealized schemata that describes the process by which an agent gradually identifies a target state by taking into account the accessibility relation, limitations on effort, and the information gathered from the previous stage.

We developed the Distributed Constraint Agreement (DCA) model as a way to explore agreement problems in complex state spaces (where there are multiple, interacting features and significant restrictions on communication and effort). DCA is similar to DisCSP/DisCOP, but differs significantly in fundamental assumptions of agent behavior.

For the twin purposes of exploring an instance of the DCA model and shedding light on an interesting linguistic phenomena we used the DCA to model the Great English Vowel Shift – an instance of a *chain shift*. We showed that a simple iterative improvement algorithm under significant communication constraints – only 1 variable was communicated – can result in a "chain shift" in a population of agents. To our knowledge this is the first computational model of chain shifting in large populations of agents.

This work was published in [Lakkaraju et al., 2009].

6.2 Ongoing and future work

There are many ways in which this work can be extended.

Formal Analysis of Sampled Majority Vote A formal analysis of the relationship between sampling fraction and agreement time would provide great insight into the impact of information on agreement time. In particular, we intend to combine the analysis from the majority-rule model [Chen and Redner, 2005b, Chen and Redner, 2005a, Krapivsky and Redner, 2003] on complete graphs with work on voter models on degree-heterogenous graphs, [Sood et al., 2008, Sood and Redner, 2005]. There are significant difficulties to be overcome due to the fact that we are utilizing the discrete hypergeometric distribution. However, there is quite a bit of work on approximations to the tail of the hypergeometric distribution that can be useful in simplifyng the analysis, [Skala, 2009, Lahiri et al., 2007, Pinsky, , Chvatal, 1979].

Exploration of the Small-World Graphs in the SMV Currently our empirical evaluations of the SMV protocol are restricted to complete and scale-free graphs. It would be interesting to see how modulating the sampling fraction would affect agreement time on different types of graphs, especially small world graphs, which capture properties of many natural and technical networks [Strogatz, 2001]. However, preliminary investigation of SMV on small-world graphs has shown that time to agreement varies tremendously from run to run (in the range of millions of iterations difference). A similar situation was reported for the GSM protocol [Delgado, 2002, Delgado,] which is an instantiation of the SMV protocol with $\theta = 1.0$. We do not know why there is such variance in agreement time, but we intend to investigate this phenomena.

Phase transitions in DCA For CSPs and DisCSPs it is known that a phase transition exists in terms of problem difficulty; when there are few constraints or when there are many constraints finding a solution is easy; however there is a "mushy" region inbetween where finding a solution is very difficult [Yokoo, 2001, Prosser, 1996, Meisels, 2008]. It is unknown whether this region exists for DCA as well, and how it is impacted by communication and effort limitations.

More realistic phonological space Our model of chain shifting makes some simplifying assumptions; such as a one dimensional phonological space and an extremely simple hearer update algorithm that does not include the concept of vowel merging, or the introduction of new vowels. One direction of future research is to develop a more realistic model of the vowel space – the work of [de Boer, 2000] will be extremely applicable as they have developed a more realistic model of phonological change that incorporates articulatory and acoustic aspects.

Appendix A

A.1 Albert-Barabàsi Extended Model

We use the Extended Barabasi-Albert algorithm to generate the scale free graph. The benefit of this algorithm is that it allows some tuning of the parameter v – although we will suffer some finite-scale effects.

The Albert-Barabàsi extended model depends upon four parameters; m_0 is the initial number of nodes, $m(\leq m_0)$ is the number of links that are added or rewired every step of the algorithm, p is the probability of adding links, and q is the probability of rewiring an edge ($p + q = 1$). The algorithm to generate the network is as follows. Start with m_0 isolated nodes, and at each step perform one of these three actions:

1. With probability p add m new links. Choose the start of the link uniformly randomly and the end point with distribution:
$$\Pi_i = \frac{k_i + 1}{\sum_j (k_j + 1)} \tag{A.1}$$
where Π_i is the probability of selecting the ith node and k_i is the number of edges of node i. This process is repeated until m new links are added to the graph. If m links cannot be added we add as many as possible.

2. With probability q rewire m edges. Pick uniformly randomly a node i and link l_{ij} between node i and node j. Delete this link and choose another node k according to the probability distribution Π_i with the constraints that $k \neq i, j$ and l_{ik} does not already exist. Add the link l_{ik}.

3. With probability $1 - p - q$ add a new node with m links – the new links with connect the new node to m other nodes chosen according to Π_i.

References

[Albert and Barabàsi, 2000] Albert, R. and Barabàsi, A.-L. (2000). Topology of evolving networks: Local events and universality. *Physical Review Letters*, 85(24):5234–5237.

[Axelrod, 1997] Axelrod, R. (1997). The dissemination of culture: A model with local convergence and global polarization. *The Journal of Conflict Resolution*, 41(2).

[Barabasi and Albert, 1999] Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509.

[Barbosa, 1996] Barbosa, V. C. (1996). *An Introduction to Distributed Algorithms.* MIT Press.

[Baron et al.,] Baron, D. E., Bailey, R. W., and Kaplan, J. P. Brief for professors of linguistics and english dennis e. baron, ph.d,. richard w. bailey, ph.d. and jeffrey p. kaplan, ph.d. in support of petitioner. Amicus Brief.

[Baronchelli et al., 2005] Baronchelli, A., Dall'Asta, L., Barrat, A., and Loreto, V. (2005). Topology induced coarsening in language games. *Physical Review E*, 73:015102.

[Baronchelli et al., 2008] Baronchelli, A., Loreto, V., and Steels, L. (2008). In-depth analysis of the naming game dynamics: The homogenous mixing case. *International Journal of Modern Physics C.*

[Baronchelli et al., 2006] Baronchelli, A., Loretto, V., Dall'Asta, L., and Barrat, A. (2006). Bootstrapping communication in language games: Strategy, topology and all that. In *Proceedings of the 6th International Conference on the Evolution of Language.*

[Beule, 2006] Beule, J. D. (2006). Simulating the syntax and semantics of linguistic constructions about time. In Gontier, N., van Bendegem, J. P., and Aerts, D., editors, *Evolutionary Epistemology, Language and Culture - A non-adaptationist, systems theoretical approach*, Theory and Decision Library - Series A: Philosophy and Methodology of the Social Sciences. Springer, Dordrecht.

[Blondel et al., 2005] Blondel, V. D., Hendricx, J. M., Olshevsky, A., and Tsitsiklis, J. N. (2005). Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference.* IEEE.

[Blumrosen and Feldman, 2006] Blumrosen, L. and Feldman, M. (2006). Implementation with a bounded action space. In *Proceedings of the 7th ACM conference on Electronic Commerce*, pages 62–71.

[Blumrosen et al., 2007] Blumrosen, L., Nisan, N., and Segal, I. (2007). Auctions with severely bounded communication. *Journal of Artificial Intelligence Research*, 28:233–266.

[Campos et al., 2003] Campos, P. R. A., de Oliveira, V. M., and Moreira, F. G. B. (2003). Small-world effects in the majority-vote model. *Physical Review E*, 67(026104).

[Castellano et al., 2007] Castellano, C., Fortunato, S., and Loreto, V. (2007). Statistical physics of social dynamics. ArXiv.

[Castellano and Pastor-Satorras, 2006] Castellano, C. and Pastor-Satorras, R. (2006). Non-mean-field behavior of the contact process on scale-free networks. *Physical Review Letters*, 96.

[Chen and Redner, 2005a] Chen, P. and Redner, S. (2005a). Consensus formation in multi-state majority and plurality models. *Journal of Physics A: Mathematical and General*, 38.

[Chen and Redner, 2005b] Chen, P. and Redner, S. (2005b). Majority rule dynamics in finite dimensions. *Physical Review E*, 71.

[Chvatal, 1979] Chvatal, V. (1979). The tail of the hypergeometric distribution. *Discrete Mathematics*, 25:285–287.

[Conlisk, 1996] Conlisk, J. (1996). Why bounded rationality? *Journal of Economic Literature*, XXXIV(2):669–700.

[Cornforth et al., 2005] Cornforth, D., Green, D. G., and Newth, D. (2005). Ordered asynchronous processes in multi-agent systems. *Physica D*, 204:70–82.

[Coulouris et al., 2005] Coulouris, G., Dollimore, J., and Kindberg, T. (2005). *Distributed Systems: Concepts and Design*. Addison Wesley, fourth edition.

[Cucker et al., 2004] Cucker, F., Smale, S., and Zhou, D.-X. (2004). Modeling language evolution. *Foundations of Computational Mathematics*, 4(3):315–343.

[da F. Costa et al., 2007] da F. Costa, L., Rodrigues, F. A., Travieso, G., and Boas, P. R. V. (2007). Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242.

[Dall'Asta and Baronchelli, 2006] Dall'Asta, L. and Baronchelli, A. (2006). Microscopic activity patterns in the naming game. *Journal of Physics A: Mathematical and General*, 39:14851–14867.

[Dall'Asta et al., 2006a] Dall'Asta, L., Baronchelli, A., Barrat, A., and Loreto, V. (2006a). Agreement dynamics on small-world networks. *Europhysics Letters*, 73:969–975.

[Dall'Asta et al., 2006b] Dall'Asta, L., Baronchelli, A., Barrat, A., and Loreto, V. (2006b). Non-equilibrium dynamics of langauge games on complex networks. *Physical Review E*, 74:036105.

[de Boer, 1999] de Boer, B. (1999). *Self-organisation in Vowel Systems*. PhD thesis, Vrije Universiteit Brussel.

[de Boer, 2000] de Boer, B. (2000). Self-organization in vowel systems. *Journal of Phonetics*, 28:441–465.

[de Oliveira, 1992] de Oliveira, M. J. (1992). Isotropic majority-vote model on a square lattice. *Journal of Statistical Physics*, 66(1/2):273.

[Delgado,] Delgado, J. Personal communication. email.

[Delgado, 2002] Delgado, J. (2002). Emergence of social conventions in complex networks. *Artificial Intelligence*, 141(1-2):171–185.

[Delgado et al., 2003] Delgado, J., Pujol, J. M., and Sanguesa, R. (2003). Emergence of coordination in scale-free networks. *Web Intelligence and Agent Systems*, 1(2).

[Demers et al., 1987] Demers, A. J., Greene, D. H., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H. E., Swinehart, D. C., and Terry, D. B. (1987). Epidemic algorithms for replicated database maintenance. In *PODC*.

[Diensberg, 1998] Diensberg, B. (1998). Linguistic change in english: The case of the great vowel shift from the perspective of phonological alternations due to the wholesale borrowing of anglo-french loanwords. *Folia-Linguistica-Historica*, 19(1-2):103–117.

[Ettlinger, 2007] Ettlinger, M. (2007). An exemplar-based model of chain shifts. In *Proceedings of the 16th International Congress of the Phonetic Science*, pages 685–688.

[Fax and Murray, 2004] Fax, J. A. and Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465.

[Gibson, 2005] Gibson, D. R. (2005). Concurrency and commitment: Network scheduling and its consequences for diffusion. *Journal of Mathematical Sociology*, 29(4):295–323.

[Gibson and Wexler, 1994] Gibson, E. and Wexler, K. (1994). Triggers. *Linguistic Inquiry*, 25(3):407–454. i do have the pdf for this now.. kl 08/12/2009.

[Giordano, 1997] Giordano, N. J. (1997). *Computational Physics*. Prentice Hall.

[Gmytrasiewicz, 2002] Gmytrasiewicz, P. J. (2002). Negotiation as a mechanism for language evolution. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy.

[Gmytrasiewicz et al., 2002] Gmytrasiewicz, P. J., Summers, M., and Gopal, D. (2002). Toward automated evolution of agent communication languages. In *Proceedings of the 35th Hawaii International Conference on System Sciences, HICSS-35*.

[Goldstone and Janssen, 2005] Goldstone, R. L. and Janssen, M. (2005). Computational models of collective behavior. *Trends in Cognitive Science*, 9(9).

[Habiba et al., 2008] Habiba, Yu, Y., Berger-Wolf, T. Y., and Saia, J. (2008). Finding spread blockers in dynamics networks. In *Proceedings of the 2nd SNA-KDD Workshop*.

[Hock and Joseph, 1996] Hock, H. H. and Joseph, B. D. (1996). *Language History, Language Change, and Language Relationship: An Introduction to Historical and Comparitive Linguistics*. Mouton de Gruyter.

[Johnson and Kotz, 1969] Johnson, N. L. and Kotz, S. (1969). *Distributions in Statistics: Discrete Distributions*. Series in Statistics. Houghton Mifflin Company.

[Kempe et al., 2006] Kempe, D., Kleinberg, J., and Demers, A. (2006). Spatial gossip and resource location protocols. In *Proceedings of STOC'06*.

[Kempe et al., 2003] Kempe, D., Kleinberg, J., and Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of SIGKDD*.

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*.

[Kermarrec et al., 2003] Kermarrec, A.-M., Massouliè, L., and Ganesh, A. J. (2003). Probabilistic reliable dissemination in large-scale systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(2).

[Kleinbergn, 2007] Kleinbergn, J. (2007). Cascading behavior in networks: Algorithmic and economic issues. In an dT. Roughgarden, N. N., Tardos, E., and Vazirani, V., editors, *Algorithmic Game Theory*. Cambridge University Press.

[Komarova, 2004] Komarova, N. L. (2004). Replicator-mutator equation, universality property and population dynamics of learning. *Journal of Theoretical Biology*, 230(2):227–239.

[Komarova et al., 2001] Komarova, N. L., Niyogi, P., and Nowak, M. A. (2001). The evolutionary dynamics of grammar acquisition. *Journal of Theoretical Biology*, 209(1):43–59.

[Komarova and Nowak, 2001] Komarova, N. L. and Nowak, M. A. (2001). The evolutionary dynamics of the lexical matrix. *Bulletin of Mathematical Biology*, 63(3):451–485.

[Koren, 2005] Koren, J. (2005). Minimizing communication cost in an n-agent distributed bayesian network by using a decentralized mdp. Master's thesis, Southern Illinois University.

[Kossinets et al., 2008] Kossinets, G., Kleinberg, J., and Watts, D. (2008). The structure of information pathways in a social communication network. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[Krapivsky and Redner, 2003] Krapivsky, P. L. and Redner, S. (2003). Dynamics of majority rule in two-state interacting spin particle system. *Physical Review Letters*, 90(23).

[Lahiri et al., 2007] Lahiri, S., Chatterjee, A., and Maiti, T. (2007). Normal approximation to the hypergeometric distribution in nonstandard cases and a sub-gaussian berry-esseen theorem. *Journal of Statistical Planning and Inference*, 137:3570–3590.

[Lakkaraju and Gasser, 2007] Lakkaraju, K. and Gasser, L. (2007). A unified framework for multi-agent agreement (short paper). In *Proceedings of AAMAS '07*, Honolulu, Hawaii.

[Lakkaraju and Gasser, 2008a] Lakkaraju, K. and Gasser, L. (2008a). Norm emergence in complex ambiguous situations. In *Proceedings of the Workshop on Coordination, Organizations, Institutions and Norms at AAAI*.

[Lakkaraju and Gasser, 2008b] Lakkaraju, K. and Gasser, L. (2008b). Three issues in modeling the language convergence problem as a multiagent agreement problem. In *In Proceedings of EvoLang 2008*.

[Lakkaraju and Gasser, 2009a] Lakkaraju, K. and Gasser, L. (2009a). Collective learning: Principled information strategies for improving collective learning: Principled information strategies for improving performance in norm emergence and multi-agent agreement. In *Submitted to IJCAI 2009*.

[Lakkaraju and Gasser, 2009b] Lakkaraju, K. and Gasser, L. (2009b). Minimizing information-centric convergence cost in multi-agent agreement problems. In *Proceedings of the Adaptive Learning Agents Workshop at AAMAS 2009*.

[Lakkaraju et al., 2009] Lakkaraju, K., Swarup, S., and Gasser, L. (2009). Consensus under constraints: Modeling the great english vowel shift. In *In Proceedings of the Social Simulation Workshop at the International Joint Conference on Artificial Intelligence (IJCAI)*.

[Laubenbacher et al., 2008] Laubenbacher, R., Jarrah, A. S., Mortveit, H., and Ravi, S. S. (2008). A mathematical formalism for agent-based modeling. In *Encyclopedia of Complexity and System Science*. Springer Verlag.

[Leith, 1997] Leith, D. (1997). *A Social History of English*. Routledge, 2nd edition.

[Lerer, 2007] Lerer, S. (2007). *Inventing English: A Portable History of the Language*. Columbia University Press.

[Lewis, 1969] Lewis, D. K. (1969). *Convention*. Harvard University Press.

[Lieberman et al., 2005] Lieberman, E., Hauert, C., and Nowak, M. A. (2005). Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316. Supplementary notes in: lieberman05evolutionary-supp.pdf.

[Lima et al., 2008] Lima, F., Sousa, A., and Sumour, M. (2008). Majority-vote on directed erdos-renyi random graphs. *Physica A*, 387:3503–3510.

[Lorenz, 2007] Lorenz, J. (2007). Continuous opinion dynamics under bounded confidence: A survey. *International Journal of Modern Physics C*, 18(12):1819–1838.

[Lynch, 1997] Lynch, N. A. (1997). *Distributed Algorithms*. Morgan Kaufmann.

[Mason et al., 2007] Mason, W. A., Conrey, F. R., and Smith, E. R. (2007). Situating social influence processes: Dynamic, multidirectional flows of influence within social networks. *Personality and Social Psychology Review*, 11.

[Matsen and Nowak, 2004] Matsen, F. A. and Nowak, M. A. (2004). Win-stay, lose-shift in language learning from peers. *PNAS*, 101(52):18053–18057.

[Meisels, 2008] Meisels, A. (2008). *Distributed Search by Constrained Agents: Algorithms, Performance, Communication*. Advanced Information and Knowledge Processing. Springer-Verlag.

[Mobilia and Redner, 2003] Mobilia, M. and Redner, S. (2003). Majority versus minority dynamics: Phase transition in an interacting two-state spin system. *Physical Review E*, 68.

[Modi and Veloso, 2005] Modi, P. J. and Veloso, M. (2005). Bumping strategies for the multiagent agreement problem. In *Proceedings of Autonomous Agents and Multi-Agent Systems, (AAMAS)*.

[Moore and Newman, 2000] Moore, C. and Newman, M. E. J. (2000). Epidemics and percolation in small-world networks. *Physical Review E*, 61(5).

[Morris, 2000] Morris, S. (2000). Contagion. *Review of Economic Studies*, 67(1):57–78.

[Newman and Watts, 1999] Newman, M. E. J. and Watts, D. J. (1999). Scaling and percolation in the small-world network model. *Physical Review E*, 60:7332–7342.

[Nowak et al., 2000] Nowak, M. A., Plotkin, J. B., and Jansen, V. A. A. (2000). The evolution of syntactic communication. *Nature*, 404:495–498.

[Olfati-Saber and Murray, 2004] Olfati-Saber, R. and Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533.

[Olfati-Saber and Murray, 2003] Olfati-Saber, R. and Murray, R. M. (2003). Consensus protocols for networks of dynamic agents. In *Proceedings of the 2003 American Control Conference*.

[Oliphant and Batali, 1997] Oliphant, M. and Batali, J. (1997). Learning and the emergence of coordinated communication. *The newsletter of the Center for Research in Language*, 11(1).

[Oudeyer, 1999] Oudeyer, P.-Y. (1999). Self-organization of a lexicon in a structured society of agents. To Appear.

[Pastor-Satorras and Vespignani, 2001a] Pastor-Satorras, R. and Vespignani, A. (2001a). Epidemic dynamics and endemic states in complex networks. *Physical Review E*, 63(066117).

[Pastor-Satorras and Vespignani, 2001b] Pastor-Satorras, R. and Vespignani, A. (2001b). Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14).

[Patel et al., 2006] Patel, J. A., Gupta, I., and Contractor, N. (2006). Jetstream: Achieving predictable gossip dissemination by leveraging social network principles. In *Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)*. IEEE Computer Society.

[Pereira and Moreira, 2005] Pereira, L. F. C. and Moreira, F. G. B. (2005). Majority-vote model on random graphs. *Physical Review E*, 71(016123).

[Perkins, 1977] Perkins, J. (1977). A sociolinguistic glance at the great vowel shift of english. *Papers in Psycholinguistics and Sociolinguistics. Working Papers in Linguistics*, (22).

[Pinsky,] Pinsky, M. A. The normal approximation to the hypergeometric distribution. Note on webpage.

[Pottie and Kaiser, 2000] Pottie, G. and Kaiser, W. (2000). Wireless integrated networked sensors. *Communications of the ACM*, 43(5).

[Prosser, 1996] Prosser, P. (1996). An empirical study of phase transitions in binary constraint satisfaction problems. *Artificial Intelligence*, 81:81–109.

[Pujol et al., 2005] Pujol, J., Delgado, J., Sangüesa, R., and Flache, A. (2005). The role of clustering on the emergence of efficient social conventions. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence IJCAI-05*.

[Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligencce: A Modern Approach*. Prentice Hall, 2nd edition.

[Saber and Murray, 2003] Saber, R. O. and Murray, R. M. (2003). Agreement problems in networks with directed graphs and switching toplogy. In *Proceedings of the 2003 Conference on Decision and Control (CDC)*.

[Sawyer, 2005] Sawyer, R. K. (2005). *Social Emergence: Societies As Complex Systems*. Cambridge University Press.

[Schwartz et al., 1997] Schwartz, J.-L., Boe, L.-J., Vallee, N., and Abrey, C. (1997). The dispersion-focalization theory of vowel systems. *Journal of Phonetics*, 25:255–286.

[Shannon et al., 2003] Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: A software environment for integrated models of biolmolecular interaction networks. *Genome Research*, 13:2498–2504.

[Shen et al., 2006] Shen, J., Becker, R., and Lesser, V. (2006). Agent interaction in distributed pomdps and its implications on complexity. In *Proceedings of AAMAS '06*.

[Shen et al., 2002] Shen, J., Lesser, V., and Carver, N. (2002). Controlling information exchange in distributed bayesian networks. Technical report, University of Massachusetts at Amherst.

[Shen et al., 2003] Shen, J., Lesser, V., and Carver, N. (2003). Minimizing communication cost in a distributed bayesian network using a decentralized mdp. In *Proceedings of AAMAS '03*.

[Shoham and Tennenholtz, 1997] Shoham, Y. and Tennenholtz, M. (1997). On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166.

[Shultz and Lepper, 1996] Shultz, T. R. and Lepper, M. R. (1996). Cognitive dissonance reduction as constraint satisfaction. *Psychological Review*, 103(2):219–240.

[Skala, 2009] Skala, M. (2009). Hypergeometric tail inequalities: ending the insanity. Online.

[Smith, 2004] Smith, K. (2004). The evolution of vocabulary. *Journal of Theoretical Biology*, 228(1):127–142.

[Sood, 2007] Sood, V. (2007). *Interacting Particle Systems on Graphs*. PhD thesis, Boston University.

[Sood et al., 2008] Sood, V., Antal, T., and Redner, S. (2008). Voter models on heterogenous networks. *Physical Review E*, 77.

[Sood and Redner, 2005] Sood, V. and Redner, S. (2005). Voter models on heterogenous graphs. *Physical Review Letters*, 94. 178701.

[Steels, 1996] Steels, L. (1996). Emergent adaptive lexicons. In *Proceedings of the Fourth International Conference on Simulating Adaptive Behavior*.

[Steels, 1998] Steels, L. (1998). The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194.

[Steels, 2003] Steels, L. (2003). The evolution of communication systems by adaptive agents. In Alonso, E., D. K. and Kazakov, D., editors, *Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning. LNAI 2636*, pages 125–140. Springer Verlag, Berlin.

[Steels, 2005a] Steels, L. (2005a). The emergence and evolution of linguistic structure: from lexical to grammatical communication systems. *Connection Science*, 17:213–230.

[Steels, 2005b] Steels, L. (2005b). The role of construction grammar in fluid language grounding. *Submitted*.

[Steels and Vogt, 1997] Steels, L. and Vogt, P. (1997). Grounding adaptive language games in robotic agents. In Harvey, I. and Husbands, P., editors, *ECAL97*, Cambridge, MA. MIT Press.

[Steels and Wellens, 2006] Steels, L. and Wellens, P. (2006). How grammar emerges to dampen combinatorial search in parsing. In et al., P. V., editor, *Symbol Grounding and Beyond: Proceedings of the Third International Workshop on the Emergence and Evolution of Linguistic Communication*, pages 76–88. Springer.

[Strang and Soule, 1998] Strang, D. and Soule, S. A. (1998). Diffusion in organizations and social movements: From hybrid corn to poison pills. *Annual Review of Sociology*, 24:265–90.

[Strogatz, 2001] Strogatz, S. H. (2001). Exploring complex networks. *Nature*, 410(6825):268–276.

[Suchecki et al., 2004] Suchecki, K., Eguiluz, V. M., and Miguel, M. S. (2004). Conservation laws for the voter model in complex networks. ArXiv.

[Suchecki et al., 2008] Suchecki, K., Eguiluz, V. M., and Miguel, M. S. (2008). Voter model dynamics in complex networks: Role of dimensionality, disorder and degree distribution. *ArXiv*. cond-mat/0504482v1.

[Sundaram and Hadjicostis, 2008a] Sundaram, S. and Hadjicostis, C. (2008a). Distributed calculation of linear functions in noisy networks via linear iterations. In *Proceedings of the 47th Conference on Decision and Control*.

[Sundaram and Hadjicostis, 2008b] Sundaram, S. and Hadjicostis, C. (2008b). Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications: Issue on Control and Communications*, 26(4).

[Swarup, 2007] Swarup, S. (2007). *Artificial Language Evolution on a Dynamical Interaction Network*. PhD thesis, The University of Illinois at Urbana-Champaign.

[Swarup et al., 2006] Swarup, S., Lakkaraju, K., and Gasser, L. (2006). Learning a common language through an emergent interaction topology. In *Proceedings of AAMAS '06*.

[Thagard and Verbeurgt, 1998] Thagard, P. and Verbeurgt, K. (1998). Coherence as constraint satisfaction. *Cognitive Science*, 1:1–24.

[Tsang, 1993] Tsang, E. (1993). *Foundations of Constraint Satisfaction*. Computation in Cognitive Science. Academic Press.

[Tserdanelis and Wong, 2004] Tserdanelis, G. and Wong, W. Y. P., editors (2004). *Language files: materials for an introduction to language and linguistics*. The Ohio State University Press, ninth edition.

[Vylder, 2007] Vylder, B. D. (2007). *The Evolution of Conventions in Multi-Agent Systems*. PhD thesis, Artificial Intelligence Lab, Vrije Universiteit Brussel.

[Vylder and Tuyls, 2006] Vylder, B. D. and Tuyls, K. (2006). How to reach linguistic consensus: A proof of convergence for the naming game. *Journal of Theoretical Biology*.

[Wagner et al., 2003] Wagner, K., Reggia, J., Uriagereka, J., and Wilkinson, G. (2003). Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11:37–69.

[Walker and Wooldridge, 1995] Walker, A. and Wooldridge, M. J. (1995). Understanding the emergence of conventions in multi-agent systems. In *ICMAS95*, pages 384–389, San Francisco, CA.

[Wallace and Freuder, 2005] Wallace, R. J. and Freuder, E. C. (2005). Constraint-based reasoning and privacy/efficiency treadoffs in multi-agent problem solving. *Artificial Intelligence*, 161:209–227.

[Watts, 2002] Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(9):5766–5771.

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

[Williams et al., 2007] Williams, J. L., III, J. W. F., and Wilsky, A. S. (2007). Approximate dynamic programming for communication-constrained sensor network management. *IEEE Transactions on Signal Processing*, 55(8).

[Wilson and Keil, 1999] Wilson, R. A. and Keil, F. C., editors (1999). *The MIT Encyclopedia of Cognitive Science*. The MIT Press.

[Wittgenstein, 1953] Wittgenstein, L. (1953). *Philosophical Investigations*. Blackwell Publishing.

[Xiao and Boyd, 2004] Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53:65–78.

[Yokoo, 2001] Yokoo, M. (2001). *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-Agent Systems*. Springer Series on Agent Technology. Springer Verlag, Berlin.

[Yokoo and Hirayama, 2000] Yokoo, M. and Hirayama, K. (2000). Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi-Agent Systems*, 3:185–207.

Author's Biography

Kiran Lakkaraju was born on September 2nd 1979 in Melrose Park, Illinois. Apart from a few summers in India and one summer in Florham Park, New Jersy, he has spent his entire life in the cornfields and suburbs of Illinois.

He earned a Bachelors of Science degree in Computer Science from Northern Illinois University in DeKalb in May of 2000. Subsequently, Kiran straightaway migrated south to the University of Illinois at Urbana-Champaign where he received a Masters of Science in May of 2002, and then a doctorate in December of 2009, both in Computer Science.

During his time at the University of Illinois, Kiran has had the pleasure of working with the National Center for Supercomputing Applications at the University of Illinois. Kiran has more than 25 publications spanning areas as diverse as network security visualization, network log anonymization, language evolution, emergence of norms and conventions, and cumulative learning.

After finishing his doctoral work Kiran will finally leave Illinois to join Sandia National Laboratories in Albuquerque, New Mexico as a senior member of technical staff.