12-22-2010

Strategic and Secure Interactions in Networks

Jinsong Tan
jinsong@seas.upenn.edu

STRATEGIC AND SECURE INTERACTIONS IN NETWORKS

Jinsong Tan

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2010

Supervisor of Dissertation

Michael Kearns, Professor, Computer and Information Science

Graduate Group Chairperson

Jianbo Shi, Associate Professor, Computer and Information Science

Dissertation Committee:

Sanjeev Khanna, Professor, Computer and Information Science, University of Pennsylvania

Ali Jadbabaie, Associate Professor, Electrical and Systems Engineering, University of Pennsylvania

Ben Taskar, Assistant Professor, Computer and Information Science, University of Pennsylvania

Samuel Ieong, Microsoft Research

STRATEGIC AND SECURE INTERACTIONS IN NETWORKS

Acknowledgements

First and foremost, I would like to thank my dissertation advisor Michael Kearns for introducing me to a fascinating new area of research. I vividly remember how my very first few research meetings with him were teeming with such creativity from him, that it brought me refreshing new perspectives and inspired me to pursuit research in this area. Working with him throughout the years has changed the way I think about and approach research. I am very fortunate to have him as my advisor, colleague, and friend.

I would like to thank my dissertation committee, Sanjeev Khanna, Ali Jadbabaie, Ben Taskar, and Sam Ieong. Their helpful advice and suggestions helped guide my dissertation work to completion. I would also like to thank Mike Felker, the graduate coordinator of our department, for all the administrative help throughout my PhD years at Penn.

I would like to thank my coauthors, Tanmoy Chakraborty, J. Stephen Judd and Jenn Wortman Vaughan, with whom I collaborated on works that lead to this dissertation. It has been a pleasure working with them and I have learned a great deal from them in our collaborations.

My thanks to my friends at Penn as well, who are only too many to be listed here in perfect completion, for making my life at Penn wonderful.

I am also grateful to those who have advised and mentored me academically from outside of Penn: To Hon-Wai Leong, for giving me the very first opportunity in doing academic research when I was still an undergraduate at the National University of Singapore; to Louxin Zhang, for being an early inspiration and mentor in my academic career and for the very enjoyable research we collaborated on when I worked as a research assistant for him; and to Sam Ieong (again) for the great research experience that I had when I worked with him as a summer intern at Microsoft Research.

Finally, I would like to extend my deepest gratitude to my family – mum, dad, and my wife Weile – for their love, support and encouragement over all these years. Without them this dissertation would have been impossible and I dedicate this dissertation to them.

ABSTRACT

STRATEGIC AND SECURE INTERACTIONS IN NETWORKS

Jinsong Tan

Michael Kearns

The goal of this dissertation is to understand how network plays a role in shaping certain strategic interactions, in particular biased voting and bargaining, on networks; and to understand how interactions can be made secure when they are constrained by the network topology. Our works take an interdisciplinary approach by drawing on theories and models from economics, sociology, as well as computer science, and using methodologies that include both theories and behavioral experiments.

First, we consider biased voting in networks, which models distributed collective decision making processes where individuals in a network must balance between their private biases or preferences with a collective goal of consensus. Our study of this problem is two-folded. On the theoretical side, we start by introducing a diffusion model called *biased voter model*, which is a natural extension of the classic *voter model*. Among other results, we show in the presence of biases, no matter how small, there exists certain networks where it takes exponential time to converge to a consensus through distributed interaction in networks. This is a stark and interesting contrast to the well-known result that it always takes polynomial time to converge in the voter model, when there are no biases. On the experimental side, a group human subjects were arranged in various carefully designed virtual networks to solve the biased voting problem. Along with analyses of how collective and individual performance vary with network structure and incentives generally, we find there are well-studied network topologies in which the minority preference consistently wins globally, and that the presence of "extremist" individuals, or the awareness of opposing incentives, reliably improve collective performance

Second, we consider bargaining in networks, which has long been studied by economists and sociologists. A basic premise behind the many theoretical study of bargaining in networks is that pure topological differences in agents' network positions endow them with different bargaining power. Complementary to these theories, we conduct a series of controlled behavioral experiments, where human subjects were arranged in various carefully designed virtual networks to playing bargaining games. Along with other findings of how individual and collective performance vary with network structures and individual playing styles, we find that the number of neighbors one can negotiate with confers bargaining power, whereas the limit on the number of deals one can close undermines it, and we find that competitions from distant parts of the network, though invisible locally, also play a significant and subtle role in shaping bargaining powers.

And last, we consider the question of how interactions in networks can be made secure. Traditional methods and tools from cryptography, for example secure multi-party computation, can be applied only if each party can talk to everyone else directly; but cannot be directly applied if interactions are distributed over a network without completely eradicating the distributed nature. We develop a general 'compiler' that turns each algorithm from a broad class collectively known as *message-passing algorithms* into a secure one that has exactly the same functionality and communication pattern. And we show a fundamental trade-off between preserving the distributed nature of communication and the level of security one can hope for.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Commercial aviation based international travelling can make an outbreak of flu in a distant village a worldwide pandemic in a matter of weeks, the word-of-mouth campaign run on friendship based social networks can have significant influences over the success of a new product or the outcome of an election, the interconnectedness of a country's power grid can expose it to the kind of vulnerability where the failure of a single power line can knock out power supply and wreak havoc in millions of people's lives.

All these phenomena share a common structural underpinning: Our modern society is connected in myriad and complex ways through technology, and such connectedness often comes with profound implications not yet fully understood. A particularly life altering example is the proliferation of the Internet during the past decade: The Web democratizes the creation of information and makes its dissemination faster than ever before, the popularization of social networking sites like Facebook transforms our notion of privacy, and online shopping sites such as eBay and Amazon revolutionize the way sellers interact with buyers.

These networks are often inherently technological, social and economic at the same time, therefore to fully understand them one needs to draw on theories, models and methodologies from economics, sociology, as well as computer science.

In response to the increasingly important role network plays in our modern life, one line of efforts that is generally known as *social network theory* has undergone fast development in the past decade. Social network theory has traditionally been mainly concerned with the observation and measurement of structural properties and patterns of naturally occurring networks, and in turn having the empirical understanding thus gained inform the invention of generative models that can reproduce various kind of structural properties that seem to be ubiquitous in these naturally occurring networks.

Properties as such include for example having a *giant component*, meaning the vast majority of nodes in a network are connected, either directly or through other nodes; and *small diameter*, which means the average distance between any two nodes in a given network tends to be extremely small when it is compared to the size of the network; and *scale-free distribution*, meaning that the degree distribution of the nodes follows a power law distribution. Correspondingly, generative models have been invented in social network theory to reproduce these properties: Erdös and Rényi proposed the *random graph* model [29, 30, 13], probably the simplest probabilistic model one can think of where edges between nodes are generated independently with a constant probability, that explains how giant component can suddenly emerge from a network if the edge density across certain threshold; the *small-world* model developed by Watts and Strogatz [83], which starts with a ring lattice and then rewire some of the local connection to random distant nodes, naturally generate networks with small diameters; and the *preferential attachment* model invented by Barabási and Albert [7], where new nodes are introduced to the network sequentially and randomly attached to existing nodes with probability proportional to their degrees, generates networks with a power law degree distribution.

Despite the success of these and many other generative models in reproducing many of the statistical properties and patterns observed in naturally occurring networks, they in general do not model the nodes in the networks as intelligent and strategic beings who act in their own interests, but mindless automatons who only make decisions in a purely probabilistic or mechanical fashion.

This hampers our ability in modeling and analyzing the dynamics of interactions that occur in the networks we are interested in, which are often inherently social and economic. It is thus natural for one to turn to economics, and in particular game theory, for the languages and tools to describe and analyze these networks.

Game theory is a branch of applied mathematics that attempts to model strategic interactions between rational and self-interested beings, which in game-theoretic terminology are known as *agents*. Traditional models in game theory consider situations where there are n agents, and each agent interacts directly with *everyone* else and has to take into account the actions everyone else might take when making his own in order to maximize his own benefit. Although in principle one can use models like this to describe strategic interactions in a network where only local interactions matter, it quickly becomes extremely cumbersome as the population size grows larger because it takes space exponential in n to just describe the game. To circumvent this problem, Kearns, Littman and Singh in their seminal work [54] introduced *graphical games* as a succinct language to describe strategic interactions in networks. In a graphical game, a network is imposed exogenously on the n agents, and each agent, still being fully rational and self-interested, can now only interact with his immediate neighbors in the network and only have to take into account his neighbors actions in order to optimize his own. The graphical game model not only offers tremendous efficiency in describing games in networks by reducing descriptional complexity from exponential in n to exponential in k, where k is the degree of the graph and can be extremely small compared to n, but also provides us with a fresh perspective in looking at and understanding social and economic networks.

Subsequent and related works have taken a step further to try to blend the two fields of *social network theory* and *game theory*. One line of research generally known as *network formation games* [34, 4, 2, 32, 72, 47, 31, 43, 33, 64, 65, 5, 8] sought to understand how the topology of social and economic networks could have arisen from and shaped by various kind of strategic interactions between self-interested agents. Another line takes networks as given exogenously, and focus on implications of network topology on dynamics, e.g. diffusion of information or disease (see [79] for an overview of the topic), and games, e.g. trading [50, 33, 12], bargaining [27, 15, 63, 20, 51], on networks.

The first part of this dissertation continues the line of research that study games and dynamics in networks. We consider two kinds of strategic interactions, namely *biased voting* and *bargaining* in networks, which we give a brief overview in Section 1.1.

1.1 Strategic Interactions in Networks

1.1.1 Biased Voting in Networks

Many collective decision-making or voting processes in politics, business, and other arenas must balance diverse individual preferences with a desire for collective unity. Such processes often take place in social or organizational networks, in which individuals are most influenced by, or aware of, the current views of their network neighbors. A recent, if approximate, example of this phenomenon is the 2008 Democratic National Primary race in the United States. On the one hand, individual voters held opposing and sometimes strong preferences that were apparently very nearly balanced across the population; on the other hand, there was a strong and explicit desire among them that once the winning candidate was identified, the entire party should unify behind that candidate.

We formalize such settings as a biased voting problem over an undirected graph, whose local structure models the social influences acting on individual voters. Our model contrasts with the significant literature on diffusion of opinion in social networks as it is the first to study this topic in the presence of both private bias and strong incentives toward collective unity. The theoretical analyses of our model yielded an interesting insight: No matter how infinitesimal the bias is, a broad class of stochastic opinion diffusion processes, which includes natural generalizations of the well-studied *voter model*, takes exponential time to converge to consensus on some networks. And this is in stark contrast to the well-known result that the *voter model*, which is studied in the absence of private biases, always takes polynomial time to converge to consensus on any network. On the other hand, we propose a simple and local stochastic updating protocol that provably converges to the *collectively preferred* consensus in polynomial time on any network even in the presence of biases. And we further extend this protocol to a strategic setting where individual may have incentives to deviate from the prescribed protocol, and showed the new protocol is an approximate Nash equilibrium of the resulting game.

Complementary to the theoretical work, an extensive session of behavioral experiments on biased voting in networks of individuals were conducted. In each of 81 experiments, 36 human subjects arranged in a virtual network were financially motivated to reach global consensus to one of two opposing choices. No payments were made unless the entire population reached a unanimous decision within 1 minute, but different subjects were paid more for consensus to one choice or the other, and subjects could view only the current choices of their network neighbors, thus creating tensions between private incentives and preferences, global unity, and network structure. Along with analyses of how collective and individual performance vary with network structures and incentives generally, our key findings include the following:

- There are well-studied network topologies in which the minority preference consistently wins globally;
- The presence of “extremist” individuals, or the awareness of opposing incentives, reliably improve collective performance;
- Certain behavioral characteristics of individual subjects, such as “stubbornness” and “stableness”, are strongly correlated with earnings.

We develop the theoretical work on biased voting in networks in Chapter 2 and the corresponding behavioral experiments in 3.

1.1.2 Bargaining in Networks

In Chapter 4 we consider an extensive session of behavioral experiments on *bargaining in social networks*.

Bargaining has been studied extensively in economics and sociology, and the popular setting is of two parties negotiating a single deal. The deal yields a fixed total wealth if the two parties can agree on how to split it and otherwise both parties receive nothing. Bargaining in a network is the setting where nodes represent agents, who can close a limited number of deals, and edges represent bilateral opportunities for deals between neighboring agents. There has been a long line of previous theoretical work which tried to relate wealth to network topology, and a notable feature of these theories is the prediction that there may be significant local variation in splits purely as a result of the imposed deal limits and structural asymmetries in the network. Our experiments constitute a test of human subjects' actual behaviors in this game and are among the first and largest behavioral experiments on network effects in bargaining conducted to date.

Our key findings pertaining to existing network bargaining theory include the following:

- Deals are often struck with unequal shares, though more than one-third of the deals are equally shared;
- Higher degree tends to raise bargaining power, while higher deal limits tend to decrease bargaining power; on the other hand, higher deal limits in the first neighborhood tend to raise bargaining power whereas higher degrees in the first neighborhood tend to lower it;
- Local topology affects bargains, but invisible competition in other parts of the network also affects it, even when the local topologies are indistinguishable.

Other findings that speak to no existing theory are the following:

- Social efficiency is positively correlated with inequality of the deals;
- Social efficiency is higher in the presence of uncertainty;
- Deals left unclosed due to failure to agree form the greater part of missing efficiency.
- People who are patient bargainers tend to make more money.

These findings are in need of theoretical development and they are among the future research areas I will pursue. The nature and the potential implications of these findings argue for the further need to integrate the fields of economics, game theory, sociology, psychology, and computer science.

1.2 Secure Interaction in Networks

In the second part of the dissertation, we turn our attention to security issues arising from interactions on networks.

1.2.1 Network Faithful Secure Computation

As the use of technology permeate through our modern life, sharing details of personal or other sensitive information in social networks is becoming an increasingly prevalent practice. Through an important class of highly distributed protocols on graphs known broadly as *message-passing algorithms*, which only requires direct information sharing between local neighbors in a network, each node in the network can compute sophisticated aggregated global functions of the whole population. While doing so has the benefit of allowing one to obtain useful information without directly revealing sensitive information to the entire population, it is almost always the case that one reveals information beyond what is intended during the process.

Consider, for example, a large social network in which each node represents an individual and each edge represents a relationship between individuals. Imagine that each party in this network would like to compute his or her own probability of having contracted a contagious disease, which depends on the probabilities that each of his or her neighbors in the network have been exposed to the disease, and that in turn depend on the exposure probabilities of the neighbors' neighbors, so on and so forth. This could be accomplished by running the standard *belief propagation* algorithm on the network. However, if the network participants engage in standard belief propagation, each party will learn much more than his own contraction probability. In particular, each party could potentially learn information about the contraction probabilities of their neighbors, as well as more global information (such as the fraction of the population that has been infected). Such leakage of information can be highly undesirable.

Besides belief propagation [78, 85], *message passing algorithms* comprises of a broad class of distributed algorithms or protocols that are of great interest in areas such as artificial intelligence, machine learning, statistics, and signal process. Notable examples include Gibbs sampling [18, 37], Nash propagation in graphical games [54, 76], gossip algorithms [14], survey propagation [16], constraint propagation [28], and many others. Message-passing formalisms have long been studied in distributed computing. With rising interest in large-scale, decentralized networks such as the Internet, message-passing algorithms are of increasing importance due to their localized communication and their lack of any need for non-local topological information; in most instances parties do not even need to know the overall size of the network, yet they can compute sophisticated global functions, such as joint distributions and Nash equilibria.

The growing prevalence of message-passing algorithms in computation related to social and economic interactions raises security concerns as mentioned above. To tackle this, one approach would be to simply apply classic and powerful cryptography tool called secure multi-party computation [84, 39] to the message-passing algorithms, preserving their input-output functionality while imbuing them with very strong privacy properties. Unfortunately, secure multi-party computation requires each party being able to talk directly to everyone else, which is often impossible due to the enormous size of the population and the lack of sufficient communication resources. Even if such communication resources were available, by requiring all parties to maintain and communicate distributed shares of every computation even with very "distant" parties in the network, this straightforward approach would largely eradicate the benefits of the message-passing framework in the first place.

In the second part of this dissertation, we seek to ask the question: Can we design algorithms that get the best of both worlds? On the one hand, these algorithms need to preserve the highly distributed, local communication pattern of the original message-passing algorithms. On the other hand, it comes with (at least some of) the traditional privacy assurances of secure multi-party computation. We call this *network faithful secure computation.*

We answer this question in the affirmative. In Chapter 5, we develop secure versions of belief propagation and Gibbs sampling by drawing on secure multi-party computation and other classic tools from cryptography. We then further generalize this result

In Chapter 5 we show not only belief propagation, but also *any* message-passing algorithms can be made secure. Specifically, we have the following results

- We construct a general compiler that turns *any* message-passing protocol into one computing the same functionality, but that is secure against single-party adversaries (1-privacy).

- We propose a simulation-based definition of what it means for a secure protocol to be faithful to the original network structure and protocol, and a proof that our compiler produces extremely faithful 1-private protocols.

- And last we give an impossibility result showing a trade-off between faithfulness and security against coalitions. In particular, we show that for certain functionalities, any highly faithful protocol must be vulnerable to collusion by small coalitions, thus proving the optimality of our compiler with respect to this trade-off.

1.3 Bibliographic Notes

The model, analysis and algorithms of networked biased voting in Chapter 2 are based on joint work with Michael Kearns [57]. The behavioral experiments on networked biased voting in Chapter 3 are based on joint work with Michael Kearns, Stephen Judd and Jennifer Wortman [53]. The behavioral experiments on bargaining in networks in Chapter 4 are based on joint work with Tanmoy Chakraborty, Stephen Judd, and Michael Kearns [19]. The work on network faithful secure computation in Chapter 5 is based on a joint work with Michael Kearns and Jennifer Wortman [58].

Chapter 2

The Networked Biased Voting Problem

2.1 Introduction

The tension between the expression of individual preferences and the desire for collective unity appears in decision-making and voting processes in politics, business, and many other arenas. Furthermore, such processes often take place in social or organizational networks, in which individuals are most influenced by, or aware of, the current views of their network neighbors.

The 2008 Democratic National Primary race offers a recent, if approximate, example of this phenomenon. On the one hand, individual voters held opposing and sometimes strong preferences that were apparently very nearly balanced across the population; however, there was a strong and explicit desire that once the winning candidate was identified, the entire party should unify behind that candidate [86]. Obviously primary voters could be influenced by many global factors (such as polls and mainstream media) outside the scope of their individual social and organizational networks, but presumably for many voters these local influences still played an important and perhaps even dominant role.

Although there is now a significant literature on the diffusion of opinion in social networks

[62, 42, 81], the topic is typically studied in the absence of any incentives toward collective unity. In many contagion- metaphor models, individuals are simply more or less susceptible to "catching" an opinion or fad from their neighbors, and are not directly cognizant of, or concerned with, the global state. In contrast, we are specifically interested in scenarios in which individual preferences are present but are subordinate to reaching a unanimous global consensus.

In this chapter, we formalize such scenarios as the "Networked Biased Voting Problem" (NBVP) over an undirected graph, whose local structure models the social influences acting on individual voters. In this model, each voter i is represented by a vertex in the network and a real-valued weight $w_i \in [0, 1]$ expressing their preference for one of two candidates or choices that we shall abstractly call *red* and *blue*. Here $w_i = \frac{1}{2}$ is viewed as indifference between the two colors, while $w_i = 0$ (red) and $w_i = 1$ (blue) are "extremist" preferences for one or the other color.

Our overarching goal is to investigate distributed algorithms in which three criteria are met:

1. *Convergence to the Global Preference:* If the global average W of the w_i is even slightly bounded away from $\frac{1}{2}$ (indifference), then *all* members of the population should eventually settle on the globally preferred choice (i.e. all red if $W < \frac{1}{2}$, all blue if $W > \frac{1}{2}$), even if it conflicts with their own preferences (party unity).

2. *Speed of Convergence:* Convergence should occur in time polynomial in the size of the network.

3. *Simplicity and Locality:* Voters should employ "simple" algorithms in which they communicate only *locally* in the network via (stochastic) updates to their color choices. These updates should be "natural" in that they plausibly integrate a voter's individual preferences with the current choices of their neighbors, and do not attempt to encode detailed information, send "signals" to neighbors, etc.

The first two of these criteria are obviously formally precise. While it might be possible to formalize the third as well, we choose not to do so here for the sake of brevity and exposition. However, we are explicitly *not* interested in algorithms in which (for instance) voters attempt to encode and broadcast their underlying preferences w_i as a series of binary choices, or similarly unnatural and complex schemes. In particular, in our main protocol it will be very clear that voters are always updating their current choices in a way that naturally integrates their own preferences and the statistics of current choices in their local neighborhood.

We note that the formalization above clearly omits many important features of "real" elections. Foremost among these is the fact that real elections typically have strong global coordination and communication mechanisms such as polling, while we require that all communication between participants be entirely local in the network. On the other hand, our framework does allow for the presence of high-degree individuals, including ones that are indifferent to the outcome ($w_i = \frac{1}{2}$) and can thus act as "broadcasters" of current sentiment in their neighborhood. Variation in degrees can also be viewed as a crude model for the increasing variety of global to local media sources (from "mainstream" publications to influential blogs to small discussion groups).

There is a large literature on the diffusion of opinion in social networks [42, 81, 62], but the topic is usually studied in the absence of any force towards collective unity. In many contagion-metaphor models, individuals are more or less susceptible to "catching" an opinion or fad from their neighbors, but are not directly concerned with the global outcome. In contrast, we are specifically interested in scenarios in which individual preferences are present, but are subordinate to reaching a unanimous global consensus.

Our main results are:

- An impossibility result establishing exponential convergence time for the NBVP for a broad class of local stochastic updating rules, which includes natural generalizations of the well-studied "voter model" from the diffusion literature (and which is known to converge in polynomial time in the absence of differing individual preferences).

- A new simple and local stochastic updating protocol whose convergence time is provably polynomial on any instance of the NBVP. This new protocol allows voters to declare themselves "undecided", and has a temporal structure reminiscent of periodic polling or primaries.

- An extension of the new protocol that we prove is an approximate Nash equilibrium for a game-theoretic version of the NBVP.

Chapter Outline: In the next section, we give a formal definition of the Networked Biased Voting Problem. In Section 2.3, we review the classic *voter model*, which is an extremely simple and natural opinion diffusion process and well studied in the literature. We extend this model to what we call the biased voter model in Section 2.4, which encompasses a broad class of protocols or diffusion processes, and show that no protocol from this class can solve the NBVP. By making slight relaxations to the biased voter model, we give a protocol in Section 2.5 that efficiently solves NBVP, and another protocol in Section 2.6 that approximately solves a game-theoretic version of NBVP.

2.2 The Networked Biased Voting Problem

The networked biased voting problem (NBVP) is studied over an undirected graph $G = (V, E)$ with n nodes and m edges, where each node i represents an individual voter. Denote by $\mathcal{N}(i)$ the neighbors of i in G; we always consider i as a neighbor of himself.

There are two competing choices or opinions, that without loss of generality we shall call *blue* and *red* (or b and r for short). A voter i comes with a real-valued weight $w_i \in [0, 1]$ expressing his preference for one of the two opinions; without loss of generality, let $w_i(b) = w_i$ and $w_i(r) = 1 - w_i$ denote his preference for *blue* and *red*, respectively.

Throughout, we make the assumption that one opinion is always collectively preferred to the other. More formally, we assume there exists a constant $\epsilon > 0$, which is independent of the size of G, such that

$$\left| \sum_{i \in V} w_i(b) - \sum_{i \in V} w_i(r) \right| > \epsilon.$$

We also assume that which opinion is preferred is *not* known a priori to the nodes and the goal of the networked biased voting problem is for the entire population to actually figure this out through a distributed algorithm, or protocol, that is *simple and local*, and converges in time polynomial in n to the collectively preferred consensus. Because of the stochastic nature of the protocol we consider, it is implausible to require that it always converges to the collectively preferred consensus. Instead, we require the protocol does so *with high probability*, by which we mean the probability can differ from 1 by an amount that is at most exponentially small in n. We summarize the definition of networked biased voting problem in the following.

Networked Biased Voting Problem (NBVP)

Instance: Given an undirected graph $G = (V, E)$ with n nodes, two opinions $\{b, r\}$, and for each $i \in V$, a preference $(w_i(b), w_i(r))$ where $w_i(b), w_i(r) \in [0, 1]$ and $w_i(b) + w_i(r) = 1$. Assume there exists an opinion $\alpha \in \{b, r\}$ such that $\sum_i w_i(\alpha) > \frac{n}{2} + \epsilon$ for some constant $\epsilon > 0$. α is not known a priori.

Objective: Design a *simple and local* distributed protocol that in time polynomial in n lets V converge to α with high probability.

We will consider protocols of the following form:

1. (Initialization) At round 0, each node i in V independently and simultaneously initializes to an opinion in $\alpha \in \{b, r\}$ according to $\mathcal{I}$, a randomized function that maps i's local information to an opinion in $\{b, r\}$.

2. (Stochastic Updating) At round $t \geq 1$, a node i is chosen uniformly at random from V; i then picks a neighbor $j \in \mathcal{N}(i)$ according to a possibly non-uniform distribution over

$\mathcal{N}(i)$; this distribution is determined by function $\mathcal{F}$, which is a randomized function whose arguments are i's local information. i then converts to j's opinion.

Therefore a protocol is specified by a pair of functions, $(\mathcal{I}, \mathcal{F})$. This framework by itself does not forbid "unnatural" coding behaviors as discussed in the Introduction; however in the spirit of emphasizing algorithms that are simple and local, we restrict $\mathcal{I}$ and $\mathcal{F}$ to be functions that only depend on simple and local information of a node i. In particular, only the following arguments to either functions are considered: 1) f_i, the distribution of opinions in the neighborhood, where $f_i(b)$ and $f_i(r)$ represent the fractions of neighbors currently holding opinion *blue* and *red*, respectively; 2) i's intrinsic preferences w_i; 3) i's degree $d_i = |\mathcal{N}(i)|$.

2.3 The Classic Voter Model

The voter model, which was introduced by Clifford and Sudbury [24] and Holley and Liggett [45], is a well-studied probabilistic stochastic process that models opinion diffusion on social networks in a most basic and natural way. It consists of a class of protocols that satisfy our criterion of being *simple and local*. In fact, this class of protocols is the simplest that we examine in this paper. A voter model protocol is one where in each round, a node i is picked uniformly at random from V, and i in turn picks one of his neighbors uniformly at random and adopts his opinion; it does not specify how the initialization is done. More formally,

Definition 1 (Voter Model). *The voter model is a class of protocols of the form* $(\mathcal{I}, \mathcal{F})$ *where* $\mathcal{F}(f_i) = \alpha$ *with probability* $f_i(\alpha)$, $\forall\ \alpha \in \{b, r\}$.

Importantly, the voter model is a class of protocols in which there are no individual preferences present at all, and the only concern is with reaching unanimity (to either color). This is in sharp contrast to the networked biased voting problem that we consider here, where not only individual preferences are explicitly modelled and reaching the collectively preferred opinion is desired, but any preferences are subordinate to reaching the collectively preferred consensus. Nevertheless, we shall make use of some known results on the voter model, which we turn to now.

Let C_{vm} denote the random variable whose value is the time at which a consensus is reached in a voter model protocol. It can be shown that $\mathbb{E}(C_{vm}) = O(\log(n) \max_{i,j} h_{ij})$, where h_{ij} is the expected hitting time of node j of a random walk starting from node i (see [3] for a proof of this). This result is established by making an observation that makes a connection between the voter model protocol and another stochastic process called *coalescing random walk* on the same undirected graph G.

Coalescing random walk works as follows: Initially there is a particle on each node, each following an independent simple random walk (i.e. move to a neighbor uniformly at random) on G. In each round one particle is picked uniformly at random to make one move, if the neighboring node it moves to is already occupied by another particle, then these two particles *coalesce* into a new particle that is identical to any of the two parent particles and still follows the same simple random walk on G. It is shown in [3] that *coalescing random walk* is the dual process of voter model protocol in the sense that for any $t \geq 0$ and for any node $i \in V$, the following two probabilities are always the same

1. The probability that a consensus is reached and the consensus color originates from node i[1] after t rounds in the voter model protocol on G;

2. The probability that all the n particles have coalesced into a single particle and this particle lands on node i after t rounds in the coalescing random walk on G.

On the other hand, it is also well-known that for any graph G with self-loops (i.e. $i \in \mathcal{N}(i)$), $h_{ij} = O(n^3)$ for any node i, j [73], so $\mathbb{E}(C_{vm}) = O(n^3 \log(n))$. We summarize this in the following theorem.

[1] Note different nodes may have the same initial color, in this case we still treat them as distinct by differentiating them by their originators — in the voter model protocol it is possible to trace an eventual consensus back to a node where that color originates from.

Theorem 1 ([3])**.** *For any initialization, it takes $O(n^3 \log(n))$ time in expectation for all the n nodes to converge to a consensus opinion in a voter model protocol.*

This leads to the following corollary.

Corollary 1. *For any initialization, for any small constant $\theta > 0$,*[2] *after $O(n^{3+\theta} \log(n))$ rounds into the voter model protocol, the probability that a consensus has not been reached is $O(1/2^{n^\theta})$.*

Proof. By Theorem 1, there exists a constant C such that for any initialization, $\mathbb{E}(C_{vm}) \leq Cn^3 \log(n)$. Therefore, by Markov's inequality (see Theorem 11 in Appendix A.1), after $2Cn^3 \log(n)$ rounds the probability that a consensus has not been reached is at most $1/2$. Since this is true for any initialization, running the protocol for $2Cn^{3+\theta} \log(n)$ rounds can be view as running the protocol in n^θ independent segments, each running for $2Cn^3 \log(n)$ rounds. Therefore, the probability that a consensus is not reach after $2Cn^{3+\theta} \log(n)$ rounds is at most $\dfrac{1}{2^{n^\theta}}$. □

Denote by π the stationary distribution of a random walk on G, i.e. $\pi(i) = \dfrac{d_i}{2m}$ for all $i \in V$ and $\pi(S) = \sum_{i \in S} \dfrac{d_i}{2m}$ for all $S \subset V$. The next theorem also largely follows from established results in literature.

Theorem 2. *Let $S \subset V$ be the set of nodes initialized to opinion α in a voter model protocol, then after $O(n^{3+\theta} \log(n))$ rounds the probability that an α-consensus is reached differs from $\pi(S)$ by $O(1/2^{n^\theta})$.*

Proof. Note the duality of voter model protocol and coalescing random walk tells us that the probability that an α-consensus is reached in the voter model protocol is the same as the probability that all the n particles have coalesced and land on a node from set S. This duality combined with Corollary 1 shows that there exists constant C such that after $Cn^{3+\theta} \log(n)$

[2] Throughout the rest of the paper, θ will be used as a parameter to the running time of the voter model protocol and share the same meaning as defined here. It is also assumed that $\theta < 1$.

rounds the probability that not all n particles have coalesced is exponentially small in n. Suppose after $Cn^{3+\theta}\log(n)$ rounds, all the n particles do coalesce, then from this point on the coalesced particle will follow a simple random walk on G. If we run for another n rounds, the probability that the particle lands on a node from set S differs from $\pi(S)$ by $O(c^n)$, for some constant $c \in (0,1)$ by the *convergence theorem* on Markov chain (see Theorem 13 in Appendix A.3).

Therefore after $Cn^{3+\theta}\log(n) + n$ rounds in coalescing random walk the probability that all the n particles have coalesced into a single particle and this particle lands on a node from S is

$$\begin{aligned} & (1 - O(1/2^{n^\theta}))(\pi(S) \pm O(c^n)) \\ = \;& \pi(S) \pm O(c^n) - \pi(S)O(1/2^{n^\theta}) \mp O(c^n)O(1/2^{n^\theta}) \\ = \;& \pi(S) - O(1/2^{n^\theta}), \end{aligned}$$

where the last step follows from $\theta < 1$, an assumption made in the interest of making all the algorithms discussed in this chapter have a faster running time (note making θ larger can only decrease the difference between the probability of interest and $\pi(S)$).

Now if we apply duality again, it translates to that after $Cn^{3+\theta}\log(n)+n = O(Cn^{3+\theta}\log(n))$ rounds in the voter model protocol the probability that an α-consensus is reached differs from $\pi(S)$ by $O(1/2^{n^\theta})$, an amount that is exponentially small in n. □

Theorem 1 and 2 allow us to conclude that after $O(n^{3+\theta}\log(n))$ rounds into a voter model protocol, with high probability *some* consensus is reached. In particular, let $B, R \in V$ be the set of nodes initialized to *blue* and *red* respectively, the probability of reaching a b-consensus (resp., r-consensus) differs from $\pi(B)$ (resp., $\pi(R)$) by an amount that is exponentially small in n. Recall our goal in solving the NBVP is to find an efficient protocol that converges to the collectively preferred consensus with high probability. Since the voter model does not even consider w_i, it is clear that it does not solve the NBVP. (The voter model does not specify how initialization is done, however it is easy to prove that even if $\mathcal{I}$ is allowed to depend on w_i in an *arbitrary* way, no voter model protocol solves the NBVP.)

Therefore, the logical next thing to consider in order to solve the NBVP is to allow $\mathcal{F}$ to in addition depend on w_i. And this leads us to the natural extension of the classic voter model that we are going to define in the next section: the *biased voter model*.

2.4 The Biased Voter Model

Discussion from the previous section reveals that in order to solve the NBVP, it is necessarily to allow $\mathcal{F}$ to depend on w_i in addition to f_i, so that how an individual changes his opinion is influenced by his neighbors as well as his own intrinsic preferences. A natural class of $\mathcal{F}$ that reflect an individual's preference (or *bias*) are those that let him assume his preferred opinion α with probability higher than $f_i(\alpha)$, which is the probability he assumes opinion α in the voter model. We call the resulting model the *biased voter model* and define it formally as follows.

Definition 2 (Biased Voter Model). *The biased voter model is a class of protocols of the form* $(\mathcal{I}, \mathcal{F})$ *where for some constant* $\epsilon > 0$,

$$P\{\mathcal{F}(f_i, w_i) = \alpha\} \begin{cases} \geq \min\{f_i(\alpha) + \epsilon, 1\} & \textit{if } w_i(\alpha) > \dfrac{1}{2}; \\ \leq \max\{f_i(\alpha) - \epsilon, 0\} & \textit{otherwise.} \end{cases}$$

and $\mathcal{I}$ *is allowed to depend on* w_i *in an arbitrary way.*

Definition 2 is a generic one which only defines biased updating function $\mathcal{F}$ qualitatively without specifying how exactly it is computed. A natural choice is where each agent plays α with probability proportional to the product $f_i(\alpha)w_i(\alpha)$ [8]. In this model an agent balances their preferences with the behavior of their neighbors in a simple multiplicative fashion and we call this the *multiplicative* biased voter model.

We note the extension to the biased voter model in Definition 2 is fairly general in that $\mathcal{F}$ is allowed to include a broad class of local stochastic updating rules that reflect a node's preferences; and $\mathcal{I}$ is allowed to be *arbitrary* although it has to be independent of G. These seemingly provide us with a lot of power in the design of protocols; but perhaps surprisingly, in this section we prove that even this broad class of biased voting rules is insufficient to solve the NBVP:

Theorem 3. *No biased voter model protocol* $(\mathcal{I}, \mathcal{F})$ *solves the NBVP in the following sense: If* $\mathcal{I}$ *initializes a node to its preferred opinion with positive probability, then there exists graphs where it takes exponential time in expectation for* any *biased voter model protocol to reach a consensus; otherwise, there exists settings where it converges to the non-preferred opinion with probability 1.*

Note by our definition, the biased voter model constitutes of only *simple* and *local* protocols where $\mathcal{I}$ can only depend on a node's local information. In particular, $\mathcal{I}$ cannot be a function of n. Therefore, although Theorem 3 shows that no protocol from the class that we are interested in (i.e. the biased voter model) can solve the NBVP, it does not rule out more complicated protocols where $\mathcal{I}$ can be dependent on more global information.

The rest of this section is organized as follows. In Section A.1, we prove a technical lemma about a certain Markov chain that can be represented by a line graph. We then use this lemma to prove Theorem 3 in 2.4.2, by constructing an example on a line graph where for any voter model protocol $(\mathcal{I}, \mathcal{F})$, it either takes exponential time to converge to the globally preferred color, or convergence is to the globally non-preferred color, both violations of the NBVP requirements. A similar result is proved for a clique graph in Section 2.4.3.

2.4.1 A Markov Chain Lemma

Consider a Markov Chain on a line graph of n nodes, namely $s_1, s_2, ..., s_n$, where transition does not happen beyond adjacent nodes. In this subsection we want to show that if at any state s_i $(1 < i < n)$, the Markov chain is more likely to go 'backward' to state s_{i+1} than to go 'forward' to state s_{i-1}, then starting from state s_i (where $i \geq 2$), it takes exponential time in expectation to hit state s_1. While this is perhaps intuitive, we will need this result to be in a particular form for the later reduction.

Here are a couple of notations: Let $p_{i,j}$ $(i, j \in [n])$ be the transition probability from node i to j, by construction $p_{i,j} = 0$ if $|i - j| > 1$. Simplify notation by writing $p_i = p_{i,i-1}$ and $q_i = p_{i,i+1}$, which are the ‘forward’ and ’backward’ transition probability, respectively. Define h_i to be the expected number of rounds for the process to hit state s_1 for the first time, given that it starts from state s_i. Let

$$\begin{aligned}\gamma_{max} &= \max_{i\in\{2,3,\ldots,n-1\}} \frac{q_i}{p_i}\\ \gamma_{min} &= \min_{i\in\{2,3,\ldots,n-1\}} \frac{q_i}{p_i}\end{aligned}$$

We have the following lemma.

Lemma 1. *If $\gamma_{min} \geq 1 + \epsilon$ for some constant $\epsilon > 0$, then h_i $(i \geq 2)$ is exponential in n.*

Proof. We first claim that $h_i - h_{i-1} > \dfrac{\gamma_{min}^{n-i}}{p_n}$. To prove this claim, note h_i satisfies the following linear system

$$h_i = \begin{cases} 0 & (i = 0)\\ 1 + p_i h_{i-1} + q_i h_{i+1} + (1 - p_i - q_i) h_i & (2 \leq i \leq n-1)\\ 1 + p_n h_{n-1} + (1 - p_n) h_n & (i = n)\end{cases}$$

It is clear $h_j - h_{j-1} > 0$ for all $j > 1$ as a process starting from state s_j has to hit s_{j-1} before hitting s_1. Let $h_j - h_{j-1} = \lambda_j$, combining it with $h_{j-1} = 1 + p_{j-1}h_{j-2} + q_{j-1}h_j + (1 - p_{j-1} - q_{j-1})h_{j-1}$ gives $h_{j-1} - h_{j-2} = \dfrac{1 + q_{j-1}\lambda_j}{p_{j-1}}$, which in turn implies $\lambda_{j-1} > \left(\dfrac{q_{j-1}}{p_{j-1}}\right)\lambda_j > \gamma_{min}\lambda_j$. Repeating this inductively gives $\lambda_i > \gamma_{min}^{n-i}\lambda_n$. Since $\lambda_n = \dfrac{1}{p_n}$, this proves the claim.

Immediately following from this claim, we have $h_2 = h_2 - h_1 > \dfrac{\gamma_{min}^{n-2}}{p_n} \geq (1 + \epsilon)^{n-2}$ if $\gamma_{min} \geq 1 + \epsilon$. Since $h_i > h_2$ whenever $i > 2$, this completes the proof. □

2.4.2 Exponential Time Convergence on a Line

Our goal in this subsection is to prove Theorem 3. To this end, first consider the biased voter model on the following 3-regular line graph.

A Line Graph. *G is a line graph of $2n$ nodes, where the left half prefers* blue *and the right half prefers* red. *The leftmost and rightmost node each has two self-loops and all the other nodes have one self-loop.*

We prove two lemmas (Lemma 2 and 3) about this particular setting, as a preparation for the proof of the main theorem.

Lemma 2. *For any biased voter model protocol $(\mathcal{I}, \mathcal{F})$, given that $\mathcal{I}$ results in an initialization where all nodes initialized to* blue *are to the left of all nodes initialized to* red, *it takes exponential time in expectation to reach a consensus on the line.*

Proof. We prove this by reducing this stochastic process to the Markov process described in Section A.1. First observe that since we start with a coloring where all blues are to the left of all reds, this will hold as an invariant throughout the evolution of the whole process and the only way for the coloring to evolve is for the *blue* node adjacent to a *red* neighbor to convert to *red*, or for its *red* neighbor to convert to *blue*.

Therefore, we can always describe the state by a pair of integers $(b, 2n - b)$, where b is the number of blue-colored nodes. Now if we lump two states, $(b, 2n - b)$ and $(2n - b, b)$, into one, this model is exactly the Markov process (with $n + 1$ states) described in Section A.1 with $s_i = \{(i, 2n - i), (2n - i, i)\}$ for $i = \{0, 1, ..., n\}$.

And by definition of *biased voter model* and the way the Markov chain is constructed in the above, we have $p_i \leq \dfrac{1/3 - \epsilon}{2n}$ and $q_i \geq \dfrac{1/3 + \theta}{2n}$ $(i \in \{1, 2, ..., n - 1\})$ for some constant $\epsilon > 0$. Therefore, $\gamma_{min} \geq \dfrac{1/3 + \theta}{1/3 - \epsilon} = 1 + \delta$, for some constant $\delta > 0$. Invoking Lemma 1 shows that it takes exponential time to hit s_0 starting from state s_i (where $i \geq 1$). Therefore, it takes exponential time to reach a consensus given that one starts with a coloring where all nodes initialized to *blue* are to the left of all nodes initialized to *red*. □

Lemma 3. *For any biased voter model protocol* $(\mathcal{I}, \mathcal{F})$*, if* $\mathcal{I}$ *initializes a node* i *to his preferred opinion with positive probability, then it takes exponential time in expectation to reach a consensus on the line.*

Proof. Note $\mathcal{I}$ is independent of G, therefore whenever it initializes with positive probability, the probability is independent of n. In particular, the probability that $\mathcal{I}$ initializes the leftmost node to *blue and* the rightmost node to *red* is not exponentially small in n. Therefore we are through if we can show that *given* the leftmost node is initialized to *blue* and the rightmost to *red*, it takes exponential time to reach a consensus.

Lemma 2 does not differentiate between a b-consensus and a r-consensus. If we concern ourselves only with the outcome of, say a b-consensus, it can be shown that it still takes exponential time to reach given that we start from the same initialization described in Lemma 2 (i.e. all blues are to the right of all reds). We prove this by first observing that, conditioning on that a b-consensus is reached, the time taken is distributed exactly the same as in the modified stochastic process on the same $2n$-node line graph, with the only difference being making the leftmost node extremely biased towards *blue* so that it always votes for *blue* regardless of his neighbor's opinion. Therefore we only need to prove it takes exponential time for this modified process to reach a consensus (which can only be a *blue* one), and this follows from Lemma 2.

Of course by a similar argument we can show that starting from an initialization where all blues are to the left of all reds, it takes exponential time to reach a r-consensus.

Now consider the initialization where the leftmost node is *blue* and rightmost node is *red* and call this the case of interest. Compare it with the initialization where the leftmost node is *blue* and all the other $n-1$ nodes are *red*, the r-consensus time of this case is clearly upper bounded by that of the case of interest. By the above discussion, it takes exponential time to reach a r-consensus even when we start with only the leftmost node *blue*; therefore, it takes exponential time to reach a r-consensus for the case of interest. By the same argument, it also takes exponential time to reach a b-consensus for the case of interest. In sum this allows us to conclude that it takes exponential time to reach a consensus given that $\mathcal{I}$ initializes the leftmost node to *blue* and the rightmost to *red*. □

We are now ready to give a proof for Theorem 3.

Proof. (of Theorem 3) In Lemma 3, we have already shown that any biased voter model protocol $(\mathcal{I}, \mathcal{F})$ fails to solve the NBVP (taking exponential time to converge) if we restrict $\mathcal{I}$ to the kind of initialization functions that initializes a node to its preferred opinion with positive probability. It is easy to see that $(\mathcal{I}, \mathcal{F})$ also fails for any $\mathcal{I}$ that does the opposite, in which case $\mathcal{I}$ initializes a node to his not-preferred opinion with probability 1: Simply construct a graph consists solely of nodes that prefer *blue*, and $\mathcal{I}$ initializes it to a r-consensus. Therefore, we conclude that *any* biased voter model protocol $(\mathcal{I}, \mathcal{F})$ fails to solve the NBVP. □

Note since the line graph we construct above is 3-regular, we have actually shown a stronger version of Theorem 3: Even if we allow both $\mathcal{I}$ and $\mathcal{F}$ to depend on d_i, no protocol $(\mathcal{I}, \mathcal{F})$ can solve the NBVP. We also note that a similar exponential convergence result can be shown for clique in certain settings.

2.4.3 Exponential Time Convergence on a Clique

In this section, we show that it takes exponential time for *any* multiplicative biased voter model protocol to converge on a clique in the following setting.

A Clique. *G is a clique of $2n$ nodes, of which n prefer* blue, *n prefer* red *and each node has a self-loop. Denote by $s_{i,j}$ $(i, j \in \{0, 1, ..., n\})$ the state where out of the n (resp. n) nodes* *who prefer blue (resp. red), i (resp. j) of them are colored blue and $n-i$ (resp. $n-j$) of them are colored red. Denote by $f = \dfrac{i+j}{2n}$ the blue fraction.*

We state our result in Theorem 4, which says that starting from an initialization with f bounded away from both 0 and 1, in expectation it takes exponential time for any biased voter model protocol to reach a consensus on G.

Theorem 4. *For any multiplicative biased voter model protocol $(\mathcal{I}, \mathcal{F})$ where all nodes share the same strength of bias $w \geq \dfrac{1}{2} + \epsilon$ (i.e. for any $i \in V$, if i prefers $\alpha \in \{b, r\}$, then $w_i(\alpha) = w$), it takes exponential time in expectation to reach either a* blue-*consensus or a* red-*consensus starting from any state with* blue fraction *f in the interval $[\delta, 1-\delta]$, for some constant $\delta > 0$.*

Proof. To prove this, we show any uniformly-biased multiplicative protocol $(\mathcal{I}, \mathcal{F})$ on a clique can be viewed as a biased voter model protocol $(\mathcal{I}', \mathcal{F}')$ on the line graph described in Sec. 2.4.2 (i.e. a line graph of $2n$ nodes where left half prefers blue and right half prefers red), therefore invoking Lemma 2 and 3 establishes the result.

First observe that the stochastic process on the clique can be reduced to a Markov chain on a $(n+1) \times (n+1)$ 2-dimensional grid, where a node (i, j) corresponds to state $s_{i,j}$. We arrange the grid so that the northwest node corresponds to $s_{0,0}$ and the southeast node corresponds to $s_{n,n}$. Denote by $p_n(i, j)$ the probability of a transition from state $s_{i,j}$ to state $s_{i-1,j}$ (i.e. going north), and define p_s, p_w and p_e analogously. We have

$$\begin{aligned}
p_n(i,j) &= \frac{(1-f)(1-w)}{fw + (1-f)(1-w)} \cdot f_b \\
p_s(i,j) &= \frac{fw}{fw + (1-f)(1-w)} \cdot \left(\frac{1}{2} - f_b\right) \\
p_w(i,j) &= \frac{(1-f)w}{f(1-w) + (1-f)w} \cdot f_r \\
p_e(i,j) &= \frac{f(1-w)}{f(1-w) + (1-f)w} \cdot \left(\frac{1}{2} - f_r\right)
\end{aligned}$$

where $f_b = \frac{i}{2n}$, $f_r = \frac{j}{2n}$ and $f = f_b + f_r$. Now define new state $s'_k = \{s_{i,j} \mid i + j = k\}$, $k \in \{0, 1, ..., 2n\}$. This reduces the Markov chain on the clique to the Markov chain on the line. Now to establish our above claim that the Markov chain on the clique can be viewed as a biased voter model protocol on the line, all that we need to show are the following:

1. The transition probability from s'_k to s'_{k-1} is greater than the transition probability from s'_k to s'_{k+1} by at least a constant gap when f is bounded away from 1 and $\frac{1}{2}$, i.e. $p_n(i,j) + p_w(i,j) - p_s(i,j) - p_e(i,j) \geq \zeta$ for some constant $\zeta > 0$ whenever $1 - \delta \geq f \geq \frac{1}{2} + \delta$ for some constant $\delta > 0$.

2. The transition probability from s'_k to s'_{k-1} is less than the transition probability from s'_k to s'_{k+1} by at least a constant gap when f bounded away from 0 and $\frac{1}{2}$, i.e. $p_n(i,j) + p_w(i,j) - p_s(i,j) - p_e(i,j) \leq \zeta$ for some constant $\zeta > 0$ whenever $\delta \leq f \leq \frac{1}{2} - \delta$ for some constant $\delta > 0$.

We give proof for the first part in the following, and the proof of the second part is essentially the same.

$$\begin{aligned}
& p_n(i,j) + p_w(i,j) - p_s(i,j) - p_e(i,j) \\
= \; & f_b \left[\frac{(1-f)(1-w)}{(1-f)(1-w) + fw} + \frac{fw}{fw + (1-f)(1-w)} \right] \\
& + f_r \left[\frac{(1-f)w}{f(1-w) + (1-f)w} + \frac{f(1-w)}{f(1-w) + (1-f)w} \right] \\
& - \frac{1}{2} \left[\frac{fw}{fw + (1-f)(1-w)} - \frac{f(1-w)}{f(1-w) + (1-f)w} \right] \\
= \; & f - \frac{1}{2} \left[\frac{fw}{fw + (1-f)(1-w)} - \frac{f(1-w)}{f(1-w) + (1-f)w} \right] \\
= \; & \left(f - \frac{1}{2} \right) \left[1 - \frac{w(1-w)}{f^2 w(1-w) + f(1-f)(1-w)^2 + f(1-f)w^2 + (1-f)^2 w(1-w)} \right] \\
= \; & \left(f - \frac{1}{2} \right) f(1-f)(4w^2 - 4w + 1) \\
\geq \; & 4\epsilon^2 \delta^2 (1 - \delta)
\end{aligned}$$

where the last inequality follows form the face that $w \geq \frac{1}{2} + \epsilon$, and $\frac{1}{2} + \delta \leq f \leq 1 - \delta$. □

2.5 A Protocol for NBVP

Previous discussions establish the limit of the classic voter model protocol and its natural extension to the biased voter model when it comes to solving NBVP. We are thus interested in the question: What are the (ideally minimal) extensions to the biased voter model needed to obtain a simple, efficient and local protocol for solving the NBVP?

In this section, we give one answer to this question by providing a provable solution to the NBVP that employs the following extensions:

1. Make the degree of G , $d(G) = \max_{i \in V} d_i$, an input to $\mathcal{F}$. This allows each node to increase its influence by adding $d(G) - d_i$ self-loops.[3]

2. Allowing initialization and evolution of a node's opinion be dependent on its degree in G;

3. Allowing multiple identical copies of the protocol to be run in G and having each node vote for the opinion converged to more frequently among the multiple runs. This can be implemented by having a slightly more powerful schedule that after every $n^{3+\theta} \log(n)$ steps, re-initializes each node.

We give the protocol in Algorithm 1. This protocol consists of $T = poly(n)$ phases. In each phase, each node simultaneously and independently initializes his opinion to either *blue* or *red* with probabilities exactly proportional to his preferences of these two opinions. And then the standard voter model protocol is run on the augmented $d(G)$-regular graph, where each node has added $d(G) - d_i$ extra self-loops. The reason for adding self-loops is to allow

[3] We note that in a strategic or game-theoretic setting, a node, among other deviations that he can make, might of course choose to add more self-loops than this to further increase its influence. We examine this topic in Sec. 2.6

Algorithm 1 A Simple and Local Voting Protocol

```
Each node $i$ maintains an array $R_i$ of size $T$
Each node $i$ adds $d(G) - d_i$ self-loops
for $phase = 1$ to $T$ do
    Each node $i$ simultaneously and independently initializes its color to $b$ or $r$ with probability $w_i(b)$ and $w_i(r)$, respectively;
    Run the (standard) voter model protocol for $n^{3+\theta}\log(n)$ rounds
    Each node $i$ records his last round opinion of this phase of the voter model process in $R_i[phase]$
end for
// Each node now has his local 'outcomes' of all $T$ phases
Each node $i$ identify a majority between $b$ and $r$ in $R_i$; breaking ties arbitrarily
Each node $i$ vote for this majority identified as his final vote
```

individuals of otherwise different degrees to have equal degree, and thus 'influence', in the voter model protocol. At the end of each phase, the standard voter model protocol is run for $n^{3+\theta}\log(n)$ rounds and each node records his opinion in the last round as the 'outcome' of this phase. After T phases, each node identifies the majority between *blue* and *red* among the n outcomes; he then vote for this majority as his *final vote*.

We now proceed to prove that Algorithm 1 indeed solves the NBVP.

Lemma 4. *Each of the T phases of Algorithm 1 leads to a blue-consensus (resp. red-consensus) with probability that differs from* $\frac{\sum_{i\in V} w_i(b)}{n}$ $\left(\textit{resp. } \frac{\sum_{i\in V} w_i(r)}{n}\right)$ *by* $O(1/2^{n^\theta})$.

Proof. We give proof for the case of a *blue*-consensus, and the proof for *red*-consensus follows a similar argument.

Denote by $p(B)$ the probability that $B \subseteq V$ is the set of nodes initialized to blue, and $p(b \mid B)$ the probability that there is a *blue*-consensus after a single phase of Algorithm 1 given that B is the set of nodes initialized to blue. So

$$P_b = \sum_{B \in 2^V} p(B)p(b \mid B)$$

is the probability that a single phase of Algorithm 1 results in a b-consensus. By Theorem 2 we have $|p(b \mid B) - \pi(B)| = O(1/2^{n^\theta})$, or $p(b \mid B) = \pi(B) \pm O(1/2^{n^\theta})$, therefore

$$\begin{aligned}
P_b &= \sum_{B \in 2^V} p(B)(\pi(B) \pm O(1/2^{n^\theta})) \\
&= \sum_{B \in 2^V} p(B)\pi(B) \pm O(1/2^{n^\theta}) \sum_{B \in 2^V} p(B) \\
&= \sum_{B \in 2^V} p(B)\pi(B) \pm O(1/2^{n^\theta}) \\
&= \frac{\sum_{B \in 2^V} p(B)|B|d(G)}{nd(G)} \pm O(1/2^{n^\theta}) \\
&= \frac{\sum_{i \in V} w_i(b)}{n} \pm O(1/2^{n^\theta})
\end{aligned}$$

Therefore, we conclude that $\left| P_b - \frac{\sum_{i \in V} w_i(b)}{n} \right| = O(1/2^{n^\theta})$ is exponentially small in n. □

Recall our goal is to let V converge to the collectively preferred consensus. And by definition of the NBVP one opinion is significantly preferred than the other, i.e. $|\sum_{i \in V} w_i(b) - \sum_{i \in V} w_i(r)| \geq \epsilon$ for some possitive constant ϵ; this assumption turns out to be sufficient for Algorithm 1 to achieve this goal if we set $T = n^{2+\tau}$ for any constant $\tau > 0$. We prove this in the following theorem.

Theorem 5. *Setting $T = n^{2+\tau}$ (for any constant $\tau > 0$) in Algorithm 1 solves the NBVP. Specifically, Algorithm 1 let V converge to the collectively preferred consensus with probability $1 - O(c^{n^\tau})$ (for some constant $c \in (0,1)$) and it runs in $O(n^{5+\theta+\tau} \log(n))$ time.*

Proof. Without loss of generality, we assume *blue* is the collectively preferred color. By Lemma 4 we have $P_b \geq \frac{\sum_{i \in V} w_i(b)}{n} - O(1/2^{n^\theta})$ and $P_r \leq \frac{\sum_{i \in V} w_i(r)}{n} + O(1/2^{n^\theta})$. Therefore the gap between P_b and P_r is at least $\frac{\epsilon}{n} - O(1/2^{n^\theta})$, so there exists a positive constant $\delta < \epsilon$ such that the gap between P_b and P_r is at least $\frac{\delta}{n}$ whenever n is sufficiently large.

Let T_b and T_r be the number of b-consensuses and r-consensuses among the T trials, the bad event happens when $T_b \leq T_r$. For this bad event to happen, the event $T_b <$

$\left(P_b - \frac{1}{3} \cdot \frac{\delta}{n}\right) T$ has to happen. We now use Chernoff-Hoeffding bound (see Appendix A.2 for details) to show that when $T = O(n^{2+\tau})$, this happens with probability exponentially small in n.

$$\begin{aligned} P\left(T_b < (P_b - \frac{\delta}{3n})T\right) &= P\left(T_b < TP_b - \frac{\delta}{3n}T\right) \\ &= P\left(T_b - \mathbb{E}(T_b) < -\frac{\delta}{3n}T\right) \\ &\leq e^{-2T(\delta/3n)^2} \end{aligned}$$

Therefore, setting $T = n^{2+\tau}$ makes $P\left(T_b < (P_b - \frac{\delta}{3n})T\right) = O(c^{n^\tau})$ for some constant $c \in (0, 1)$. Since $P(T_b \leq T_r) < P\left(T_b < (P_b - \frac{\delta}{3n})T\right)$, we have shown that Algorithm 1 converge to *blue* with probability $1 - O(c^{n^\tau})$. Also it is obvious to see that the running time is $O(n^{5+\theta+\tau}\log(n))$. This completes the proof. □

We note that the above analysis in fact suggests that the running time of Algorithm 1 can be expressed as a product of the mixing time of the voter model protocol on G and $n^{2+\tau}$, so that on graphs where random walk mixes faster (e.g. expanders), Algorithm 1 has a running time better than $O(n^{5+\theta+\tau}\log(n))$.

Before closing this section, we note that there is an alternative protocol that is a natural variant of Algorithm 1. In this variant, we do not need to let each node know $d(G)$, instead we introduce a third opinion u that we call *undecided*.[4] We then modify Algorithm 1 so that at the beginning of each of the T phases, each node initializes its opinion to *blue*, *red*, and *undecided* with probability $\frac{w_i(b)}{d(i)}$, $\frac{w_i(r)}{d(i)}$, and $\frac{d(i)-1}{d(i)}$, respectively. These initialization probabilities are properly chosen so that the probability of reaching an α-consensus ($\alpha \in \{b, r\}$) is proportional to $\sum_{i \in V} w_i(\alpha)$. After the initialization phase, the alternative protocol does the same thing as Algorithm 1 by running the classic voter model protocol for $n^{3+\theta}\log(n)$ rounds. And using essentially the same analysis it can be shown that this alternative protocol also solves the NBVP in polynomial time.

[4] It is interesting to note that this extension of giving nodes the ability to temporarily declare oneself *undecided* has obvious analogue in many real political processes. And the same can be said about the extension of allowing multiple runs in an election.

2.6 An ϵ-Nash Protocol for the Networked Biased Voting Game

Our protocol for solving NBVP assumes that each individual will actually follow the protocol honestly. However in a strategic setting, an individual may have incentives to deviate from the prescribed protocol. For example, a node i who prefers *blue* may deviate from Algorithm 1 in a way that increases the chance of reaching a *blue*-consensus, even when this consensus is not collectively preferred.

This naturally leads us to consider the Networked Biased Voting Game (NBVG), which is an extension of NBVP to the strategic, or game theoretic, setting. In NBVG, a node with preference $(w_i(b), w_i(r))$ receives payoff $w_i(b)$ (resp., $w_i(r)$) if the game results in an unanimous global *blue*-consensus (resp., *red*-consensus) and payoff 0 if no consensus is reached. A solution to NBVG is a protocol that solves the NBVP (which must be *simple and local* and in polynomial time converge to the collectively preferred consensus with high probability) and at the same time constitutes a Nash equilibrium of the game. We note that NBVG may also be viewed as a distributed, networked version of the classic “Battle of the Sexes” game, or as a networked coordination game [68].

In the rest of this section, we show the existence of a protocol that is an ϵ-approximate Nash equilibirum, or ϵ-Nash for short, of NBVG. This means although a node can deviate unilaterally from this protocol and increases his expected payoff, the amount of this increase is at most ϵ and we show ϵ is exponentially small in n and can be made arbitrarily small. To this end, we need to make the following mild assumptions.

1. The removal of any node from G leaves the remaining graph connected. Formally, let G_{-i} be the graph induced by $V\backslash\{i\}$, we assume G_{-i} is connected for all $i \in V$.

2. The exclusion of any node does not change the collectively preferred consensus, and moreover, it still leaves a significant (constant) gap between $\sum_{j\in V(G_{-i})} w_j(b)$ and

$\sum_{j \in V(G_{-i})} w_j(r)$.

3. Each node i is identified by a unique ID, $ID(i)$, which is an integer in $\{1, 2, ..., n\}$.

Our ϵ-Nash protocol consists of n runs of the non-Nash protocol Algorithm 1, each on a subgraph G_{-i}. Each run of Algorithm 1 polls the majority opinion of $V \backslash \{i\}$, which by assumption is the same as that of V; however by excluding i from participating, we prevent him from any manipulation of this particular run of the non-Nash protocol. When all the n runs of non-Nash is done, each node ends up with $n-1$ 'polls' and with high probability they should all point to the same collectively preferred consensus. In case it does not, it is strong evidence that some run(s) of the non-Nash protocol had been manipulated and the contingency plan is for each node to ignore all the polling results entirely and toss a (private) fair coin to decide whether to vote for *blue* or *red* — and this turns out to be a sufficient deterrent of unilateral deviation from the non-Nash protocol.

We note conceptually we are making yet another simple extension in the protocol's expressiveness by allowing it to be run on a subgraph G_{-i}. To implement this, it is important for each node i to be uniquely identified by his neighbors so that they know when to ignore i; and this is the reason we need assumption 3 listed above. We give this ϵ-Nash protocol in Algorithm 2 and claim the following theorem.

Theorem 6. *Algorithm 2 constitutes an ϵ-Nash equilibrium of the NBVG. Algorithm 2 runs in $O(n^{6+\theta+\tau}\log(n))$ time and $\epsilon = O(nc^{n^{\tau}})$ for some constant $c \in (0,1)$.*

Proof. Suppose each node follows the protocol faithfully, by our assumption that the exclusion of any node does not change the collectively preferred consensus, say *blue*, each of the n runs of Algorithm 1 results in a b-consensus with probability $1 - O(c^{n^{\tau}})$ by Theorem 5 (for some constant $c \in (0,1)$). So by union bound the probability that all the n runs of Algorithm 1 have all resulted in a b-consensus is at least $1 - O(nc^{n^{\tau}})$. Therefore the final votes result in the collectively preferred b-consensus with high probability.

Algorithm 2 A Simple and Local Protocol that is ϵ-Nash

```
Each node $i$ maintains an array $E_i$ of size $n - 1$
for $episode = 1$ to $n$ do
    Let $i$ be the node such that $ID(i) = episode$
    Run Line 1 - 8 of Algorithm 1 on $G_{-i}$
    Each node $j \in V \backslash \{i\}$ records in $E_j[episode]$ the majority between $b$ and $r$ he identifies
    on Line 8 of (this run of) Algorithm 1
end for
// Each node has now participated in $n-1$ runs of Algorithm 1
for all $i \in V$ do
    if both $b$ and $r$ are present in the $n - 1$ entries of $E_i$ then
        Tossing a private fair coin to decide between $b$ and $r$, and vote for it as $i$'s final vote
    else
        Vote for the only opinion present as $i$'s final vote
    end if
end for
```

Now we examine why faithfully executing this protocol is an ϵ-Nash strategy for each node, where $\epsilon = O(nc^{n^\tau})$. For a node i that prefers *red* (i.e. the opinion not collectively preferred), assuming everyone else is following Algorithm 2, the expected payoff to i for doing the same is at least his payoff in a b-consensus minus a number exponentially small in n, i.e. $O(nc^{n^\tau})$, because of the exponentially small probability that a b-consensus may not be reached even if every node follows Algorithm 2 faithfully. Now we consider what happens if it deviates. There are two stages during which i can deviate: the first or the second for-loop in Algorithm 2. i's effort during the first for-loop is obviously immaterial if none of the $n-1$ runs of Algorithm 1 is turned into a r-consensus, and in this case, with high probability all the n runs of Algorithm 1 result in a b-consensus. Therefore, i will have no incentive to deviate during the second for-loop because everyone else is going to vote for b.

Next consider the case where i successfully turns some of the global outcomes of Algorithm 1 into a r-consensus (i.e. all nodes identify r as the majority on Line 8 of Algorithm 1), then with high probability (at least $1 - O(c^{n^\tau})$) the n runs of Algorithm 1 result in both r-consensus and b-consensus because the single run of it without i participating results in a b-consensus with probability $1 - O(c^{n^\tau})$. In this case, at least $n - 2$ nodes out of $V \backslash \{i\}$ see both *blue* and *red* as outcomes from the $n-1$ runs of Algorithm 1 they each participated in and will vote for either b or r by tossing a private fair coin, which means the probability of reaching a b-consensus or r-consensus among them, independent of whatever strategy i adopts in the second for-loop, is $\left(\frac{1}{2}\right)^{n-2}$. Therefore, no matter what strategy i adopts in the second for-loop, his expected payoff is exponentially small and obviously worse than what he would have gotten by not deviating. Therefore, we conclude that executing Algorithm 2 faithfully is actually a Nash strategy for i.

Now consider a node j who prefers a b-consensus. By the same discussion as above, Algorithm 2 results in a b-consensus in the final voting with probability at least $1-O(nc^{n^\tau})$, therefore the expected payoff to j is at least his payoff in a b-consensus minus an exponentially small number of $O(nc^{n^\tau})$. Therefore by deviating j can only hope to improve his expected payoff by $O(nc^{n^\tau})$. And this allows us to conclude that each node following Algorithm 2 faithfully constitutes an ϵ-Nash equilibrium for the game, where $\epsilon = O(nc^{n^\tau})$. Also it is easy to see that Algorithm 2 runs in $O(n^{6+\theta+\tau}\log(n))$ time. This completes the proof. □

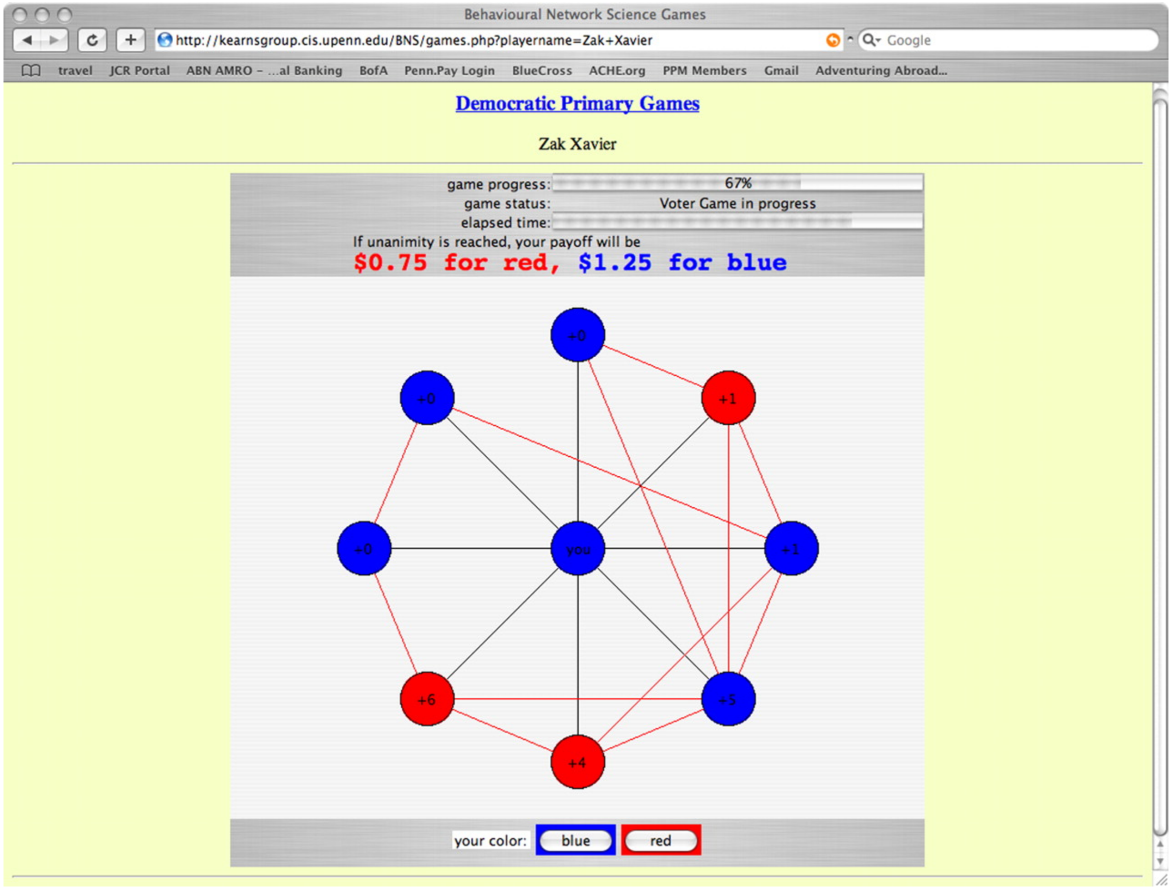
Chapter 3

A Behavioral Study on Biased Voting in Networks

3.1 Introduction

In recent years there has been much research on network based models in game theory, in both the computer science and economics communities. Topics receiving considerable attention include the effects of network topology on equilibrium properties [46, 50, 55, 49, 61], price of anarchy analyses of selfish routing and other networking problems [80], game-theoretic models of network formation (see [31] and citations therein), equilibrium computation in networked settings (see [52, 77] and citations therein), and many others. This large and growing literature has been almost exclusively theoretical, with few accompanying empirical or behavioral studies [21, 36] examining the relevance of the mathematical models to actual behavior.

In this chapter and the next, we report on two series of highly controlled human subject experiments in *networked biased voting* and *networked bargaining*, respectively. We focus on networked biased voting in this chapter, which is modelled exactly the same as the theoretical problem of the same name considered in Chapter 2, and is similarly motivated by distributed collective decision-making processes where balances have to be struck between



Democratic Primary Games

Zak Xavier

game progress: 67%

game status: Voter Game in progress

elapsed time:

If unanimity is reached, your payoff will be

$0.75 for red, $1.25 for blue

Figure 3.1: Screenshot of the user interface for a typical experiment.

diverse individual preferences and a desire for collective unity.

In each experiment, 36 subjects each simultaneously sit at workstations and control the state of a single vertex in a 36-vertex network whose connectivity structure is determined exogenously and is unknown to the subjects. The state of a subject's vertex is simply one of 2 colors (red or blue), and can be asynchronously updated as often as desired during the 1-min experiment. Subjects are able to view the current color choices of their immediate neighbors in the network at all times but otherwise have no global information on the current state of the network (aside from a crude and relatively uninformative "progress bar"; see Fig. 3.1. No communication between subjects outside the experimental platform is permitted.

In addition, each subject is given a financial incentive that varies across the network, and specifies both individual preferences and the demand for collective unity. For instance, one player might be paid $1.25 for blue consensus and $0.75 for red consensus, whereas another might be paid $0.50 for blue consensus and $1.50 for red consensus, thus creating distinct and competing preferences across individuals. However, payments for an experiment are made only if (red or blue) global unanimity is reached, so subjects must balance their preference for higher payoffs with their desire for any payoff at all. A screenshot for a particular subject in a typical experiment is shown in Fig. 3.1.

We note that our experiments may also be viewed as a distributed, networked version of the classic "Battle of the Sexes" game, or as a networked coordination game [68]. Compared with the traditional analyses of these games, we are particularly interested in the effects arising as a result of the interactions of varying network structure and varying incentive schemes. We note that although our experimental framework deliberately omits global "broadcast" mechanisms for consensus (other than the aforementioned progress bar) that are common in many public electoral processes – such as media polls, "mainstream" media reports and analyses – many other real-world sources of both small and large-scale influence can be modeled via network structure. For instance, individuals whose opinion reaches an inordinately large number of others (such as might be expected of some political bloggers) can be modeled by high-degree vertices. Cohesive or close-knit groups of like-minded individuals can be modeled by subsets of vertices with similar incentives and dense connectivity. Our experiments deliberately introduce such structures and others. We also remark that our demand for complete unanimity before any payoffs are made is an abstraction of most real decision-making and voting processes, where a sufficiently strong consensus is typically enough to yield the benefits of unity. Although we expect most of our findings would be robust to such weakening, we leave its investigation to future research.

Our methodology and experiments mix recent lines of thought from algorithmic game theory, behavioral economics and social network theory, and are among the first and largest behavioral experiments on network effects in collective decision making to date. We adopt many of the practices of behavioral game theory [17], which has tended to focus on two-player or small-population games rather than larger networked settings. The experiments described here are part of an extensive and continuing series that have been conducted at the University of Pennsylvania since 2005, in which collective problem-solving from only local interactions in a network has been studied on a wide range of tasks, including graph coloring [56], trading of virtual goods [48], and several other problems. An overarching goal of this line of research is to establish the ways in which network structure and task type and difficulty interact to influence individual and collective behavior and performance.

Our results include a detailed examination of how network structure and incentives influence collective and individual outcome, as well as how individual behavior, or style of play, can be related to his performance. The networks imposed are drawn from models common in social network theory, including preferential attachment graphs, random (Erdös-Rényi) networks, and some carefully designed structures. Among our most striking findings are the following:

- We find that there are well-studied network topologies in which the minority preference consistently wins globally;
- We find that the presence of “extremist” individuals, or the awareness of opposing incentives, reliably improve collective performance;
- We find that certain behavioral characteristics of individual subjects, such as “stubbornness”, are strongly correlated with earnings.

All of the above results and almost all others reported in this chapter are highly statistically significant.

3.2 System and Experiment Methodology

3.2.1 System Description

Experiments were conducted using a distributed networked software system we have designed and built over the past several years for performing a series of behavioral network experiments on different games. This section briefly describes the user's view of that system in our biased voting experiments.

Fig. 3.1 shows a screenshot of the user interface for a typical experiment. Each subject sees only a local ("ego network") view of the global 36-vertex network, showing their own vertex at the center and their immediate neighbors surrounding. Edges between connected neighbors are also shown, as are integers denoting how many unseen neighbors each neighbor has. Vertex colors are the current color choices of the corresponding subjects, which can be changed at any time using the buttons at the bottom. The subject's payoffs for the experiment are shown (in this case $0.75 for global red consensus, $1.25 for blue), and simple bars show the elapsed time in the experiment and the "game progress", a simple global quantity measuring the fraction of edges in the network with the same color on each end. This progress bar is primarily intended to make subjects aware that there is activity elsewhere in the network to promote attention, and is uninformative regarding the current majority choice.

The system logs fine-grained temporal data on the exact sequence of events in each experiment. This log contains every color-change event, along with vertex or subject number, the color selected, and a time stamp with 1 millisecond resolution. We also administer an exit questionnaires to each subject.

3.2.2 Human Subject Methodology

We give in this section some further details of our experimental and human subject methodologies. The human subject methodology that we used was approved by Penn's Institutional

Review Board process.

This chapter describes one session of experiments conducted in May 2008. The session consisted of 81 individual networked biased voting experiments, each lasted up to 1 minute in time. The session was held in a laboratory containing 38 workstations, of which 36 are used in the actual experiments with the other 2 as spare ones. This determined the number of subjects for each experiment as 36, who were recruited from a Penn undergraduate computer science class on a topic related to the experiments.

Each of the 81 experiments had a fixed network and incentive structure, and the system assigned each of the 36 subjects randomly to one of the 36 network positions at the start of each experiment, thus assuring there was no systematic bias in the position of subjects in the networks. To prevent the establishment of social conventions that could trivialize the experiments (such as all subjects playing red for the remainder of the session following a successful global consensus to red), the system used a local randomization scheme on the colors, which might make what appeared red to one player appear blue to another.

Prior to the main session of experiments, subjects attended a compulsory briefing session in which they were instructed about the networked biased voting problem, and the working of the system. We also ran a preliminary set of experiments to ensure the subjects' understanding of the game and the working of the software system.

Before the experiments, physical partitions were set up to prevent subjects from viewing other subjects or their screens. The whole session of experiments were carefully proctored to ensure that the only kind of communication that takes place between the subjects is through the experimental software platform. Each experiment ended either after 1 minute or when the 36 subjects successfully reached a unanimous consensus, whichever came first. At this point, the session proceeded to the next voting experiment. The whole session lasted between 2 and 3 hours.

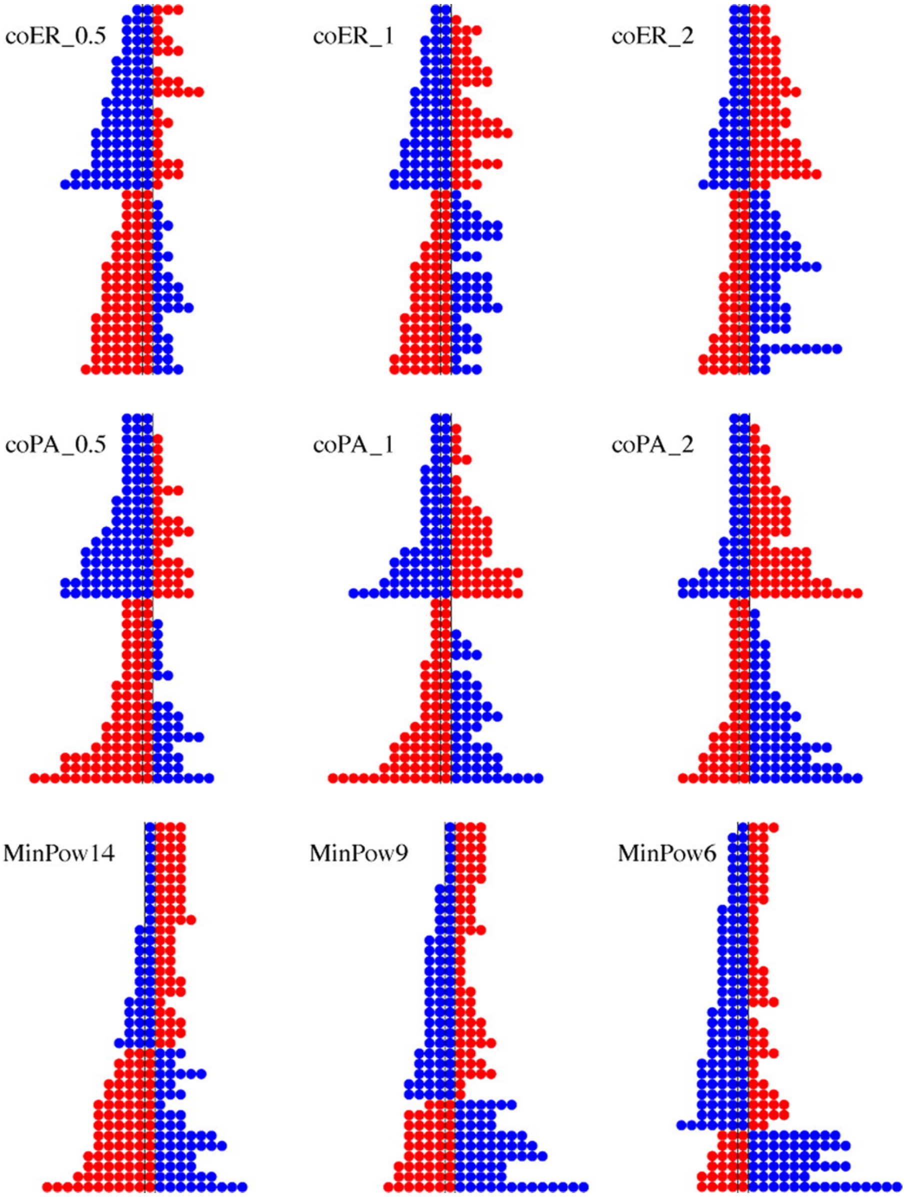


Figure 3.2: Visualization of network and incentive structures.

3.2.3 Experiment Design

There are two main design variables underlying our experiments: the connectivity structure of the underlying network and the financial incentives and their placement in the network. In each experiment, the network structure and the incentives were chosen in a coordinated fashion to examine specific scenarios or hypotheses. We now describe these choices and hypotheses in greater detail.

The 81 experiments fell into 2 broad categories that we call the *Cohesion experiments* (54 experiments) and the *Minority Power experiments* (27 experiments), named for the phenomena they were designed to investigate. All of the networks used had 36 vertices and nearly identical edge counts (101 ± 1), thus fixing edge density; only the arrangement of connectivity varied, and not the amount.

In the Cohesion experiments (named in part for a particular measure of inter- and intra-group connectivity [71]), vertices were divided into 2 groups of 18. Vertices in one group (the "red" group) were given incentives paying more for a red global consensus, whereas vertices in the other group (the "blue" group) were given incentives paying more for a blue global consensus. The relative strengths of these incentives were varied, as were the amount and nature of the connectivity within and between the two groups. In particular, we varied whether the typical vertex had more or fewer inter-group than intra-group edges, thus controlling whether local neighborhoods were comprised primarily of individuals with aligned incentives (high cohesion), competing incentives (low cohesion), or approximately balanced incentives. We also varied the nature of this connectivity; half of the Cohesion experiments used networks whose edges were generated (subject to the inter/intra group constraints) by a random or Erdös-Rényi process [13] (in which all edges are chosen randomly and independently with some fixed probability), the other half by the preferential attachment process [7] (which is known to generate the oft-observed power law distribution of connectivity). These two network formation models are well-studied and together provide significant variation over a number of common structural properties, including network diameter, degree distribution, and clustering.

The overarching goal of the Cohesion experiments was to systematically investigate how collective and individual performance and behavior varied with neighborhood diversity and the strength of preferences. Although it is perhaps most natural to hypothesize that increased inter-group connectivity should improve collective performance – this would be consistent with several mathematical network theories and metrics, including the aforementioned cohesion, and notions of expansion from the graph theory literature [13] – the degree of improvement, and how it might be influenced by the detailed structure (Erdös-Rényi vs. preferential attachment), the variability of individual human behavior, and so on, are difficult to predict.

In the Minority Power experiments, all networks were generated via preferential attachment [7]. A minority of the vertices with the highest degrees (number of neighbors) were then assigned incentives preferring red global consensus to blue, whereas the remaining majority were assigned incentives preferring blue global consensus. The size of the chosen minority was varied (6, 9, or 14), as were the relative strengths of preferences.

For each of the different network structures in the Cohesion and Minority Power families, we ran experiments in which there were "strong symmetric", "weak symmetric", and "asymmetric" incentive structures. By "symmetric" we mean that the incentives of those players preferring blue and those preferring red were symmetrically opposed (such as \$0.75/\$1.25 for consensus to red/blue vs. \$1.25/ \$0.75); by "weak" and "strong" we refer to the relative magnitudes of the preferred and non-preferred payments (\$1.25 to \$0.75 for weak, \$1.50 to \$0.50 for strong). In the asymmetric incentives experiments, the group preferring one color would be given strong incentives, whereas groups preferring the other color would be given weak incentives. We thus imposed scenarios in which 2 opposing groups "cared" equally but mildly about the global outcome, equally and strongly, or in which one group cared more than the other.

Fig. 3.2 shows a visualization of the network and incentive structures in our design. For each of the 9 network and incentive structures there is a diagram consisting of 36 rows of colored dots. Each row corresponds to a single subject or vertex in the network, and the dots in that row represent that subject and his or her network neighbors. The color of the central dot indicates the preferred (higher payoff) color for the corresponding subject, according to the incentives. The dots to the left of center indicate the number of neighboring subjects with the same preference; the dots to the right indicate the number with the opposite preference.

Vertices are ordered within groups by their overall degree.

The top three designs are Cohesion experiments with Erdös-Rényi connectivity in which there is more intra- than inter-group connectivity between the two groups (specifically a 1:2 inter:intra ratio) in the design on the left; balanced connectivity (1:1 ratio) in the design on the center; and more inter- than intra-group connectivity (2:1 inter:intra ratio) in the design on the right. This is demonstrated by the migration of dots from left of center to right of center as we move from column 1 to 2 to 3.

The middle row corresponds to Cohesion experiments with preferential attachment connectivity in the same inter:intra ratios as the coER row above. Comparison with the first row reveals clear differences in the overall degree distributions, because the variance in the total number of neighbors of subjects is much higher for preferential attachment and those diagrams reveal the presence of subjects with very large numbers of neighbors.

The bottom row corresponds to Minority Power experiments, where again we see the heavy-tailed degree distributions typical of preferential attachment but in which now the blue-preferring vertices are selected to be a minority of varying sizes (14, 9, and 6) with the highest degrees. Each of these 9 network structures was combined with payoff amounts that were weak symmetric, strong symmetric and asymmetric, yielding 27 distinct scenarios that were each executed in 3 trials, for a total of 81 experiments.

The overarching goal of the Minority Power experiments was to systematically investigate the influence that a small but well connected set of individuals could have on collective decision making – in particular, to investigate whether such a group could reliably cause their preferred outcome to hold globally and unanimously.

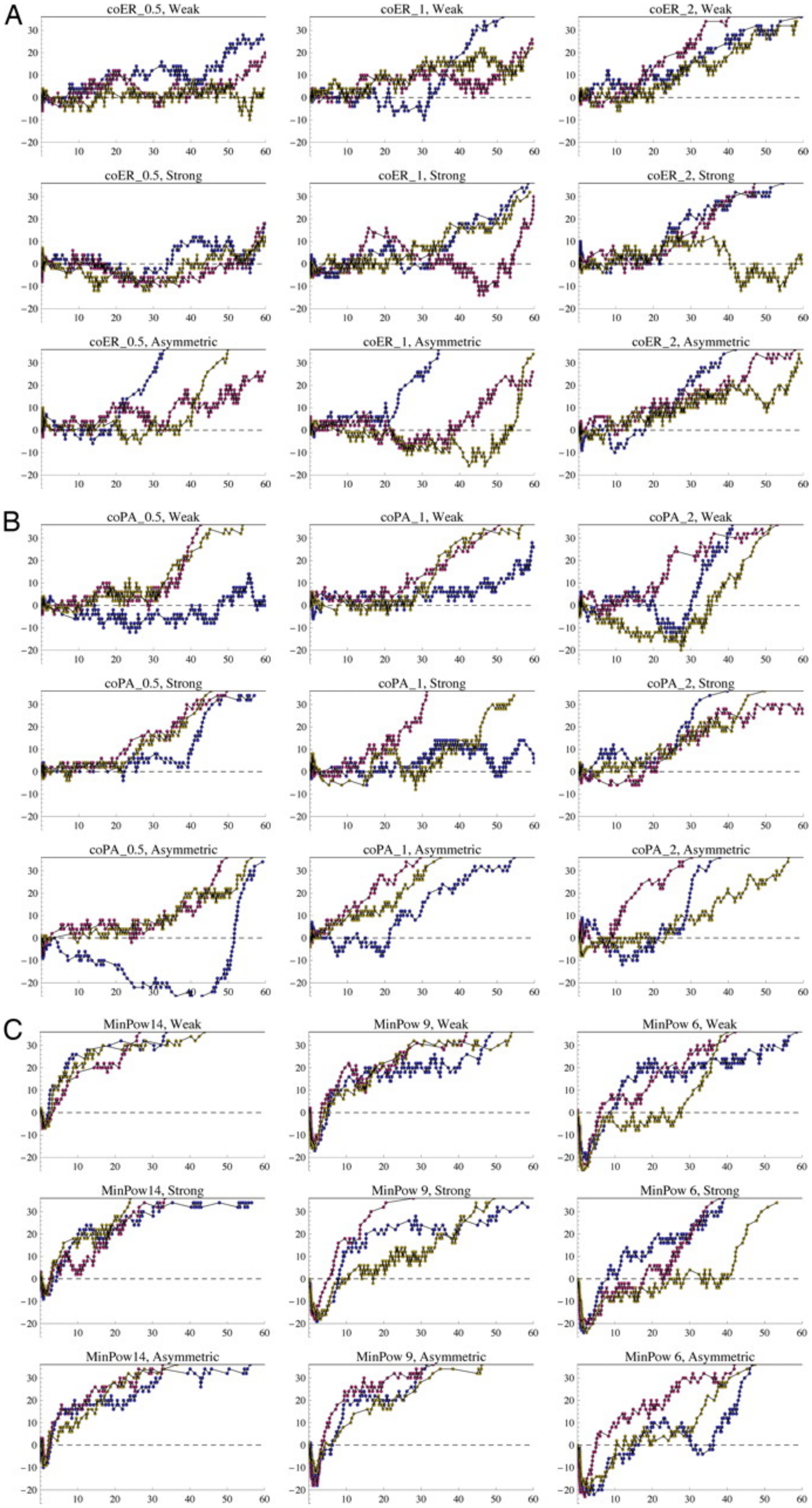


Figure 3.3: Visualization of the collective dynamics for all 81 experiments.

3.3 Results

3.3.1 Collective Behavior

Overall the subject population exhibited fairly strong collective performance. Of the 81 experiments, 55 ended in global consensus within 1 min (resulting in some payoff to all participants), with the mean completion time of the successful experiments being 43.9 s (standard deviation 9.6 s). We now proceed to describe more specific findings quantifying the impact of network structure, incentive schemes, and individual behavior. Network structure influenced collective performance in a variety of notable ways. The Cohesion experiments were considerably harder for the subjects than the Minority Power experiments; only 31 of 54 of the former were solved compared with 24 of 27 of the latter (difference significant at $P < 0.001$). Furthermore, in all 24 of the successfully completed Minority Power experiments, the global consensus reached was in fact the preferred color of the well-connected minority. Together these results suggest that not only can an influentially positioned minority group reliably override the majority preference, but that such a group can in fact facilitate global unity. Within the Cohesion experiments, generating connectivity according to preferential attachment (20/27 solved) yielded better collective performance than generating it via Erdös-Rényi (11/27 solved; difference significant at P $\approx$ 0.013). When combined with the high success rate of the preferential attachment Minority Power experiments (the difference between the 44/54 solved instances of all preferential attachment networks and the 11/27 solved Erdös-Rényi networks is significant at $P < 0.001$), this finding indicates that, for this class of consensus problems, preferential attachment connectivity may generally be easier for subjects than Erdös-Rényi connectivity, an interesting contrast to problems of social differentiation such as graph coloring [56], where preferential attachment networks appear to create behavioral difficulties. Independent of the method for generating connectivity, Cohesion performance improved systematically as within-group connectivity was replaced by between-group connectivity, with the strongest performance coming from Cohesion networks in which most subjects might have a preferred color different from those of a majority of their neighbors. Across all Cohesion experiments, the success rate on the networks with the highest level of inter-group connectivity (14/18 solved) and the success rate when connectivity was either mainly intra-group or balanced (17/36 solved) are significantly different ($P < 0.03$). Thus, increased awareness of the presence of opposing preferences improves social welfare. In terms of behavioral collective dynamics, it appears that this awareness leads to early "experimentation" with subjects' non-preferred colors, resulting in more rapid mixing of the population choices. Across all network structures, asymmetric incentives yielded the strongest collective performance (the overall asymmetric success rate of 22/27 differs from the combined weak/strong symmetric success rate of 33/54 at $P < 0.05$), and, indeed, the extremist's preferences were dominant, determining the consensus outcome in 18 of the 22 successful asymmetric experiments. Strong symmetric incentives (14/27 successes) yielded worse performance than weak symmetric ones (19/27 successes). Thus, it appears most beneficial to have extremists present in a relatively indifferent population, and most harmful to have 2 opposing extremist groups.

The results on collective behavior described so far have focused on the final outcomes of experiments. The collective dynamics within individual experiments is also revealing, and shows notable effects of network structure.

In Fig. 3.3 we provide visualizations of the collective dynamics in each of the 81 experiments, grouped by network structure and incentive scheme. For each network and incentive structure there is a set of axes with 3 plots corresponding to the three trials of those structures. Each plot shows the number of players choosing the eventual collective consensus or majority color minus the number of players choosing the opposite color (y axis) at each moment of time in the experiment (x axis). All plots start at 0 before any color choices have been made;plots reaching a value of 36 within 60 s are those that succeeded in reaching unanimous consensus. Negative values indicate moments where the current majority color is the opposite of its eventual value. Plots are grouped by network structure first (Cohesion experiments with Erdös-Rényi connectivity in A; Cohesion experiments with preferential attachment connectivity in B; Minority Power experiments in C), and then labelled with details on the network and incentive structure. Within the Cohesion experiments, intergroup connectivity increases from left to right; within the Minority Power experiments, the minority size is decreasing from left to right. Several distinctive effects of network structure on the dynamics can be observed. Many Cohesion experiments spend a significant period "wandering" far from the eventual consensus solution. In contrast, Minority Power experiments invariably experience an initial rush into negative territory as the majority select their preferred color, but are then quickly influenced by the well-connected minority. Several instances of rather sudden convergence to the final color can also be seen, even after long periods of near consensus to the opposite color (e.g., blue plot in lower left corner axes of *B* at about 50 s). Fig. 3.4 below provides a visual summary of some of the qualitative effects of network structure on these dynamics.

Notable features include a ritual initial flurry of activity away from the minority preference in the Minority Power experiments, followed by an inevitable assertion of the minority influence over the population. There are also many instances in which a significant fraction of the experiment is spent quite far away from the eventual consensus choice, including near-total reversals of the collectively chosen color; see Fig. 3.3 and for further details.

Although these visualizations of the dynamics are rich in detail, it is difficult to extract meaningful structural effects from them. In Fig. 3.4 we thus show the results of fitting simple 2-segment random walk models to the experimental dynamics within each family of experiments (fixed network and incentive structure).

For each of the 81 individual experimental plots in Fig. 3.3, we fit a 2-segment random walk model to the data – one segment for the first 20 s of the experiment, and one for the remainder of the experiment (similar findings result from different cut points between the two segments). Within each segment, we simply compute the fraction p of "upwards" moves

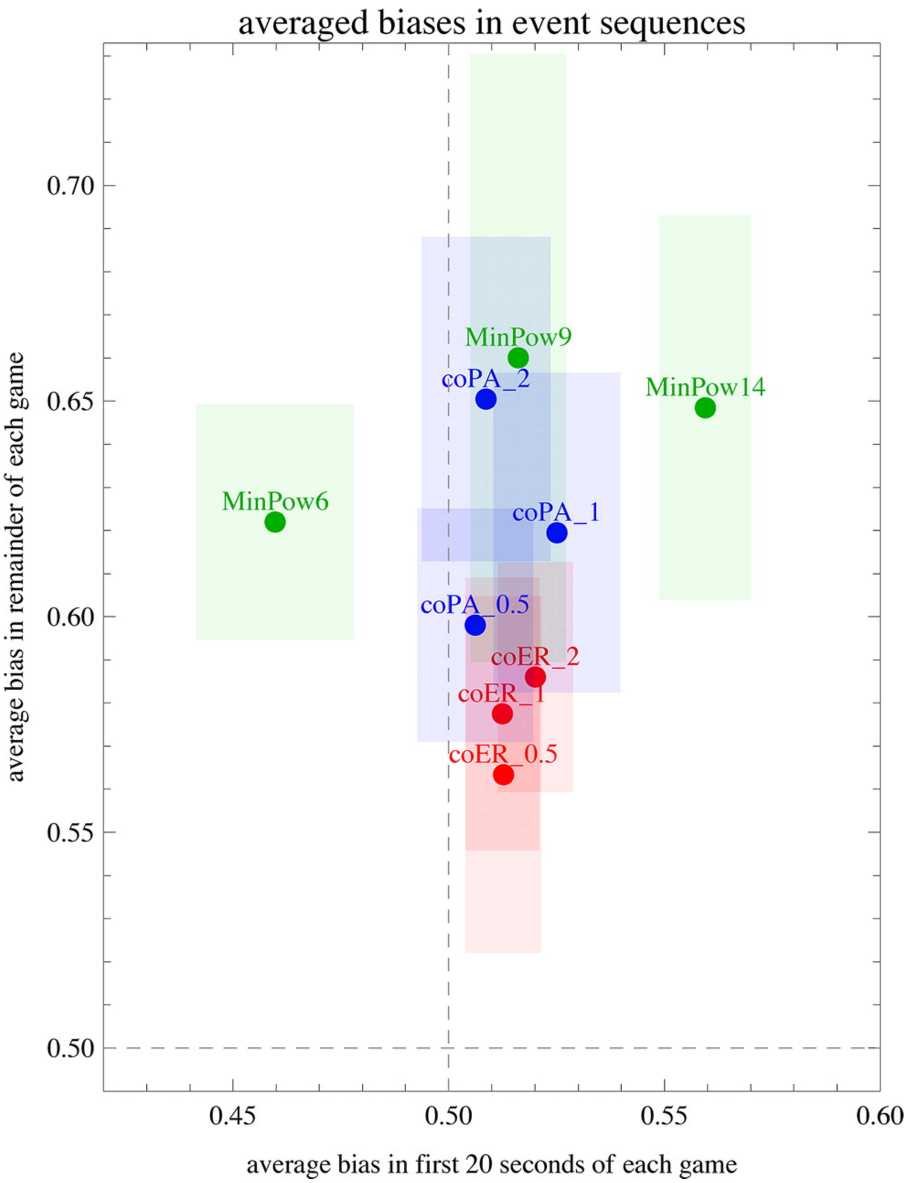


Figure 3.4: Visualization of biased random-walk model fits to the dynamics of Fig. 3.3.

(the number of moves toward the eventual majority color, divided by the total number of moves within the segment). This can be interpreted as modeling the collective dynamics by a random walk with probabilities p and $1-p$ of upwards and downward moves, respectively; we refer to p as the *bias* of the model. Permitting independent bias values in the two segments allows us to separately model the dynamics in the early and later portions of each experiment. This yields a 2-parameter model for each of the 81 plots. Above we show the result of averaging over all incentive schemes and all repeated trials within the 9 families of network structures (Cohesion with Erdös-Rényi connectivity and 3 settings of inter- vs. intra-group connectivity; Cohesion with preferential attachment connectivity in 3 inter- vs. intra- settings; and Minority Power with 3 different minority group sizes). For each of these 9 families, we plot a point showing the average bias in the two segments, along with a shaded rectangle delimiting the standard deviation in both bias parameters for that family. The dashed lines show $p = 0.5$, where the model is unbiased (equal upward and downward probability). Several qualitative effects of network structure are apparent. For instance, Cohesion experiments tend to begin slowly (bias only slightly larger than 0.5), but preferential attachment connectivity leads to more rapid convergence in the later portion than does Erdös-Rényi connectivity. Increasing inter-group connectivity speeds the later dynamics regardless of the connectivity type. Minority Power experiments tend to conclude rapidly, but their early dynamics are strongly dependent on the minority size, with smaller minorities slowing the early progress toward the eventual majority choice. When the minority size is only 6, the first 20 seconds typically have a downward drift (bias $p < 0.5$).

These models clearly show the effects of structure on collective dynamics: In terms of the rate of approach to the eventually favored color, Cohesion experiments with Erdös-Rényi connectivity tend to both begin and end slowly, whereas those with preferential attachment connectivity begin slowly but end more rapidly. Higher inter-group connectivity consistently increased late-game speed toward consensus. The Minority Power dynamics ended relatively fast, but early speed was heavily influenced by the size of their minorities.

3.3.2 Individual Behavior

It is natural to investigate the extent to which different human subjects exhibited distinct strategies or styles of play across the experimental session, and the degree to which such stylistic differences did or did not influence individual earnings. For any measure M of individual subject behavior within an experiment (such as the number of color changes

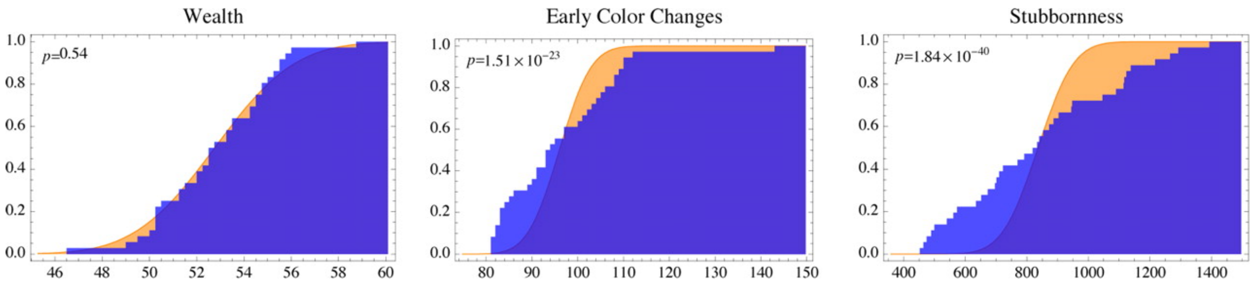


Figure 3.5: Illustration of the "random observer" method for detecting meaningful variation in subject behavior.

made by the subject), we can compute the 36 average values for M obtained by taking the 81-game average for each subject, and compare these to the distribution of "random observer" averages, obtained by picking a random subject to observe in each experiment, and averaging the resulting 81 M values. Because subjects were in fact randomly assigned their network positions and incentives at the start of each experiment, if the variance of the 36 actual subject averages significantly exceeds that of the random observer distribution (according to a standard variance test), we can conclude that subjects exhibited meaningful (greater than chance) variation on measure M.

Fig. 3.5 is an illustration of the "random observer" method for detecting meaningful variation in subject behavior. The figure on the left shows in blue empirical cumulative distribution function (CDF) of total player wealth, in which wealth (x axis) is plotted against the fraction of the 36 subjects earning at least that amount (y axis). It is very well-modeled by the theoretical expected CDF generated by choosing a random player's wealth independently in each experiment (which is shown in orange), so we may conclude that the variation in player wealth is explained by the random assignments to network position. In contrast, the CDFs of the number of color changes taken by each player in the first several seconds (the middle figure) and the total amount of "stubborn" time (the right figure) are poorly modeled by the random observer CDF, showing considerably greater variance in both cases.

We note that it is particularly noteworthy that when the measure is wealth, subjects did not exhibit meaningful variation – thus the disparity in average or total wealth across the session (which ranged from \$46.50 total earnings to \$58.75, with a mean of \$52.76 and standard deviation of \$2.46) is already well-explained by the random assignment of subjects to positions. However, this finding in no way precludes the possibility that subjects still display distinct "personalities", nor that these differences might strongly correlate with final wealth. For instance, subject "stubbornness" – as measured by the amount of time a subject is playing their preferred color, but is the minority color in their neighborhood – varies meaningfully (Fig. 3.5) and is positively correlated with average wealth (correlation coefficient ≈ 0.43, $P < 0.01$). Being stubborn at the outset of an experiment (during the first 9 s) shows even stronger correlation with wealth (correlation coefficient ≈ 0.55, $P < 0.001$). The number of color changes made by subjects in the opening seconds of an experiment also varies significantly (Fig. 3.5) and is strongly negatively correlated with wealth (-0.58, $P < 0.001$). Together, these results suggest that stubborn and stable players set the tone of an experiment early.

Player stubbornness warrants further investigation, because it strikes at the heart of the tension that is a focal point of the experiments – by being stubborn, one might improve the chances of swaying the population toward one's preferred color, but one also risks preventing global consensus being reached in time (and thus forgoing any payoff). It is clear that no subject was infinitely stubborn: The wealthiest player had their preferred color 28 times out of 55 successful games but acquiesced to group dynamics and accepted the lower payoff 27 times. All other players acquiesced more often – up to as many as 40 times out of 55. In the 26 games that failed to achieve unanimity, there were only 30 individual cases of players defying all of their neighbors as time expired, and only 5 games ended in failure due to players that defied all neighbors for more than the last 2 seconds of play. Only 3 individual players ever caused this kind of failure; one did it 3 times, but also acquiesced 38 out of 55 times and garnered relatively poor overall earnings. These facts combined with the aforementioned correlation of stubbornness with wealth suggest that successful players managed to be “tastefully” stubborn, and that overall behavior was quite acquiescent.

In addition to the raw experimental data, subjects were given an exit survey in which they were invited to comment on their own and others’ strategies, and these surveys provide a rich and often consistent source of insight into individual styles of play. Twenty four subjects explicitly mentioned starting off by choosing the color that would give them the higher payoff upon consensus. Twenty seven subjects mentioned either trying to signal others, or noticing others trying to signal; however, many also found this behavior annoying and said that it did not help. Twenty-one subjects noticed others being irrationally stubborn, or expressed suspicion that others were being irrationally stubborn. (Here we use the term “stubborn” in the informal way it was given in the surveys, as opposed to the formal measure discussed above.) Three subjects mentioned being stubborn themselves because they did not want small payoffs. Seven subjects mentioned using different strategies depending on whether their incentives were weak ($0.75 vs. $1.25) or strong ($0.50 vs. $1.50). Three subjects mentioned changing their behavior as the night progressed, 1 subject developed a strategy, and 2 subjects simply became tired. We note that there is no evidence in the data of the collective performance improving or degrading significantly as the session progressed; for instance, plotting the accumulated collective wealth vs. the progression of experiments in the order they were conducted yields an almost perfectly linear curve.

Finally, 27 subjects mentioned following the action choices of their high degree neighbors and/or being more stubborn when they themselves had high degree. It is interesting to note that the average degree of subjects is much more weakly correlated with their wealth (0.38, $P \approx 0.09$) than the stubbornness and stability properties discussed above, despite these reports of conditioning behavior on degrees. There is no inherent contradiction here, because conditioning on degrees may appear primarily in the decision on how stubborn and stable to play.

Despite the observed and reported variations in individual subject strategies, it is interesting that one can approximately reproduce salient aspects of the collective behavior with rather simple and homogenous theoretical models of individual behavior. For example, consider a "multiplicative" model in which a player who is paid $w(c)$ for global convergence to color c, and a fraction $f(c)$ of whose neighbors are currently playing c, plays c in the next time step with probability proportional to $w(c)f(c)$ [60]. Such agents combine their preferences (as given by the values $w(c)$) with the current trend in their neighborhoods (the $f(c)$) to stochastically select their next color in a natural manner. If such agents are simulated using the same networks and incentives as in the 81 human subject experiments, and the number of simulation steps is capped (as it effectively is by the 1-min time limit of the human experiments), there is rather strong correlation (0.60, $P < 0.001$) between subject and simulation times to consensus.

3.4 Discussion

A number of further investigations are suggested by the findings summarized here. In particular, the variations in individual behavior and the apparently helpful presence of "extremists" raise the question of whether certain mixtures of behaviors and attitudes are required for optimal collective problem-solving. It would also be interesting to use the data from our experiments to develop richer statistical models of individual and population behavior, whose predictions in turn could be tested on further behavioral experiments.

Chapter 4

A Behavioral Study on Bargaining in Networks

4.1 Introduction

In this chapter, we continue our effort in bringing behavioral experiments to the study of games in networks. We report on a series of highly controlled human subject experiments in *networked bargaining*.

Networked bargaining is modelled as follows: players are the nodes of the networks, and each edge in the network represents some fixed amount of money that can be realized by its endpoints if they agree on how to split the amount. This agreement shall be referred to as *closing a deal*. In addition, there is a *deal limit* on each node, which is the maximum number of deals a player at that node is allowed to close, which could be less than its degree.

We were partly inspired by a long line of previous theoretical work which tried to relate wealth to network topology in bargaining settings [27, 35, 10, 69, 15, 63, 6, 20]. A notable feature of these theories is the prediction that there may be significant local variation in splits purely as a result of the imposed deal limits and structural asymmetries in the network. One can view our experiments as a test of human subjects' actual behavior at this game in a distributed setting using only local information. Our experiments are among the first and largest behavioral experiments on network effects in bargaining conducted to date.

As we did for the behavioral experiments in networked biased voting described in Chapter 3, we adopt many of the practices of behavioral game theory, which has tended to focus on two-player or small-population games rather than larger networked settings. In each of our experiments, three dozen human subjects simultaneously engage in one-to-one bargaining with partners defined by an exogenously imposed network. Our work continues a broader line of research in behavioral games on networks at the University of Pennsylvania[56, 48, 53]. Closest in spirit to the current work is that investigating networked exchange economies [48], but the experiments here and the theories underlying networked bargaining differ significantly from networked trading models.

In an extensive and diverse series of behavioral experiments, and the analysis of the resulting data, we address a wide range of fundamental questions, including: the relationships between degree, deal limits, and wealth; the effects of network topology on collective and individual performance; the effects of degree and deal limits on various notions of "bargaining power"; notions of "fairness" in deal splits; and many other topics.

The networks used are inspired from common models in social network theory, including preferential attachment graphs, and some specifically-tailored structures.

In all our experiments, the number of deals that were closed was above 85% of the maximum possible number. This is high enough to demonstrate real engagement, and low enough to demonstrate real tension in the designs.

Most of the deeper findings can be related to existing network bargaining theory. Although deals are often struck with unequal shares, more than one-third of the deals are equally shared, thus indicating that people, while behaving as self-interested actors, also have an aversion towards inequality.

Network topologies have enough of an effect that they can be distinguished statistically via individual wealth levels and other measures. Higher degree, for example, tends to raise bargaining power while higher deal limits tend to decrease it. But while local topology affects bargains, invisible competition also affects it, even when the local topologies are indistinguishable. We find the expected effects of higher deal limits in the first neighborhood and higher degrees in the first and second neighborhoods, but neither degree distribution nor deal limit distribution is sufficient to determine the inequality of splits. In sum, there is a rich interaction between network and wealth that needs more study.

Other findings that speak to no existing theories but might provoke some new ones are the following:

- There is a positive correlation between inequality and social efficiency.
- Failures to agree on a split (as opposed to failures to find the best global trade configuration) form the greater part of missing efficiency.
- Social efficiency was higher when some uncertainty existed about a partner's costs.

Finally, there are two curios that seem more about psychological dynamics than economics: People who are patient bargainers tend to make more money; and an incidental asymmetry in our protocol for closing a deal is correlated with a bias in the split.

In the ensuing sections, we review relevant networked bargaining theories, describe our experimental design and system, and present our results.

4.2 Background

Networked bargaining with deal limits on the nodes, also known in the sociology literature as networked exchange with substitutable or negatively connected relations (eg. [15]), has been studied for decades. Several theoretical models have been designed to predict or propose how wealth should be divided [35, 69, 27, 82], and human subject experiments have been conducted on a few small graphs (up to 6 nodes) [25, 26, 82], albeit with different interfaces and mechanisms than ours. Some of the theoretical models are based on limited experimentation, along with simulated human behavior on slightly larger graphs [26]. A few models are based strongly on notions of game-theoretic rationality and are natural extensions of standard economic literature to social networks. Two models that belong to this class were introduced by Cook and Yamagishi [27] and by Braun and Gautschi [15]. We shall mainly focus on these two models.

The model given by Cook and Yamagishi, sometimes referred to as equidependence theory, is the most recognized theoretical model, and has received a lot of recent focus from the theoretical computer science community [63, 6]. Though Cook and Yamagishi[27] considered only unique exchange networks (that is, where each vertex may close only a single deal), the model is easily extendable to networks with varying deal limits. Every node is assumed to play strategically with selfish game-theoretic rationality. An *outcome* describes the division of wealth on various edges of the network. The *outside option* of a node is the highest offer it can rationally receive from any of its neighbors, such that closing that deal would benefit both parties, compared to the given state. An outcome is said to be *stable* if every player's earning is more than its outside option. Game-theoretic rationale suggests that an outcome should be stable if the players act in a myopically selfish manner. Cook and Yamagishi propose that the achieved outcome must be stable; moreover, they propose that the achieved outcome should be *balanced*, that is, two parties that close a deal should have equal additional benefit from this edge, where additional benefit is measured as the amount by which the earning of a player exceeds its outside option. Kleinberg and Tardos [63] showed that a stable and balanced outcome exists on all bipartite networks, but may not exist in all networks, and if it does, the closed deals in a stable outcome form a maximum matching. This equal division of surplus is stipulated by standard two-player bargaining solutions such as the Nash Bargaining Solution and Proportional Bargaining Solution, for players with linear utilities [75, 11].

Though a balanced outcome seems to be the most robust theoretical model, it has several drawbacks, first and foremost that it does not exist on even simple networks such as a triangle; and when it exists, there is a balanced outcome for every maximum matching in the network. This makes it computationally hard to even enumerate all the balanced outcomes in a network, and non-uniqueness reduces the predictive value of such a model. Another drawback is that the model often suggests that some edges will be shared so that one party gets an infinitesimal share, and the other party gets practically the entire amount. For example, a node that has at least two leaves (nodes of degree 1) as neighbors always ends up with maximum possible profit, due to competition between the leaves. However, even previous small-scale experiments [82] have suggested that such a phenomenon does not happen, and in our experiments, players rarely close a deal that extremely favors one of the players. Thus, when human subjects are involved, perfect local rationality seems to be an incorrect assumption.

The model given by Braun and Gautschi [15] defines a “bargaining power function” on nodes that depends only on the degree of the node and degrees of its neighbors. This function increases with increase in degree of the node, and decreases with increase in degrees of its neighbors, and is independent of all other network aspects. On each edge, the division of wealth, if a deal is made, is stipulated to be proportional to the bargaining power of the adjacent nodes. The bargaining power functions do not distinguish between different limits on nodes, and generally assume that relations are negatively connected: that is, for any given player, closing one deal reduces the maximum value that can be obtained from other edges deals. This makes the model quite inadequate as a predictor for our experiments. The other feature of this model is that network effects are quite local in nature, since even slightly distant properties such as the degrees of neighbors of neighbors do not have an effect on the bargaining power function. However, the model attempts to capture the notion that the earning of a player depends positively on its own degree and negatively on the degree of its neighbors. We test this notion on fairly large graphs for the first time, and we also show that the degrees of neighbors of neighbors do affect a node positively. Such alternating effects were predicted in previous theoretical models such as that by Markovsky et. al. [69], which said that odd length paths from a node enhance its earning, while even length paths reduce it.

The most significant set of previous experiments were done by Skvoretz and Willer [82], who conducted experiments on 6 small networks (each has at most 6 nodes), with only unit deal limits in 4 of them. They found that some common intuitions held true in those networks. For example, players who have deal limit one and multiple leaves as neighbors gets the bigger fraction of a closed deal, and that this fraction reduces if the limit of the player is raised. Among other results, we test such hypotheses extensively on much larger graphs with much more variance in their degree and limit distributions, and establish these hypotheses with very high statistical significance. Larger graphs also allow us to study the effects of network topology aspects that are more involved than the degree or limit of the player.

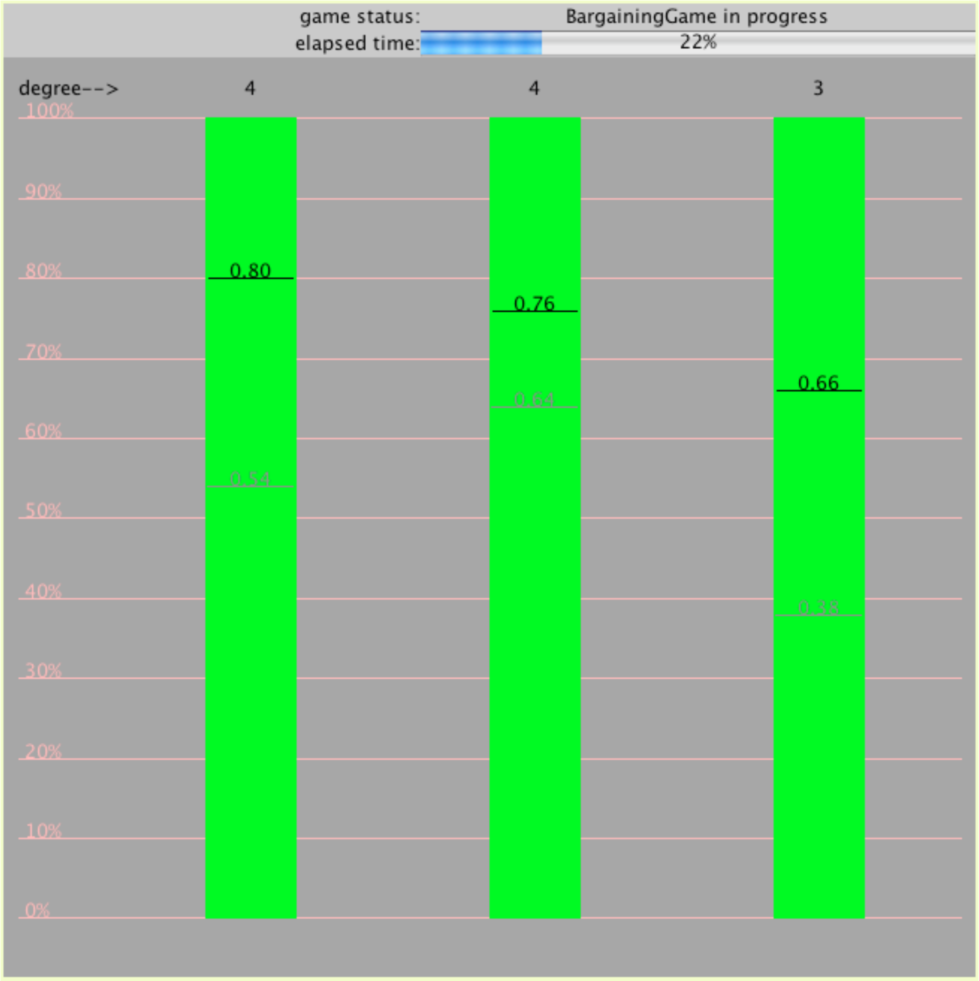
Recently, Chakraborty et. al. [20] designed an extension of the Cook-Yamagishi model, in the setting where there are no limits on the number of deals. In this case, the model predicts that all deals should be closed, and if players have linear utility (which is assumed in the Cook-Yamagishi model), all deals should be shared equally. Unequal splits may occur only if players have non-linear utility.

4.3 Experiment Design Overview

Although we mostly report the results of one session of experiments, we actually ran networked bargaining games in 2 sessions, and one of them had 2 parts.

The first session entirely consisted of networked bargaining games where there was no limit on the number of deals a vertex can close (in other words, a player at a vertex can close deals on all edges incident on it). The first part of the first session involved no cost for closing deals. The second part of the first session had a cost, specific to each vertex of the network, that the player had to pay for each closed deal. We refer to these two settings as *basic setting* and *cost setting* respectively.

The second session, which we refer to as the main session in the sequel, is the focus of this chapter. In this session, there were never any costs involved, but there was a limit on the number of deals a particular vertex could close. We refer to this setting as the *limit setting*. Note that a game in the basic setting can be viewed as a game in the limit setting, but only if the limit of every node is equal to its degree.


game status: BargainingGame in progress

elapsed time: 22%

Figure 4.1: Screenshot of player's interface for bargaining.

4.3.1 Basic Setting

Fig. 4.1 shows a screen shot of the game interface used by each subject in our experiments in the basic setting. The elements of it will be described roughly from top to bottom. The **game status** shows "pending", "in progress", or "completed", which helps the player stay synchronized as a series of different games start, stop, and change. The **elapsed time** shows the fraction of a minute (the length of each individual experiment) that has elapsed since active play started.

The large section in the middle contains multiple elements. First, **neighbor bands** are green vertical bands that show data about individual neighboring vertices or players in the network; this example has 3 such vertical bands, indicating the presence of 3 neighboring bargaining partners.

In the top line is a row of degree numbers, which indicate the number of *other* neighbors each one of the user's neighbors has in addition to himself.

Within the green band there are 2 pieces of data: the offer asked by the user from the neighbor, and the counter-offer made by the neighbor to the user (the fraction that the user would receive if he agrees to the neighbor's offer). Each offer is shown as a horizontal bar in the green band, with a number above it, ranging from 0.00 to 1.00, indicating the fraction of the offer. The user can move the offer bar, which appears darker than the counter-offer bar, which changes only upon the action of the neighbor, and simply conveys information to the user.

The user can alter her offer to a particular neighbor by mouse-clicking on the green band. The range of offers is divided into 50 equal parts, so that every offer must be a multiple of 2 %. Initially, the offer bar is set to the highest point in the neighbor band, that is, each user is assumed to be demanding the entire value of the deal. Moving the mouse over the green band also indicates the offer that would be made if the user clicks at that point, so that the user can easily determine the place to click on, to make a particular offer.

Closing of a Deal When a player makes an offer to a neighbor, and the neighbor matches the offer (which is accomplished by making the offer bar coincide with the counter-offer bar), then the deal is ready to be closed. We refer to the former player as the *proposer*, while the neighbor who matches the offer is called the *acceptor*. A button labelled CLOSE now appears on the screen of the proposer below the appropriate neighbor band, while a wait

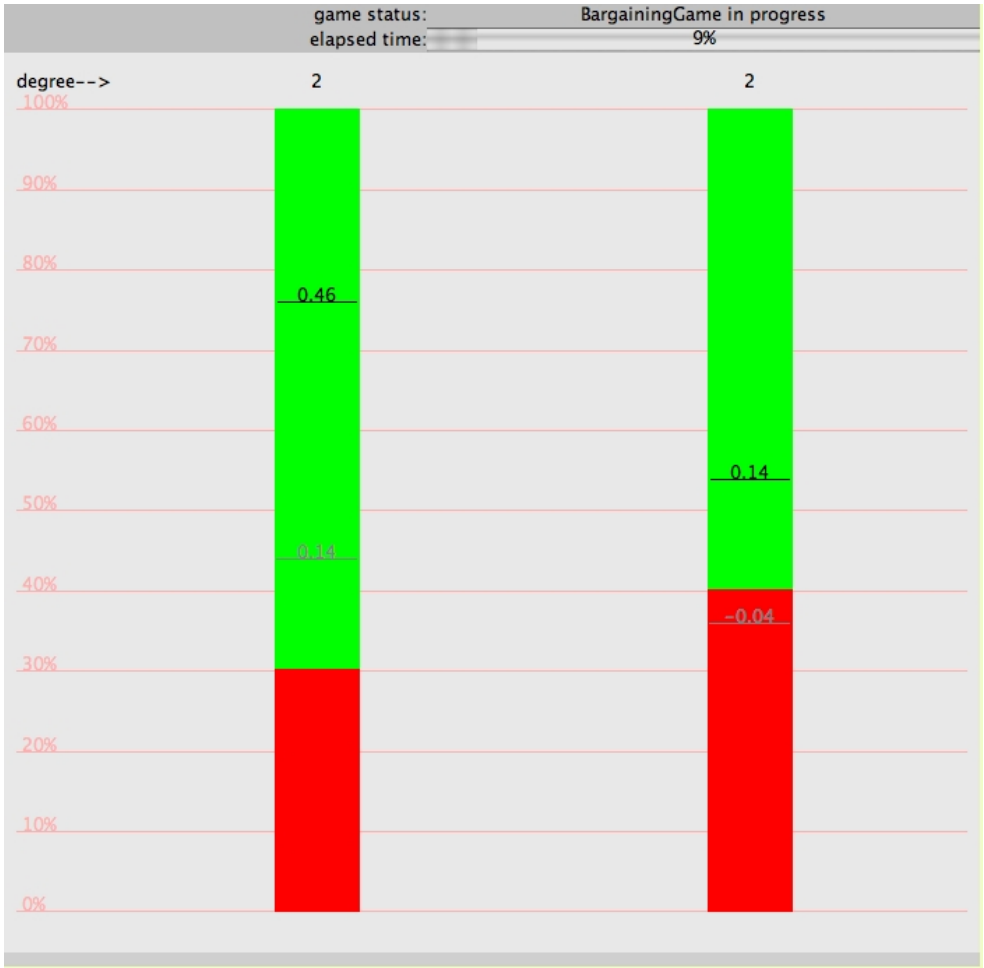
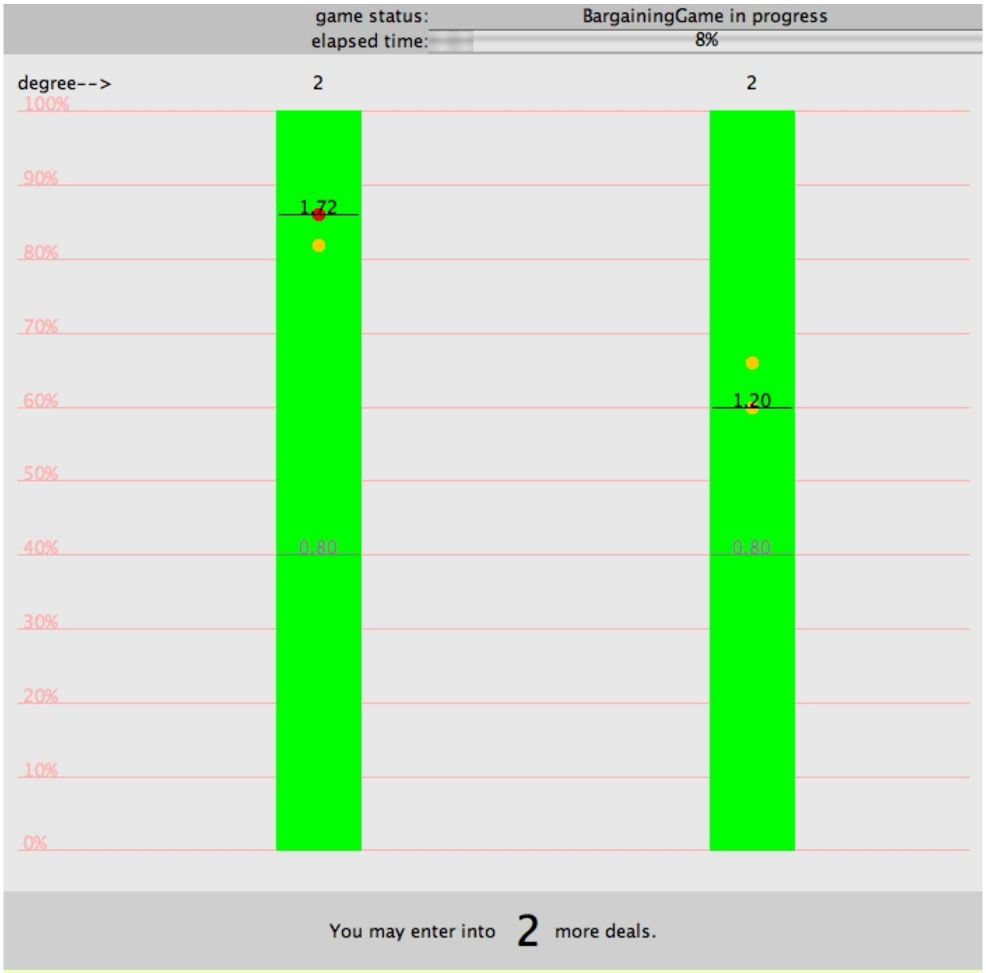

game status: BargainingGame in progress
elapsed time: 9%

Figure 4.2: Screenshot of player's interface for bargaining in the cost setting.

sign appears on the screen of the acceptor below. The proposer can now close the deal by clicking on the CLOSE button. Once the deal is closed, the money on that edge is realized and split according to the offer that both parties agreed upon, with no negotiations possible in the future. The neighbor bands on both users' interfaces corresponding to this deal then switches color to a faded green, and the offer bars become immovable. The bands also get labelled as "closed".

If either party changes his offer away from the agreed split before the CLOSE button is pushed, then negotiations continue.



game status: BargainingGame in progress

elapsed time: 8%

You may enter into 2 more deals.

Figure 4.3: Screenshot of player's interface for bargaining in the limit setting.

4.3.2 Cost Setting

The interface in the cost setting (Figure 4.2 shows a screen shot) had all the features as that in the basic setting, as well as a section at the bottom of the green band that could be colored red. The height of the red band indicates the cost of closing the deal for that user. Since there is only one cost for every vertex, the height of the red part is the same on all neighbor bands in a single user's interface. Note that a player can actually lose money in this game, if he closes a deal where the offer bar lies in the his region.

4.3.3 Limit Setting

The interface in the limit setting (Figure 4.3 shows a screen shot) had all the features as that in the basic setting, and some added features related to the introduction of limits. Because fewer deals could be made due to the limits compared to the first session, we increased the value of each deal to 2 dollars. At the bottom of the screen, a message tells the user how many more deals she can close. This number is initially equal to the limit of that vertex, and is reduced by one whenever this user closes a deal. We call this number the *residual limit*. A neighbor band is active only if both the user and the corresponding neighbor have positive residual limit, else the band is faded out and is labelled "voided". Further, there is more information on the neighbor band. On each neighbor band, there are dots indicating the offers that the respective neighbor is receiving at that time from all its neighbors who have positive residual limit. Offers on closed deals are not shown. This information is valuable, since the user is in a sense competing with these offers. If a particular neighbor has residual limit k at some point of time, then intuitively, at that time, the neighbor can be expected to close its top k offers. Thus the top k offers received by the neighbor is shown in the corresponding neighbor band as red dots, while the remaining offers are shown in black. Intuitively, the user should place her offer so that it lies among the red ones, if she wants to get considered for a deal closure by that neighbor.

As mentioned the main session of experiment that we focus on in this chapter is the one of *limit setting*, and we go on the describe it in detail in the next section.

4.4 Experimental Design of the Main Session

For our main session of experiments, we designed 18 different experimental scenarios (consisting of specific choices of networks and arrangement of deal limits; each such scenario received 3 trials, for a total of 54 short experiments). These scenarios were based on 8 different graphs with a wide variety of details. The sole property they share is that they all have

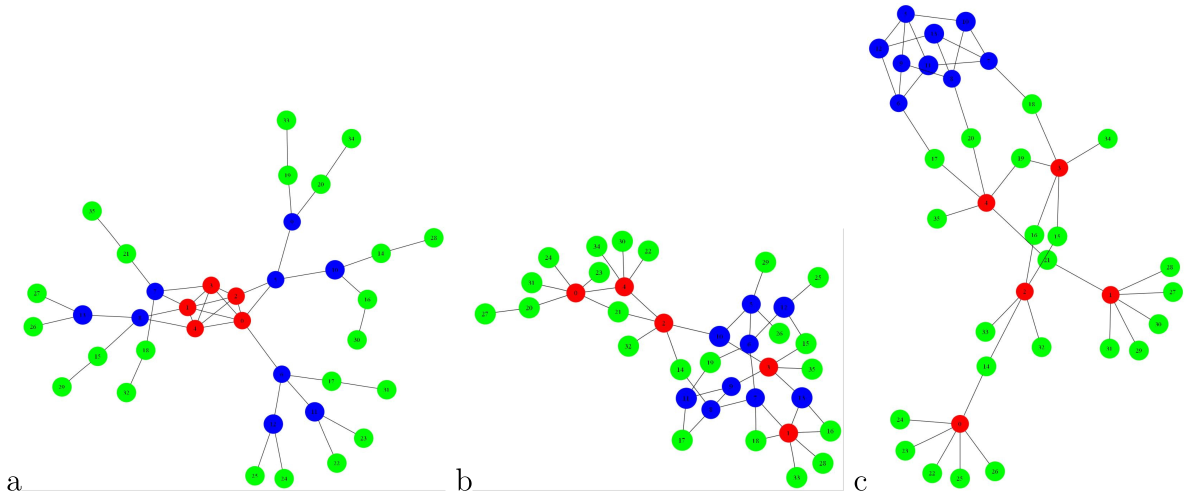


Figure 4.4: (a) PLP network with with $LOH = 0.9294$. (b) PL0 network with $LOH = 0.0013$. (c) PLN network with $LOH = -0.8461$.

36 nodes. We are thus casting our experimental nets wide here regarding network topology, as in much of our previous behavioral work. This section describes all the scenarios, at least at a high level.

The networks fall into 2 categories: regular graphs (to isolate and explore the effects of variations in deal limits), and irregular graphs (which contain an assortment of different degrees).

4.4.1 Irregular Graphs

We were interested in how bargaining behavior changes with changes in local network structure, and especially with differences in degree. Out of the huge space of such networks, we chose four. The first three we describe all had a common degree sequence, but differed in the way that nodes of each degree connected to nodes of other degrees. We generated a single degree sequence with a distribution that approximately follows a power law, and used it to build three graphs with different patterns of degree-to-degree profiles. We refer to these graphs as PL (for Power Law) graphs.

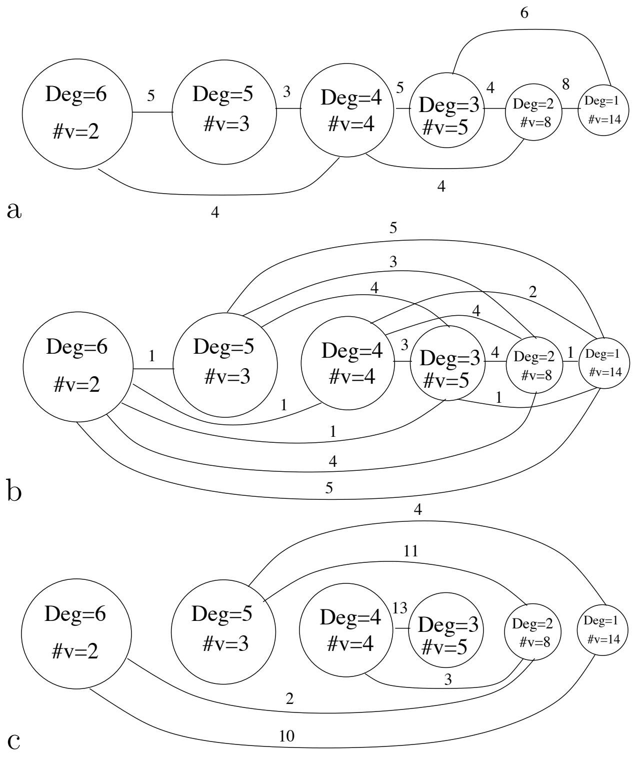


Figure 4.5: Schematic for a network with (a) positive LOH, (b) zero LOH, and (c) negative LOH

Power Law Graphs

Since we suspected that degree might have a large influence on bargaining power (to be confirmed below), it matters to the success of any node what the degrees are of other nodes they need to bargain with. Hence it was important to manipulate the *degrees of neighbors* as well.

By connecting nodes in different ways, we generated 3 graphs that differ in this manner but have the same basic degree distribution (each has 2 nodes of degree 6, 3 of degree 5, 4 of degree 4, 5 of degree 3, 8 of degree 2 and 14 of degree 1). The difference in connection pattern across these 3 graphs is measured by what we call *level of homophily* (LOH), and it is defined as follows: For each of the edges in a graph, give it a pair of numbers being the degree of its two ends arranged with the smaller one first. The correlation between the pairs is the LOH of the graph.

- **PL with positive LOH (PLP).** This network has a large variance in degree distribution but low variance within each neighborhood. This models a 'segregated' world. No edge bridges two nodes whose degrees differ by more than 2. On the other hand, the graph is highly connected in the sense that there are considerable connectivity between adjacent classes and within high-degree classes.

  PLP is shown in Fig. 4.4a and it has a level of homophily of 0.9294. A schematic of this network is shown in Fig. 4.5a, where nodes of the same degree are clustered into a single node, and numbers on the links indicate the count of edges across the two clusters in the original network.

- **PL with zero LOH (PL0).** PL0 has the same degree distribution as PLP but has $LOH = 0.0013$. This network is shown in Fig. 4.4b and a schematic of it is shown in 4.5b. PL0 models a world where nodes of different degrees mingle freely with each other.

- **PL with negative LOH (PLN).** PLN has the same degree distribution as PLP and PL0, but has $LOH = -0.8461$. This network is shown in Fig. 4.4c and a schematic of it is shown in 4.5c. PLN models a world where the poor are likely to be 'captivated' by the connection-rich.

With each of the PL graphs above, we used each of the following 3 deal limit schemes to obtain $3 \times 3 = 9$ different scenarios. The first is the well-studied unique exchange situation (uniq): all nodes have deal limit 1. The other 2 are neither unique exchange nor unlimited, but represent two points in another large space of possibilities in between those notions. They are best thought of as having random deal limits drawn uniformly between 1 and the degree of the node. We call them *limA* and *limB*, and the difference is just that they are different randomizations.

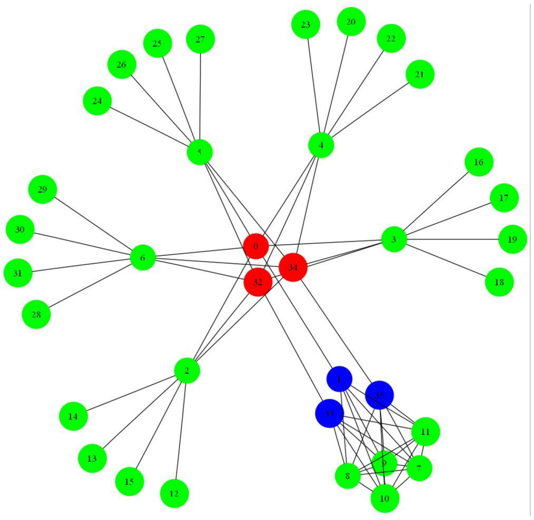


Figure 4.6: The 2ndHood graph for testing second neighborhood effects.

Identical First Neighborhoods

The final irregular graph was designed specifically to test if structure outside the immediate neighborhood of a node would affect its behavior. The network used for this test has two sets of three identical nodes, which are colored blue and red in Figure 4.6. Both sets have degree 6, and each of their neighbors have degree 7, so the local neighborhoods are indistinguishable in our GUI views. Any differences in behavior must be due to the second neighborhood or aspects even more distant. The second neighborhood of these two sets of nodes are drastically different; the second neighbors of the red nodes includes the 20 leaves while the second neighbors of the blues does not. This graph helped us identify the effects of second neighborhood when the first neighborhoods of two nodes were identical. We used it only with all nodes having deal limit 1. We refer to this scenario as *2ndHood*.

4.4.2 Regular Graphs

The 8 remaining scenarios were all based on regular graphs. This allowed us to test effects other than degree, like differing deal limits or large-scale market imbalances. One graph is a 36-node cycle shown. Four of them are identical tori with different deal limit schemes.

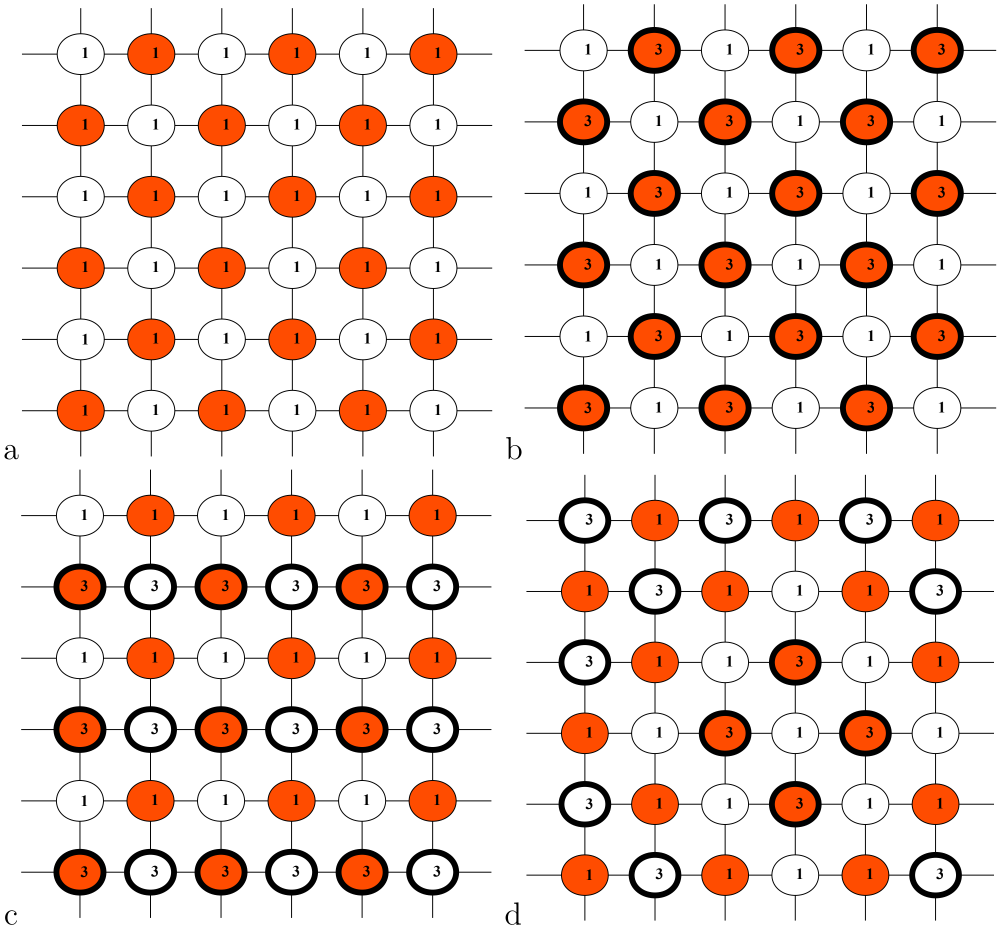


Figure 4.7: The edges running off the top and bottom denote wrap-around connections, as do those off the sides. (a) Torus Uniform. (b) Torus Checkerboard. (c) Torus Rows. (d) Torus Diamond.

Finally, three other graphs were used to observe the effects of a global supply imbalance, and are described in the following.

Tori

The 4 tori are topologically the same, and are differentiated only through deal limits:

- **Torus Uniform (torUniq).** In this network all nodes have deal limit 1. It is shown in Fig. 4.7a.

- **Torus Checkerboard (torChkb).** In this network all white nodes have deal limit 1, the others have deal limit 3. It is shown in Fig. 4.7b.

- **Torus Rows (torRows).** In this network alternating rows have deal limit 1 and deal limit 3. It is shown in Fig. 4.7c.

- **Torus Diamond (torDiamnd).** In this network some vertices have deal limit 1 and some have deal limit 3. It is shown Fig. 4.7d.

Imbalanced Supply Networks

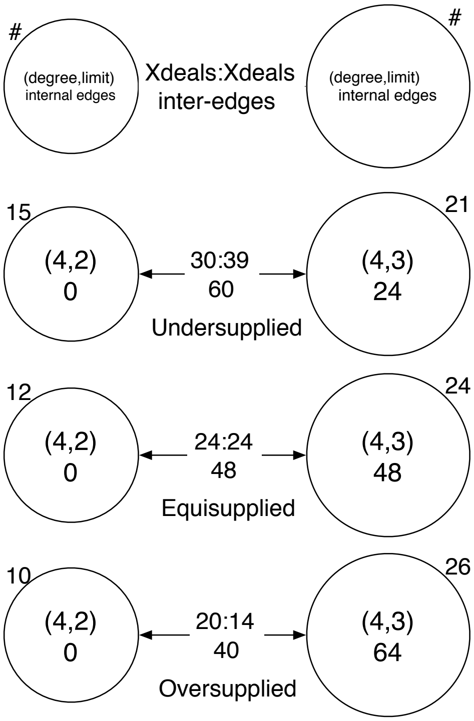


Figure 4.8: The top line is a template for interpreting the others. Xdeals means external demand. The "supply" in the 3 names refers to the number deals that the right side wants versus the number available.

The supply networks are 3 regular graphs which were designed to study the effect of a capacity issue which is not apparent at the node, but becomes apparent when contrasting the deal limits of two *groups* of nodes. Let the *external demand* of a group be the sum of deal limits of the nodes in the group minus the maximum number of deals that can be closed within the group.

In the supply networks, we defined the groups as the left group and right group as shown in Figure 4.8. All nodes have degree 4. All vertices in the left group have deal limit 2, and all vertices in the right group have deal limit 3. In each network, the right group has two different types of neighbors: those that belong to the right group, and those that belong to the left group. It is their differential treatment of the two types that was of interest. Nodes on the left have only one kind of neighbor; they exist just to set up the market conditions for those on the right.

The three graphs share the fact that all deal limits are either 2 or 3, and the ones on the right have both types of neighbours. They are different in the ratios of external demands between the left and right groups; in the Undersupplied case the right nodes are somewhat starved for deals (seeking 39 when only 30 could be forthcoming), in Equisupplied they are just balanced, and in Oversupplied they have more offers than they can use.

4.5 System Overview

Experiments were conducted using a distributed networked software system we have designed and built over the past several years for performing a series of behavioral network experiments on different games. This section briefly describes the user's view of that system in our bargaining experiments.

Like most microeconomic exchange models, the model described in Section 4.2 does not specify an actual temporal mechanism by which bargaining occurs, but of course any behavioral study must choose and implement one. At each moment of our experimental system, and on each edge of the network, each human subject is able to express an *offer* that is visible to the subject's neighbor on the other end of the edge. See Figure 4.1. The offer expresses the percentage of the benefit that a player is asking for.

When the portions on either end of an edge add up to exactly 100%, one of the players is able to *close* the deal by pressing a special button. Individuals can always see the offers made to them by their neighbors, as well as some additional information (including the degrees and limits of their neighbors, and the current best offers available to their neighbors). When a deal is closed, or when one of the partners has used up his limit of deals, the relevant edge mechanisms are frozen and no further action is allowed on them. Every game is stopped after 60 seconds. Any money riding on deals not closed within that time is simply "left on the table", i.e. the players never get it.

All communication takes place exclusively through this bargaining mechanism. Actions of a user are communicated to the central server, where information relevant to that action is recorded and communicated to the terminals of other users.

4.5.1 Human Subject Methodology

Our methodology for the recruitment, treatment and compensation of human subjects has Institutional Review Board approval at the University of Pennsylvania and broadly follows established practices in behavioral economics.

36 undergraduate Penn students were recruited from a related course taught by author Kearns. Subjects were familiar with simple graph concepts and their role in various real-life situations, but had no prior knowledge of the particular games to be played.

A single lab with enough Linux workstations for all 36 subjects was used. Each one ran a browser in a common account that was devoid of the students' personal distractions. The computer screens were arranged facing in opposite directions along long tables, so that it was difficult or impossible to see any other screen. In addition to the authors, several graduate student proctors were present during the experiments. All players were visible to proctors at all times, so any attempt to communicate via sight or sound would have been detectable. No books or electronics or any other materials were allowed anywhere but on the floor.

All players were let into the room together, and instructed to act as though they were taking an exam. No private conversation was allowed. We gave a presentation explaining the game to be played. In the first session, we started with games in the basic setting. It involved a review of the GUI, the mouse and keyboard controls, the goals of the players, the fact that graphs were generated according to different schemes not divulged, and the fact that players would be assigned to vertices in those graphs in an unbiased random fashion at the start of each experiment. We emphasized that players had no information on the global topology of any network used.

It was stressed that players' physical neighbors in the room were not necessarily neighbors in the graph, that the graph neighborhood would change with every game, and that the identities of players would not be made known during the game or at any time afterward, including all publications. Then all players logged into their machines. We did ask players to provide their name, but made it clear that the sole use of that information was to compute and distribute payments at a later date. One player's screen was temporarily projected on a large display at the front of the room while examples of play dynamics were demonstrated. Questions were taken and answered aloud. When all players were satisfied that they understood the purpose, mechanics, and semantics of the game and interface, we provided two sample games for them to play in which cash rewards were not given but questions were solicited.

We then started the sequence of 18 paying games in the basic setting. Each one was preceded by an empty screen saying “waiting for game”, then the game GUI appeared on the players' screens, and we announced this aloud in the room in order to verify that everyone's machine was functioning properly. Players became familiar with the local structure of their neighborhoods in the upcoming game. Their interfaces were live and would take inputs, but the server would not yet process orders. After a few seconds, we announced the beginning of the game and pressed a key on the server to enable play. A small bell rang on each computer, followed by a silent 2 minute period, whose progress was displayed on the elapsed time display at the top of every players' applet GUI. A small bell rang when time was up, and we also announced the end of play aloud. The screens remained frozen in their final game state so that players could take note of it. After a moment, at a command from the server console, the screens reverted to the “waiting” display. On average, less than half a minute was spent waiting for the next game to be ready.

After these games were completed, we gave a presentation describing the game in the cost setting, with a review of the GUI and the features added to that of the basic setting. We then started the sequence of 57 games, in the same way as we did for the games in the basic setting.

The second session was arranged in the same fashion. After the initial presentation that described the game the GUI for the limit setting, we followed it up by two demo games to familiarize the students with the GUI, and then conducted the sequence of 54 games.

At the end of either session, we asked players to fill out an online survey form to record observations, strategies, complaints, and suggestions about the games, the preparation, the equipment, and the general event. On average, players earned about $70 in 3 hours.

As noted early in this chapter, this report shall always be describing the second session (the main session) of experiments unless mentioned otherwise.

4.6 Results

Our results come under three broad categories. The first is about collective performance and social efficiency. The second category examines questions about the differential fates of nodes, depending on their position in the networks and the deal limits they each had. The third category is about the general performance of humans summarizing behavior across all the games they played. This is an area that no economic theory attempts to cover.

4.6.1 Social Welfare

Humans were quite effective at playing these games, but they paid a surprising price for their refusal to close some deals.

To quantify how well humans did on this problem, we implemented a greedy algorithm for comparison. Given a graph and deal limits, it first generates a random permutation of the edges, and then checks the edges one by one to close deals on them if this is possible, i.e. if both endpoints have not already saturated their deal limits. This process is repeated many times to obtain an average number of deals closed by the greedy algorithm. To normalize both the human and greedy systems we divide by the *Maximum Social Welfare*, which is the maximum number of deals that can close in each network, subject to both topology and deal limits. The *social welfare* is the number actually closed, and the ratio between this and the max is the *social efficiency*.

The observed human efficiencies are rendered in blue dots in Fig. 4.9, and the corresponding data are shown in Table . In 6 of the networks (those below the diagonal), the humans did worse than the greedy algorithm. Full efficiency is rare in both systems. One might view this as the *behavioral price of anarchy* due to selfish players operating with only local information. The greedy algorithm obtained an average of 92.14% of the maximum welfare in our networks. In comparison, human subjects achieved an average social welfare of 92.10% of the maximum welfare when averaged over all 3 trials, a surprisingly similar figure.

There are two parts to this story, though, because solving these problems involves both selecting edges and closing deals on them. The greedy algorithm does not address the deal-closing issue and perforce never leaves a potential deal unclosed; the humans often did. In 36 of the 54 experiments, the solution found by the human subjects was not even maximal – there were adjacent vertices that both could have closed another deal. Presumably this was because they simply could not agree on a split. However, the humans left the system in a state that could be improved post facto. We started the greedy algorithm in the final state the humans reached and allowed it to attempt to find more deals, thus producing a new state with no further unclosed deals. In all cases, this new state had a *higher* social efficiency than the greedy algorithm achieved alone. This is shown in the orange dots of Figure 4.9. A line connects the human performance to the potential human performance,

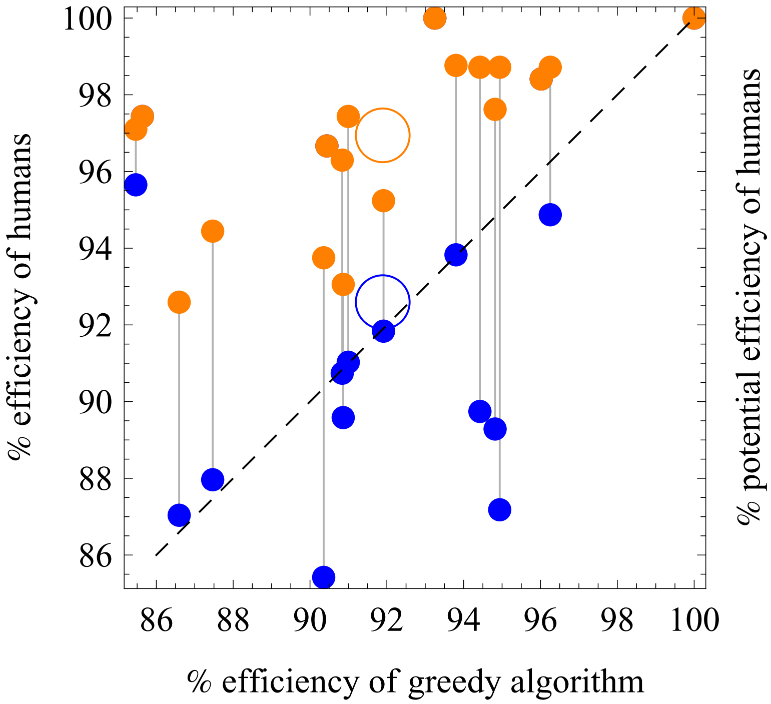


Figure 4.9: Blue dots are what the humans actually achieved. Orange dots are the result of applying the greedy algorithm to the final state of human play, which is what the humans could have achieved without obstinacy. Vertical lines thus show the price of obstinacy. The dotted line indicates equality of the two scales. The open circles represent the average values over all scenarios.

and we might dub this difference the *price of obstinacy*. In total, 7.9% of the money was "left on the table", but 4.5% was due to obstinacy (more than half the lost value).

We conclude that the humans found better matchings in the graph, and hence their behavioral price of *anarchy* is lower (better) than the greedy algorithm. But due to their additional *obstinacy*, their overall performance was no better.

4.6.2 Nodal Differences

There was much evidence that nodal income depends on its deal limit, its degree, and properties of the non-local neighborhood.

network	M	trials	Gr	Hu	Po
PLPlimA	27	26,24,26	0.94	0.94	0.99
PLPlimB	28	25,24,26	0.95	0.89	0.98
PLPuniq	16	14,12,15	0.9	0.85	0.94
PLNlimA	26	24,25,25	0.96	0.95	0.99
PLNlimB	26	24,22,22	0.95	0.87	0.99
PLNuniq	10	10,10,10	0.93	1.	1.
PL0limA	26	25,24,22	0.91	0.91	0.97
PL0limB	26	22,25,23	0.94	0.9	0.99
PL0uniq	13	12,13,13	0.86	0.97	0.97
cycle	18	16,17,14	0.86	0.87	0.93
Equisup	48	40,45,44	0.91	0.9	0.93
Oversup	49	43,46,46	0.92	0.92	0.95
Undrsup	42	41,41,42	0.96	0.98	0.98
2ndHood	10	10,10,9	0.9	0.97	0.97
torChkr13	18	18,18,18	1.	1.	1.
torDiamnd	23	23,21,22	0.86	0.96	0.97
torRows13	36	32,32,31	0.88	0.88	0.94
torUniq	18	15,18,16	0.91	0.91	0.96

Table 4.1: M is the maximum social welfare possible. Trials lists the number of deals closed in each of the 3 replications. Gr is the average efficiency of a greedy algorithm. Hu is the average efficiency of the humans. Po is the average potential of the humans.

Unequal Splits

Most theoretical models (for example, the Yamagishi-Cook model) that apply game-theoretic rationality to bargaining suggest that deals in some networks will be split in an unequal fashion. We will report the splits using their *inequality value* (Ineq), defined as the absolute difference between the two fractional shares. It ranges from 0 (equal sharing) to 1 (one player gets everything).

A total of 1271 deals were closed in all the 54 experiments, and 423 of them were split equally (inequality=0). But most were not split equally, every possible granular division was used for some splits, and 6 edges even had inequality=1 (which is surprising in itself since one partner gains nothing by signing the deal). The histogram in Figure 4.10 shows the inequality values. For comparison, we also show the histogram (in orange) from our preliminary session, which had no deal limits and produced an overwhelming portion of deals that split 50:50. The average inequality value over all games in our main session was 0.2097, which is a ratio of about 60:40, It thus seems clear that deal limits are invoking a significant increase in imbalanced splits.

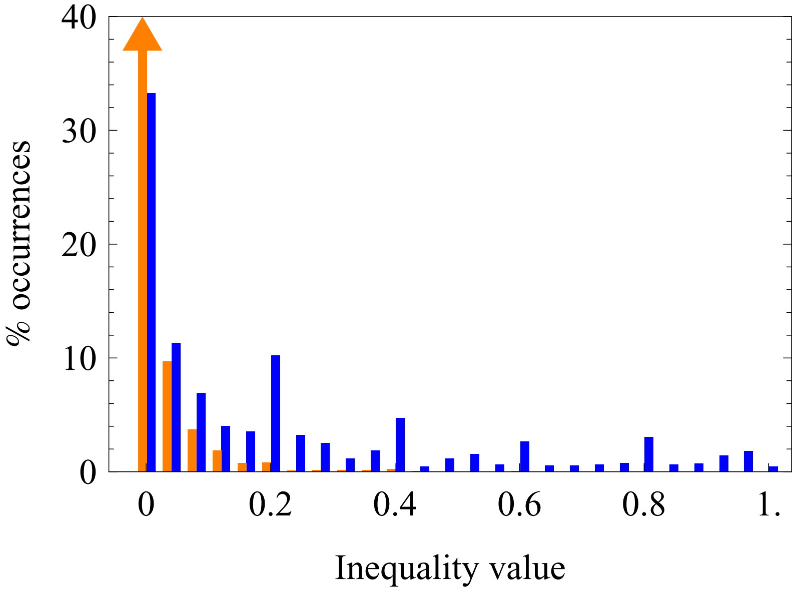


Figure 4.10: The orange distribution was from our preliminary session, and the bar at 0 (equal shares) goes to 82%. The blue bars are from our main session, where we obtained a much greater spread of unequal splits.

Inequality and Efficiency

There is a significant correlation between average inequality value and the social efficiency achieved in each scenario — that is, *when the subjects collectively tolerate greater inequality of splits, social welfare improves.* These data are plotted in Fig. 4.11 and shown in Table 4.2. The correlation coefficient is 0.52 with a confidence level of $p = .027$.

Interestingly, in an earlier session of experiments without deal limits, the same correlation is highly negative.

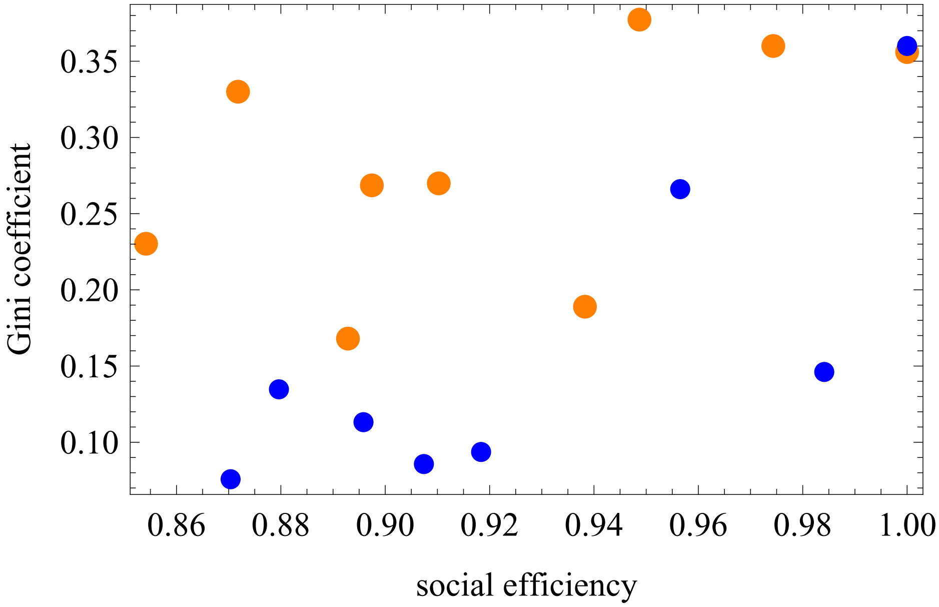


Figure 4.11: The 9 PL networks are plotted in orange, and the 8 regular networks are plotted in blue.

Degree Distribution and Equality

How well does degree distribution predict wealth distribution? We examined the PL networks to answer this. The average inequality value in closed deals is 0.23 over the 3 PLPuniq games (where nodes tend to be adjacent to nodes of similar degree), while it is 0.36 for both PLNuniq (where nodes tend to be adjacent to nodes of very different degree) and PL0uniq (where nodes tend to be adjacent to nodes of various degrees).

The inequality values of the PLPuniq experiments are less than the joint PL0uniq and PLNuniq outcomes with a one-sided $p < 0.02$. This indicates that *nodes that have an opportunity to bargain with at least one other of similar degree have more power than one that is forced to bargain only with higher-degree nodes.* (These are all unique-exchange games, so deal limit is not playing any distinguishing role.)

These three networks all have the same degree distribution. Hence, *degree distribution is not sufficient to predict inequality of wealth*, even in unique-exchange networks.

network	social efficiency	average inequality
PLPlimA	0.94	0.19
PLPlimB	0.89	0.17
PLPuniq	0.85	0.23
PLNlimA	0.95	0.38
PLNlimB	0.87	0.33
PLNuniq	1.	0.36
PL0limA	0.91	0.27
PL0limB	0.9	0.27
PL0uniq	0.97	0.36
cycle	0.87	0.08
Equisup	0.9	0.11
Oversup	0.92	0.09
Undrsup	0.98	0.15
2ndHood	0.97	0.5
torChkr13	1.	0.36
torDiamnd	0.96	0.27
torRows13	0.88	0.13
torUniq	0.91	0.09

Table 4.2: Inequality and social efficiency of each network, averaged over 3 trials.

Deal Limit Distribution and Equality

A similar story holds for deal limit distribution. Even if the distribution of deal limits for two networks are identical, the experimental results can differ widely based on whether a node bargains with nodes of similar deal limits or differing deal limits.

In Torus-Uniform, all vertices have the same deal limit. In Torus-Rows, all vertices have two neighbors with the same deal limit (1 or 3) and two neighbors with a different deal limit (3 or 1). In Torus-Checker, all nodes are bargaining with nodes of a different deal limit.

The average inequality values are 0.086, 0.13, and 0.36 for Torus-Uniform, Torus-Rows and Torus-Checker respectively. A means test shows these are all pairwise distinct with $p < .03$. These networks all have identical topologies. Thus, when network topology is not playing any distinguishing role, *if vertices bargain with vertices of similar deal limits, the deals are on more equal terms compared to when vertices with differing deal limits bargain.*

High Degree Confers Power

Over all closed deals in the PL graphs, the fractional take per closed deal of each node has a correlation of 0.47 with the degree of that node. If, to reduce the confound of differing deal limits, the study is confined to just the PL*uniq graphs, then the correlation coefficient is 0.59. Both these correlations are highly statistically significant. Thus, *bargaining power increases with the size of the local market*, at least in the setting where deal limits constrain behavior.

High Deal Limit Undermines Power

While higher degree confers bargaining power, higher deal limits had the opposite effect.

The Torus-Rows graph was designed specifically for testing the effect of deal limit on a node's bargaining power. In this graph, all nodes are identical up to relabeling, but half of them have deal limit 1 and half have deal limit 3. So if there is any systematic difference between two nodes' bargaining power, the difference in their limits can be the only explanation.

In the deals closed by limit-1 nodes, their mean fraction of the deal is 0.57. The limit-3 nodes obtained an average of 0.48. The difference was highly significant. (The two fractions do not add to unity because not all deals were between the two groups.) If only those deals between the two groups are considered, the fractions are 0.57 vs 0.43 and the difference is even more significant. The summary is that *a higher deal limit confers less bargaining power.*

Effect of Global External Demand

The supply networks were designed to study the effect of external demand. This property is not apparent at the node, but becomes apparent when contrasting the deal limits of two *groups* of nodes. In each supply network, the supply of deals from the left group (recall Figure 4.8) was manipulated to starve or overfeed the right group. How does the split of a deal depend on that relative supply?

In the Undersupplied case the right nodes must compete among themselves for the attention of nodes in the left side, so we might expect their shares to be smaller than the left side's. In the Equisupplied case, the external demands are equal, so we might expect no differential in bargaining power. In the Oversupplied case, the left nodes must compete for deals from the right side, so we might expect their share of the deals to be smaller than the right.

	external demand		avg shares	
	left	right	left	right
Undersupplied	30	39	0.57	0.43
Equisupplied	24	24	0.55	0.45
Oversupplied	20	14	0.52	0.48

Table 4.3: The average splits shown are for edges between left and right nodes. Edges between right nodes have an average share of 0.5 by definition.

Table 4.3 shows the results. There is a correlation of -0.19 ($p = 0.01$) between the external demand ratio and the deal share of the left nodes. The divisions favor the limit-2 nodes in all cases, consistent with the results of the previous section. However, that local property of relative limits is modulated by the global supply and demand ratio.

First Neighborhood Effects

We examined the three PL*uniq scenarios to find effects attributable to the degrees of first (one-hop) neighbors. For both nodes in all deals in the PL*uniq games, compute the fraction of the node's take and the average of the degrees of its neighbors. The correlation between these quantities is -0.60 and is highly significant. Similar results occur when the data are restricted to just those nodes with some fixed degree.

The clear and consistent story in unique-exchange games is that *the share obtained decreases as the average degree of the neighboring nodes increases*.

The opposite story holds when the first neighbors have higher *deal limits.* We compared the 4-regular networks torRows and torChkr. In torRows, a vertex of deal limit 1 has two neighbors of deal limit 1 and two neighbors of deal limit 3, while in torChkr, a vertex of deal limit 1 has all four neighbors with deal limit 3. The mean share of the former was 0.57 while the latter obtained 0.68. The difference is statistically significant with $p = .0001$. *The bargaining power of a vertex is enhanced when neighboring vertices have higher deal limits.*

Second Neighborhood Effects

How does the network effect propagate beyond the immediate neighborhood? The 2ndHood structure has two sets of 3 nodes, each of which have identical degree and first neighborhood degrees. The results of the previous section will be mute about how these nodes fare.

However, the second (two-hop) neighborhood of these nodes are drastically different: the neighbors' neighbors are leaves for 3 of them, and part of a clique for the other 3. The mean share of the first group was 0.347, the mean share of the second was 0.571, and the 2-sided p value was 0.027. *The bargaining power of a vertex is enhanced if its neighbors' neighbors have higher degree.*

4.6.3 Comparison with Theoretical Models

We shall now point out some structural differences in solutions given by theoretical models and those found by human subjects. For our main session, where nodes have limits, we narrow our attention to the PL*uniq networks, since the Cook-Yamagishi model [27] was originally designed for unique exchange networks, and fails to make a stable prediction on the 2ndHood network. The model predicts that maximum social welfare (maximum matching) will be achieved on all the PL*uniq networks, which is rare in the experiments, as reported in Section 4.6.1.

Further, the model predicts that a node with at least two leaves (nodes of degree 1) as neighbors always ends up with $1 - \epsilon$ fraction of a deal. This is due to myopic, rational competition between the leaves, where ϵ is the smallest non-zero amount that can be received by a node by signing a deal (this is the granularity of offers available in the GUI, and we let $\epsilon = 0.02$). Accordingly, the model predicts that there should be at least 30 such skewed deals in our experiments with the PL*uniq networks. In contrast, we find that there is 1 deal where one node gets 100%, 5 deals where one node gets 98%, and only 10 deals where one node gets more than 90%. Further, all but one of these deals are between a leaf node and a node of degree 5 or 6. This indicates that *extremely skewed deals are much rarer than what game-theoretic rationale suggests*, and is more likely when the degree differences are larger.

In our preliminary session, where nodes have no limits, unequal splits are rare, as reported in Figure 4.10. Chakraborty et. al. [20] designed a model for this setting. It predicts that all deals will be shared equally if the players ate the nodes have linear utility functions, and network effects may arise only due to non-linearity of player's utility. So the results of the experiments can be explained in this model if we assume that *in our range of payoffs, the players have near-linear utility functions.* This is not very surprising, since a player can make only a few dollars in each experiment.

4.6.4 Human Subject Differences

Humans were randomly assigned to nodes in each experiment and randomly reassigned in each replication of a scenario. Hence none of the results above could be ascribed to human differences. However behavioral literature is replete with examples of how human subjects leave their stamp, and some traits emerge in our data too.

Patience

The correlation between the average time for each human to close a deal and the average gain from closed deals, aggregated over all deals in all games, is 0.6664 ($p = 0.00$). See data in Figure 4.12. Apparently, *patience pays off.*

Proposer vs. Acceptor

The user interface mechanism involved the following protocol for completing a deal: one player (*proposer*) makes an offer, the other player (*acceptor*) accepts that offer by matching it, and then the proposer closes the deal. This was not designed to provoke any asymmetry, but

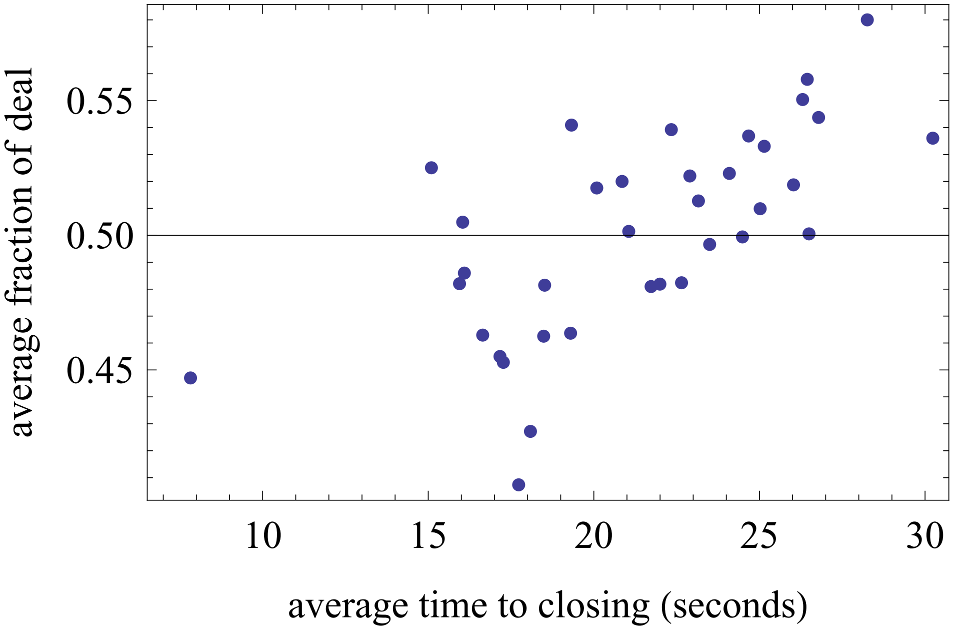


Figure 4.12: There are 36 dots here, corresponding to the 36 human players.

was intended to avoid unintended closed deals due to accidental mouse-clicks. Nevertheless, by looking only at what the shares of the two parties were, can we say which party is more likely to have been the proposer? We find that indeed we can, and the party which gets the higher share is more likely to be the proposer. The mean proposer share across all experiments was 53.6%, and the acceptor share was 46.4%. The Kolmogorov-Smirnov test rejects the hypothesis that these shares come from identical distributions with $p < 10^{-14}$. Some psychological effect is clearly being expressed by this subtle asymmetry in protocol.

All possible split ratios in closed deals were at least once proposed by someone (and accepted by someone), with the sole exception of 0:100%. The six cases where all the money went to one player were all proposed by the high-share side. It may be "irrational" for someone to agree to get 0, but it would have been even odder to see someone *propose* that he get 0.

The Effect of Uncertainty in Costs

This last section is strictly about our preliminary experimental session, in which there were no deal limits imposed–so the limit on each node was effectively its degree. Here we found that social efficiency was higher when the players were simply *uncertain* about a particular detail regarding their neighbors.

In the latter half of the preliminary session, we imposed varying *transaction costs* on nodes, which a node must pay for every deal it closes. The first half of the experiments had no costs. Occasionally during the latter half, we quietly imposed zero cost on every vertex. This allows us to compare those games to the setting without costs.

This cost was specific to each vertex. Only the player at that vertex, but not its bargaining partners, knew how much this cost was. We varied the costs significantly, from 0 up to 40% of the value of the deal. This generated enough uncertainty that in the few instances where every vertex had zero cost, no one could infer the costs of his partner. This 0-cost setting can be directly compared to the basic non-costed setting where every player *knows* that there are no costs involved. Hence the two situations were distinguished only by a lack of certainty.

Players closed more deals in the (uncertain) 0-cost case than in the (known) no-cost case. The efficiency columns of table 4.4 show the fraction of possible deals that were closed in the two cases. The fraction went up in all 5 networks; the difference is significant with $p = .004$. Evidently, the *level of obstinacy rises when people know for certain that their partner has no costs*.

The average inequality values of the deals and the standard deviations are also shown in the table. We expected the splits to be more uneven in the zero-cost case, but no consistent story was found.

	efficiency		average inequality		std. dev. of inequality	
	non	zero	non	zero	non	zero
PLP	0.85	0.96	0.012	0.01	0.033	0.037
PL0	0.84	0.93	0.009	0.009	0.025	0.029
PLN	0.72	0.93	0.027	0.012	0.057	0.051
CWC	0.84	0.95	0.015	0.023	0.035	0.076
2ndHood	0.84	0.97	0.014	0.009	0.040	0.044

Table 4.4: The CWC (short for *cycle with chords*) network was used only in session 1; all its nodes had degree 2 or 3. The others were as described for our main session.

4.7 Conclusions

The background theory is not yet prepared to describe all the phenomena we have observed here. Some bargaining theory suggests one party to a deal might get an infinitesimally small share, but our mechanism does not allow this. Hence our results cannot be exactly matched, but the scarcity of splits that are 98% or above seems to hint that the notion of "rationality" used by these theories needs to be adjusted. Other aspects of our results support theoretical models, notably the finding that phenomena at odd and even-length distances from a node alternately enhance and detract from the node's earnings.

The findings peculiar to people –namely the prevalence of obstinacy, the value of patience, the effect of protocol in the closing of a deal, and the state of knowledge about the partners– are all in need of theoretical development. It seems these findings argue for the further need to integrate the fields of economics, game theory, sociology, psychology, and computer science.

Chapter 5

Network Faithful Secure Computation

5.1 Introduction

Within artificial intelligence and machine learning, statistics, and signal processing, there has been great recent interest in an important class of highly distributed protocols on graphs known broadly as *message-passing* algorithms. Notable examples include belief propagation [78, 85], Gibbs sampling [18, 37], Nash propagation in graphical games [54, 76], gossip algorithms [14], survey propagation [16], constraint propagation [28], and many others. More generally, message-passing formalisms have long been studied in distributed computing. With rising interest in large-scale, decentralized networks such as the Internet, message-passing algorithms are of increasing appeal and importance due to their highly localized communication and their lack of any need for non-local topological information; in most instances parties do not even need to know the overall size of the network, yet they can compute sophisticated global functions, such as joint distributions and Nash equilibria.

In many applications of such algorithms, privacy may be an important consideration. Consider, for example, a large social network in which each node represents an individual and each edge represents a relationship between individuals. Imagine that each party in this network would like to compute his or her own probability of having contracted a contagious disease, which depends on the probabilities that each of his or her neighbors in the network have been infected. This could be accomplished by running the standard belief propagation algorithm on the network. However, if the network participants engage in standard belief propagation, each party will learn much more than their own probability of contracting the disease. In particular, each party could potentially learn information about the exposure probabilities of their neighbors, as well as more global information (such as the fraction of the population that has been infected). Obviously, such leakage of non-local information may be highly undesirable.

One approach would be to simply apply classic and powerful tools from secure multi-party computation [84, 39] (SMPC in the sequel) to the message-passing algorithms, preserving their input-output functionality while imbuing them with very strong privacy properties. Unfortunately, this straightforward approach would largely eradicate the benefits of the message-passing framework in the first place. Most importantly, classic SMPC would immediately "centralize" the computation, requiring all parties to maintain and communicate distributed shares of every computation in the original protocol — even if these computations are very "distant" in the network.

In this chapter we seek to get the best of both worlds — the highly distributed, local communication of message-passing, along with (at least some of) the traditional privacy assurances of SMPC. Our main results establish that essentially any message-passing protocol can be compiled into an "equally distributed" protocol that is secure with respect to individual parties misbehaving (1-privacy), and that security against even small coalitions is *impossible* without the cost of some centralization. Thus we demonstrate a fundamental trade-off between decentralization and security against coalitions that is tight.

We note that if all we ask is that the secure protocol take place on the same *graph* as the original protocol, with no attempt to pair *states* between the original and secure, there are trivial and uninteresting (and very inefficient) ways of obeying the network structure that inject security but still effectively centralize the computation. For instance, the network could simply be used as a global broadcasting mechanism for public keys, and then one could simulate the centralized solution of classical SMPC. For this reason it is important to have a more demanding definition of "faithfulness" to the original protocol, which indeed our 1-privacy compiler will obey. On the other hand, our notion of faithfulness is also quite general in that all we require is a matching of informational states for subsets of vertices between the original and secure protocols; yet we will show that *any* protocol obeying this very general definition is fundamentally limited in its immunity to collusion by coalitions.

The main contributions of this chapter are:

- A general formalization of message-passing algorithms on distributed networks that includes all of the examples mentioned above.

- A general compiler turning any message-passing algorithm into a protocol that is provably secure with respect to single parties (1-privacy). At a high level, this compiler carefully distributes and propagates the shares of each step of the computation over just those parties directly involved in it in the original protocol, as well as some of their neighbors.

- A (parameterized) definition of faithfulness to the original distributed protocol, and a proof that our compiler produces highly faithful 1-private protocols. The notion of faithfulness is information-theoretic and simulation-based, asking that views of sets of parties in the original be computable from views in the secure, and vice-versa.

- Impossibility results showing that the results above are essentially tight — namely, that protocols that are highly faithful to the original protocol must, in general, be susceptible to collusion by small coalitions. The generality of the definition of faithfulness shows that this trade-off is fundamental.

Related Work. To our knowledge, there are relatively few works that attempt to find constructions of SMPC that preserve more refined properties than just the computational complexity of the original (insecure) protocol (i.e. polynomial-time overhead). One exception is research attempting to preserve or minimize communication complexity [74]. The work most closely related to ours is that of Kearns et al. [59], who examine limited versions of the type of results presented here, giving secure protocols without proof for private belief propagation and Gibbs sampling. Here we generalize their results considerably, and also provide the aforementioned faithfulness notion and impossibility results.

5.2 Message-Passing Algorithms

Message-passing algorithms have been studied for decades in the distributed computing, artificial intelligence, signal processing, statistics, and information theory communities, among others. Several general definitions of message-passing algorithms have been proposed over the years. Aji and McEliece's generalized distributed law[1] generalizes many "sum-product" style message-passing algorithms including belief propagation, the Baum-Welch "forward-backward" algorithm, the Viterbi algorithm, fast Fourier transform, and others. The definition we provide here is strictly more general, and additionally includes any algorithms that fit into the general "message-passing model" of distributed computing.

Let $\mathcal{G}$ be a graph with vertices $\mathcal{X}$. For any node $X_i \in \mathcal{X}$, we denote by $\mathcal{N}(X_i)$ the set of neighbors of X_i. Loosely speaking, a message-passing algorithm on $\mathcal{G}$ is an algorithm in which information is passed from nodes to their neighbors, propagating through the graph over time through a series of local interactions. These messages generally include some local information pertaining to nearby nodes, but may also depend on information that originated at arbitrarily distant nodes. Formally, we define a message-passing algorithm on $\mathcal{G}$ as follows.

Definition 3 (message-passing algorithm)**.** *A distributed algorithm on a graph $\mathcal{G}$ is a* message-passing algorithm *if it can be abstracted in the following way. Each node $X_i \in \mathcal{X}$ maintains a current* state $\sigma(X_i)$ *which at all times contains its initial input and the content of the messages it has been passed.*[1] *The algorithm runs in a sequence of rounds. At each round t, there is a single distinguished node $X_{\nu(t)}$ that is central to computation, where the (possibly*

[1] The state is akin to the idea of a *view* in cryptography, but may also contain additional information.

randomized [2]*)* schedule ν *is fixed before the algorithm is run and is thus independent of the computation.*

Each round t *consists of two phases:*

1. Message Passing Phase: *Let* $\{U_1, U_2, ..., U_d\}$ *be the neighbors of* $X_{\nu(t)}$*. In the message-passing phase of the algorithm, each neighbor* U_j *passes a (possibly empty) message* $\mu(U_j)$ *to* $X_{\nu(t)}$ *where* $\mu(U_j)$ *may depend on* $\sigma(U_j)$ *in an arbitrary way.*

2. Computation Phase: *After receiving all of the messages* $\mu(U_j)$ *from its neighbors,* $X_{\nu(t)}$ *computes some (possibly vector-valued) function* $F_t(\sigma(X_{\nu(t)}), \mu(U_1), \mu(U_2), \ldots, \mu(U_d))$ *and sets his own current state to the output of this function.*

After the algorithm has been run for T *rounds, each node computes its own local output based on its current state.*

On the surface, this definition appears to differ from the standard message-passing model of distributed computing, where it is generally the case that at each round of computation, *every* node sends messages to *all* of its neighbors based on its view of the computation, and *every* party updates its view based on the messages it has received. However, note that any algorithm that fits into this framework also meets Definition 3 and vice versa, so we lose no generality by defining message-passing algorithms in this way. Our definition is more appropriate for algorithms such as belief propagation or Nash propagation in which information is propagated over time from one part of the graph to another and back again, with only a small set of vertices active at any time.

5.3 Secure Computation

Typically, a multi-party protocol is considered private in the semi-honest model of cryptography if anything a *set* or *coalition* of semi-honest parties could efficiently compute after

[2] Randomized update schedules are common, for instance, in applications of Gibbs sampling.

participating in the protocol could have been computed efficiently from their joint input and output alone. The difficulty of applying such a strong definition to distributed protocols is demonstrated in Section 5.5.1. Throughout this chapter, we consider the weaker notion of *k-privacy*, which requires privacy only against coalitions of size k or smaller. For example, 1-privacy requires that anything a *single party* could compute after participating in the protocol could be computed efficiently from that party's own input and output alone. More formally, k-privacy is defined as follows.

Definition 4 (k-privacy in the semi-honest model)**.** *Let $f : (\{0,1\}^*)^m \to (\{0,1\}^*)^m$ be an m-ary function, where $f_i(y_1,\ldots,y_m)$ denotes the ith element of $f(y_1,\ldots,y_m)$. Let Π be an m-party protocol for computing f. The* view *of the ith party, denoted $\text{VIEW}_i^\Pi(\vec{y})$ is defined as $(y_i, r_i, \vec{m}_i)$ where y_i is the private input of i, r_i is i's random string, and $\vec{m}_i$ is the sequence of incoming messages to i throughout the protocol. The* joint view *of a set $I = \{i_1,\ldots,i_\ell\}$, denoted VIEW_I^Π is defined as $(I, \text{VIEW}_{i_1}^\Pi,\ldots,\text{VIEW}_{i_\ell}^\Pi)$. We say that Π* k-privately computes *f if for every set I such that $|I| \leq k$, there exists a polynomial-time algorithm $\mathcal{S}$ (referred to as the* simulator*) such that*

$$\{(\mathcal{S}(I,(y_{i_1},\ldots,y_{i_\ell}),(f_{i_1},\ldots,f_{i_\ell})),f(\vec{y}))\}_{\vec{y}\in(\{0,1\}^*)^m} \overset{c}{\equiv} \{(\text{VIEW}_I^\Pi(\vec{y}),\text{OUTPUT}^\Pi(\vec{y}))\}_{\vec{y}\in(\{0,1\}^*)^m}$$

where $\text{OUTPUT}^\Pi(\vec{y})$ denotes the output sequence of all m parties after the given execution of Π and $\overset{c}{\equiv}$ denotes computational indistinguishability.

We will make use of the following remarkable and important theorem, which states that any m-ary function that can be computed efficiently by m parties can be jointly computed efficiently with arbitrary restrictions on who learns what. The two-party version of this result (abbreviated S2PC) was first developed by Yao [84] and was later extended to the multi-party case by Golreich et al. [39] Similar results have also been developed in the private channel model [9, 22].

Theorem 7 (Secure Multi-Party Computation (SMPC))**.** *Let $f(y_1,\ldots,y_m)$ be any (possibly* *randomized) m-input, m-output functionality that can be computed in polynomial time. Then under standard cryptographic assumptions,*[3] *there exists a polynomial time protocol Π that m-privately computes f (that is, a protocol in which party i or coalition I learns nothing not already implied by their private input and private output).*

The proof of Theorem 7 is *constructive*, providing a method to transform any polynomial circuit into a polynomial-time m-private protocol for m parties. Using this theorem, we could immediately infer the existence of a *centralized* algorithm for privately computing the output of a message-passing algorithm. We will show that our *highly distributed* protocol also maintains privacy. Our general protocol will rely on local applications of secure multi-party computation, each limited to only two parties.

5.4 The General Secure Propagation Protocol

In this section we describe the Secure Propagation protocol for securely executing any message-passing algorithm in the semi-honest model. We discuss how to extend this protocol to the malicious model in Appendix B.3.

Like the standard protocol for secure multi-party computation, Secure Propagation maintains the invariant that shares of each computation are distributed among multiple nodes in such a way that no one node can learn the values being computed. However, unlike standard SMPC, each computation in Secure Propagation is distributed among only a small number of nodes (in particular, a pair of neighbors), thus requiring that the shares be *propagated* as the center of computation moves around the graph from round to round. In particular, the state of any given node is always split between itself and each of its neighbors. The shares held by the neighbors must be updated each time the state itself is updated, effectively propagating the new information one step further in the graph.

Assume without loss of generality that the specification of the message-passing algorithm provides the schedule ν, that each node X_i knows the form of the function F_t it must calculate

[3] An example would be the existence of trapdoor permutations [38].

on all rounds t such that $\nu(t) = i$, and that each node knows how to properly calculate an outgoing message from its current state depending on which neighbor is requesting the message.[4] Let $\mathcal{D}(X_i) \in \mathcal{N}(X_i)$ denote a special *distinguished neighbor* of X_i, and assume that this neighbor has already been chosen for every i. The role of the distinguished neighbor will be to help maintain security whenever X_i is required to perform a computation.

Before the main protocol begins, each node X_i generates a public key $\text{pk}(X_i)$ and corresponding secret key $\text{sk}(X_i)$ using a key generation function G from any semantically secure public key encryption scheme and its random tape. X_i then distributes its public key to each of its neighbors, who in turn distribute it to each of their neighbors. The necessity of distributing keys to nodes two hops away in the graph is discussed in Section 5.5.1.

Throughout the execution of Secure Propagation, we will maintain the invariant that the state $\sigma(X_i)$ is split between X_i and each of its neighbors. More specifically, we assume that X_i is in possession of $\sigma_0(X_i)$ and that each neighbor of X_i is in possession of $\sigma_1(X_i)$, where $\sigma(X_i) = \sigma_0(X_i) \oplus \sigma_1(X_i)$. Thus, at the beginning of the protocol, each node X_i must split its initial state (i.e. its input) between itself and each of its neighbors. It can do this by generating $\sigma_0(X_i)$ uniformly at random from all strings of the appropriate length, setting $\sigma_1(X_i) = \sigma(X_i) \oplus \sigma_0(X_i)$, and distributing $\sigma_1(X_i)$ to each neighbor.

Now, at each round t of the message-passing algorithm, there will be some node X_i with neighbors $\{U_1, \ldots, U_d\}$ that needs to compute $F_t(\sigma(X_i), \mu(U_1), \mu(U_2), \ldots, \mu(U_d))$. This first requires computing each of the incoming messages. For each $U_j \in \mathcal{N}(X_i)$, X_i and U_j together have enough information to compute the message $\mu(U_j)$ since this depends only on $\sigma(U_j)$ which is split between them. Thus by Theorem 7 we know it is possible to construct a protocol for them to perform this computation in such a way that X_i learns only a value $\mu_0(U_j)$ and U_j learns only a value $\mu_1(U_j)$, where $\mu_0(U_j)$ and $\mu_1(U_j)$ are each distributed uniformly at random and $\mu(U_j) = \mu_0(U_j) \oplus \mu_1(U_j)$.

[4] This information should all be specified in the description of the particular message-passing algorithm.

Algorithm 3 The Secure Propagation protocol for the semi-honest model

```
// Input:  Schedule $\nu$, functions $F_t$ for all $t \in \{1,\ldots,T\}$,
// and a protocol for computing messages at each rounds
// Generation and distribution of public keys
for all nodes $X_i$ do
  $X_i$ generate public key $\mathrm{pk}(X_i)$ and secret key $\mathrm{sk}(X_i)$ using generation function $G$
  $X_i$ passes $\mathrm{pk}(X_i)$ to each $U_j \in \mathcal{N}(X_i)$, who in turn passes it to each node in $\mathcal{N}(U_j)$
end for
// Initial distribution of state information
for all nodes $X_i$ do
  $X_i$ generates $\sigma_0(X_i)$ uniformly at random
  $X_i$ sets $\sigma_1(X_i) \leftarrow \sigma(X_i) \oplus \sigma_0(X_i)$ and sends $\sigma_1(X_i)$ to each $U_j \in \mathcal{N}(X_i)$
end for
// The main protocol
for $t = 1$ to $T$ do
  Set $i \leftarrow \nu(t)$
  for all $U_j \in \mathcal{N}(X_i)$ do
    // Calculate the incoming messages for $X_i$
    // Here $\mu(U_j) = \mu_0(U_j) \oplus \mu_1(U_j)$
    $X_i$ and $U_j$ engage in S2PC; $X_i$ learns $\mu_0(U_j)$; $U_j$ learns $\mu_1(U_j)$
    if $U_j \neq \mathcal{D}(X_i)$ then
      $U_j$ sets $\mu_1^*(U_j) = Enc_{\mathrm{pk}(\mathcal{D}(X_i))}(\mu_1(U_j))$ and sends $\mu_1^*(U_j)$ to $X_i$
      $X_i$ sends $\mu_1^*(U_j)$ to $\mathcal{D}(X_i)$
      $\mathcal{D}(X_i)$ learns $\mu_1(U_j) = Dec_{\mathrm{sk}(\mathcal{D}(X_i))}(\mu_1^*(U_j))$
    end if
  end for
  // Privately calculate the updated state for node $X_i$
  // Here $\sigma(X_i) = \sigma_0(X_i) \oplus \sigma_1(X_i) = F_t(\sigma(X_i), \mu(U_1), \mu(U_2), \ldots, \mu(U_d))$
  $X_i$ and $\mathcal{D}(X_i)$ engage in S2PC; $X_i$ learns $\sigma_0(X_i)$; $\mathcal{D}(X_i)$ learns $\sigma_1(X_i)$
  for all $U_j \in \mathcal{N}(X_i) \backslash \mathcal{D}(X_i)$ do
    // Redistribute shares of $X_i$'s state to all of its neighbors
    $\mathcal{D}(X_i)$ sets $\sigma_1^*(X_i) = Enc_{\mathrm{pk}(U_j)}(\sigma_1(X_i))$ and sends $\sigma_1^*(X_i)$ to $X_i$
    $X_i$ sends $\sigma_1^*(X_i)$ to $U_j$
    $U_j$ learns $\sigma_1(X_i) = Dec_{\mathrm{sk}(U_j)}(\sigma_1^*(X_i))$
  end for
end for
// Final calculation and distribution of output
// The final output for $X_i$ is assumed to depend only on $\sigma(X_i)$
for all nodes $X_i$ do
  $X_i$ and $\mathcal{D}(X_i)$ engage in S2PC; $X_i$ learns its final output; $\mathcal{D}(X_i)$ learns nothing
end for
```

Now that $\mu(U_j)$ is split between X_i and U_j for each $U_j \in \mathcal{N}(X_i)$, it would be simple to again invoke Theorem 7 to show that there is a protocol for X_i and all of its neighbors to together securely compute the value of the function F_t. However, we would like to limit the applications of secure two-party computation to only take place only between *pairs* of nodes that are *neighbors* on the graph, and not require that it is applied to entire neighborhoods. We will accomplish this by transferring shares of information about the messages of each of X_i's neighbors to the distinguished neighbor $\mathcal{D}(X_i)$. Specifically, each $U_j \in \mathcal{N}(X_i)$ such that $U_j \neq \mathcal{D}(X_i)$ encrypts $\mu_1(U_j)$ using the public key of $\mathcal{D}(X_i)$. It sends this encrypted share of its message to X_i who passes it on to $\mathcal{D}(X_i)$. $\mathcal{D}(X_i)$ can decrypt the share using its secret key and obtain $\mu_1(U_j)$.

Once $\mathcal{D}(X_i)$ is in possession of these message shares for each neighbor, it is easy to see that X_i and $\mathcal{D}(X_i)$ will together be able to compute the value of the function F_t securely and calculate the new value of X_i's state, $\sigma(X_i) \leftarrow F_t(\sigma(X_i), \mu(U_1), \mu(U_2), \ldots, \mu(U_d))$. By Theorem 7, there is a way for them to compute it such that X_i learns only the new value $\sigma_0(X_i)$ and $\mathcal{D}(X_i)$ learns only the new value $\sigma_1(X_i)$, where $\sigma_0(X_i)$ and $\sigma_1(X_i)$ are each distributed uniformly at random and $\sigma(X_i) = \sigma_0(X_i) \oplus \sigma_1(X_i)$.

Finally, to maintain the invariant, the new value of $\sigma_1(X_i)$ must be distributed to the other neighbors of X_i. This can also be accomplished using public key encryption; $\mathcal{D}(X_i)$ encrypts $\sigma_1(X_i)$ using the public keys of each of X_i's neighbors and passes the encrypted shares to X_i who in turn passes them to the neighbors.

Once all T rounds have been completed, each node X_i and its distinguished neighbor can engage in S2PC one last time in order to allow X_i to learn its final output based on its current state. A complete formal description of the Secure Propagation protocol is given in Algorithm 3.

5.4.1 Proof of 1-Privacy

Theorem 8. *The Secure Propagation protocol* 1*-privately computes the output of a general message-passing algorithm in the semi-honest model. Furthermore, all communication takes place only between nodes that are neighbors on the original graph, and SMPC is never invoked on more than two parties.*

Proof. First, it is easy to see that Secure Propagation does in fact compute the same output as the original message-passing protocol; the proof of this, which is quite straight-forward, is omitted.

In order to show that Secure Propagation is 1-private (i.e. that it meets Definition 4 with $k = 1$), we must show that it is possible to simulate the view of any individual party given only this party's private input and output. The argument used here relies on the standard notion of an oracle-aided protocol. (See, for example, Volume II of Goldreich's *Introduction to Cryptography* [38].) For each node X_i, we will first show that X_i's view of the protocol can be simulated if each invocation of secure two-party computation is treated as a black box or *oracle call*. Then, using the fact that any application of secure two-party computation involving m parties is known to be m-private, we will show that the full view of X_i in Secure Propagation can also be simulated.

Imagine an oracle-aided protocol Π^O in which each application of S2PC is replaced with an oracle call where the values returned to each node by the oracle are the output values the nodes would have received from a real application of S2PC. We must show that it is possible to simulate the view of any node X_i during this oracle-aided protocol. Recall that the view of X_i consists of X_i's input, random bits, and incoming communication from other parties; we must then show that given X_i's input and randomness, we can simulate the incoming communication in such a way that the simulated view is computationally indistinguishable from the true view. We show how to simulate each message received by X_i during the protocol.

First, it is necessary to simulate the initial key generation and swapping phase. Since each node in the graph is assumed to generate its public and private keys from a fixed generator G, X_i can simulate this phase by generating public and private key pairs for itself and for every other node within two hops using G. X_i can simulate the one-time messages received from the nodes two hops away using the generated public keys; the secret keys can be discarded.

Throughout the remainder of the protocol, X_i will receive two types of messages: those which (by design) appear to be distributed uniformly at random (i.e. the shares of states and messages that are the results of local applications of S2PC), and those that are encrypted by its neighbors or its neighbors' neighbors (i.e. the encryptions of the message shares and the encryptions of the state shares). The former can clearly be simulated by drawing values uniformly at random; by design, these new values will be computationally indistinguishable from the shares returned by the S2PC applications.

To simulate a message that is encrypted using the public key of node X_j, the simulator simply generates a random string m and uses the value $Enc_{\mathrm{pk}'(X_j)}(m)$ where $\mathrm{pk}'(X_j)$ is the simulated public key for X_j that was generated during the simulation of the key distribution phase above. As discussed in Appendix B.1, it is the case that for any semantically secure encryption scheme, for any two strings m and m', the encryption of m is computationally indistinguishable from the encryption of m'. Since the simulated value $\mathrm{pk}'(X_j)$ is drawn from the same distribution as the public key $\mathrm{pk}(X_j)$ used in an actual execution of the protocol, $Enc_{\mathrm{pk}'(X_j)}(m)$ will be computationally indistinguishable from the true message received by X_i.

Note that on rounds in which X_i plays the role of the distinguished neighbor, it will be necessary to deal with the case in which $X_j = X_i$. In this case, we additionally need to make sure that the *decryption* of $Enc_{\mathrm{pk}'(X_j)}(m)$ is computationally indistinguishable from its counterpart in a real execution of the protocol. Because m was chosen uniformly at random, the decryption of $Enc_{\mathrm{pk}'(X_j)}(m)$ will of course yield a value that is distributed uniformly –

but this will also be the case in the real execution, since the decrypted values will be message shares that are distributed uniformly at random from the perspective of X_i.

We have shown how to design a simulator S_i^O to simulate the view of X_i in the oracle-aided protocol Π^O. Let $f_1, \ldots, f_\ell$ be the functions computed by the ℓ applications of S2PC in which X_i is involved. By Theorem 7, we can construct private two-party protocols $\Pi^{f_1}, \ldots, \Pi^{f_\ell}$ to execute each of these computations. Furthermore, from Definition 4, we know there must exist polynomial-time computable functions $S_i^{f_1}, \ldots, S_i^{f_\ell}$ that simulate the view of X_i in each of these protocols.

Let Π be the Secure Propagation protocol. By definition, this protocol is simply the oracle-aided protocol Π^O with the protocol Π^{f_i} plugged in for each application of S2PC for computing f_i. Consider a simulator S_i designed as follows. Run the simulator S_i^O as defined above, but in place of each oracle call, run the appropriate simulator $S_i^{f_i}$ with the input and output produced by S_i^O. We will show that the view of X_i in Π must be computationally indistinguishable from the output of the simulator S_i using a hybrid argument.

For $j = 1, \ldots, \ell$, let H_i^j be a hybrid simulator defined in the following way. Start by generating the view of X_i in a real execution of the protocol Π. Now, in place of the first j applications of S2PC in this view, substitute the *simulated* view of X_i using the simulator $S_i^{f_k}$ for $k = 1, \ldots, j$. (Note that the view of X_i in the execution of the real protocol will specify the input and output of each S2PC invocation and so the simulators can be run on this real input and output.)

Suppose that the view of X_i in Π is not computationally indistinguishable from the output of the simulator S_i. By a standard hybrid argument (see, for example, Goldreich [38]), it must be the case that either simulated view of X_i using H_i^1, the simulated view of X_i using S_i is distinguishable from the simulated view of X_i using H_i^ℓ, or for some $j \in \{1, \ldots, \ell - 1\}$, the view of simulated view of X_i using H_i^j is distinguishable from the simulated view of X_i using H_i^{j+1}. The second and third scenarios cannot occur due to Theorem 7. The first scenario cannot occur due to the above proof that the output of S_i^O is computationally indistinguishable from the view of X_i in Π^O. Thus it must be the case that the view of X_i in Π is computationally indistinguishable from the output of the simulator S_i, and Secure Propagation is secure. □

5.5 Faithfulness vs. Privacy

In this section, we illustrate the fundamental trade-off between the extent to which protocols are "faithful" to their (insecure) distributed message-passing sources, and the extent to which privacy against coalitions can be achieved. In particular, we give impossibility results showing that, at least for certain simple functionalities, faithfulness implies vulnerability to small coalitions.

Let us begin by observing that the Secure Propagation protocol is *not* secure against even coalitions of size two. In particular, at any point in the protocol, a vertex X_i and its distinguished neighbor $\mathcal{D}(X_i)$ can together compute the incoming messages from all of $\mathcal{N}(X_i)$. In turn, there are specific instantiations of (say) belief propagation in which such messages will reveal information about the private inputs of parties arbitrarily distant in the network.

On the other hand, we would like to argue that Secure Propagation *is* very faithful to its message-passing source. But what exactly does it mean for a protocol to be faithful? At a high level, a faithful protocol should follow the same general communication pattern as the corresponding insecure protocol. Computations should occur (approximately) in the same order, and should involve (approximately) the same parties as in the original protocol. We would not, for example, say that a secure protocol is faithful to the message-passing source if it utilizes centralized computation rather than allowing the computation to take place in a distributed manner over the graph. Before giving our formal definition of faithfulness, we briefly examine some alternatives.

- **Proposal 1:** *A secure protocol* Π_p *is faithful to the protocol* Π_o *if communication occurs* *between the same parties.*

While it is clearly desirable that the secure version of a message-passing protocol limit communication to occur between adjacent nodes on the graph, this simple definition is not enough to rule out centralized computation. For example, consider the following secure message-passing protocol. To begin, a special pair of neighboring nodes X_i and X_j generate public and private key pairs, and pass their public keys to each of their neighbors. Their neighbors then pass these public keys to each of their own neighbors and so on, until the keys have propagated all the way to the leaves of the network. Each node X_k in the network then generates its own public and private key pair and splits its initial state $\sigma(X_k)$ into two pieces, $\sigma_0(X_k)$ and $\sigma_1(X_k)$, such that $\sigma_0(X_k) \oplus \sigma_1(X_k) = \sigma(X_k)$. Next, X_k encrypts $\sigma_0(X_k)$ using the public key of X_i and $\sigma_1(X_k)$ using the public key of X_j, and propogates these encrypted state shares along with its own public key back through the network to X_i and X_j. At this point, X_i and X_j together can compute the private input of every node in the graph. As such, they can engage in S2PC and calculate shares of the private output of each node X_k, encrypted using X_k's public key. Finally, they can propogate these encrypted private outputs back through the graph, and each node can decrypt its own output.

It is easy to verify that this protocol is 1-private, and it does limit direct communication to occur only between nodes who communicated in the original message-passing algorithm. However, by centralizing all computation, it violates our notion of what it should mean for a protocol to be faithful.

- **Proposal 2:** *A secure protocol* Π_p *is faithful to the protocol* Π_o *if it requires a constant multiple of the number of steps or rounds used in* Π_o*.*

In an attempt to disallow such centralized solutions, we could instead define faithfulness based on the relationship between the number of steps required by the secure protocol and the number of steps required by the original. For example, we might say that a private protocol is faithful to the message-passing protocol if it requires a number of steps that is linear in the number of rounds of the original protocol. However, such a definition is still susceptible to many of the same problems as the first proposal. Many message-passing protocols such as belief propagation, Gibbs sampling, Nash propagation, already require a number of rounds that is linear in the size of the network. Thus the same centralized protocol described above would also meet this definition of faithfulness.

We instead choose to provide a much more general or weaker definition of faithfulness that we believe captures the right essential spirit — namely, correspondence of local information states or views during execution. Furthermore, a more general definition is a merit when proving impossibility results trading off faithfulness for privacy.

The basic idea behind our definition is as follows. We ask that there be two simulators — one for translating views in the secure protocol to views in the original, and another for the reverse direction. For any set $\mathcal{Y}$ of vertices in the network, given the view of $\mathcal{Y}$ in the original (respectively, secure) protocol, the corresponding simulator can *exactly* reconstruct the view in the secure (respectively, original) protocol *given unbounded computation time.* The reason for allowing unbounded computation time is that the secure protocol may of course employ various encryption or other information-hiding mechanisms during its simulation of the original (as ours does), but with unbounded computation time these operations can be "undone" to reveal the "real" computation taking place underneath. We wish to demand that these revealed computations be exactly those in the original. Thus an information-theoretic definition is appropriate.

Definition 5 (Faithfulness)**.** *Let Π_o be any multi-party protocol on a network and Π_p be a private protocol for executing this algorithm. Let M be the total number of messages passed in Π_o, and let $\tau(m)$ be the point in the execution of Π_o right after the mth message has been passed. We say that protocol Π_p is* faithful *to Π_o if there exist a mapping τ' from messages*

$\{1, \ldots, M\}$ to points in the execution of Π_p and simulators $\mathcal{S}_{o\to p}$ and $\mathcal{S}_{p\to o}$ such that for every subset of nodes $\mathcal{Y} \subseteq \mathcal{X}$ and for every $m \in \{1, \ldots, M\}$,

1. With unbounded computation time, $\mathcal{S}_{p\to o}$ can perfectly reproduce the joint view of $\mathcal{Y}$ in Π_o at time $\tau(m)$ from the joint view of $\mathcal{Y}$ in Π_p at time $\tau'(m)$ and set of random tapes used by $\mathcal{Y}$ in Π_p. In other words, nodes in Π_p have enough information to emulate the computation in Π_o.

2. With unbounded computation time, $\mathcal{S}_{o\to p}$ can perfectly reproduce the joint view of $\mathcal{Y}$ in Π_p at time $\tau'(m)$ from the joint view of $\mathcal{Y}$ in Π_o at time $\tau(m)$ and the random tapes used by $\mathcal{Y}$ in Π_p. In other words, nodes in Π_p don't have too much information.

As we shall see, it turns out that the right definition to establish the trade-off we seek is actually slightly more general, permitting views of $\mathcal{Y}$ and all vertices within some distance ℓ in the secure/original to be used in reconstructing views of $\mathcal{Y}$ in the original/secure. We now give the formal definition.

As before, let Π_o be the original, insecure distributed algorithm, and let Π_p be the privacy-preserving one. It will be convenient to think of Π_o as deterministic; of course any probabilistic algorithm can be converted into a deterministic algorithm by giving each party a random tape as part of their input. For a given set of nodes $\mathcal{Y}$, let $\mathcal{N}_\ell(\mathcal{Y})$ be the set containing $\mathcal{Y}$ and all nodes within ℓ hops of $\mathcal{Y}$ on the graph. For example, $\mathcal{N}_0(\mathcal{Y}) = \mathcal{Y}$, $\mathcal{N}_1(\mathcal{Y}) = \mathcal{Y} \cup \mathcal{N}(\mathcal{Y})$, and $\mathcal{N}_2(\mathcal{Y}) = \mathcal{Y} \cup \mathcal{N}(\mathcal{Y}) \cup \mathcal{N}(\mathcal{N}(\mathcal{Y}))$.

Definition 6 (ℓ-Faithfulness)**.** Define Π_o, Π_p, M, and τ as in Definition 5. We say that protocol Π_p is ℓ-faithful to Π_o if there exist a mapping τ' from messages $\{1, \ldots, M\}$ to points in the execution of Π_p and simulators $\mathcal{S}_{o\to p}$ and $\mathcal{S}_{p\to o}$ such that for every subset of nodes $\mathcal{Y} \subseteq \mathcal{X}$ and for every $m \in \{1, \ldots, M\}$,

1. With unbounded computation time, $\mathcal{S}_{p\to o}$ can perfectly reproduce the joint view of $\mathcal{Y}$ in Π_o at time $\tau(m)$ from the joint view of $\mathcal{N}_\ell(\mathcal{Y})$ in Π_p at time $\tau'(m)$ and set of random tapes used by $\mathcal{N}_\ell(\mathcal{Y})$ in Π_p.

2. *With unbounded computation time, $\mathcal{S}_{o\to p}$ can perfectly reproduce the joint view of $\mathcal{Y}$ in Π_p at time $\tau'(m)$ from the joint view of $\mathcal{N}_\ell(\mathcal{Y})$ in Π_o at time $\tau(m)$ and the random tapes used by $\mathcal{N}_\ell(\mathcal{Y})$ in Π_p.*

It is not difficult to show that the trivial centralized solutions we have discussed — for instance, using the network to broadcast public keys to establish secure pairwise channels, then using these channels to simulate classical SMPC computation in which every party has a share of every circuit wire — are not ℓ-faithful even for values of ℓ proportional to the number of vertices. This is because a centrally located vertex X may see encrypted messages between distant pairs of parties, and the unbounded computation time allows these messages to be read by X. In contrast, we have:

Theorem 9. *The Secure Propagation protocol is* 2*-faithful to the general message-passing algorithm.*

The proof involves showing that conditions 1 and 2 of Definition 6 are satisfied by Secure Propagation with $\ell = 1$ when $\mathcal{Y}$ consists of any individual node. Since the view of any individual node can be perfectly reproduced independently (with unbounded computation time), the joint views of any set of nodes can also be reproduced. Building $\mathcal{S}_{p\to o}$ is straight-forward; since messages in the original protocol are shared by a node and one of its neighbors in the secure protocol, only the views of a node and its neighbors in the private protocol are required to reconstruct any incoming messages in the original protocol. Building $\mathcal{S}_{o\to p}$ requires more care, and relies upon the availability of the random bits used in Secure Propagation by the nodes within a local neighborhood. We now give the proof in the following.

Proof. We define the mapping τ' that is required for Definition 6 in the following natural way. Suppose that message m is sent to node X_i by its neighbor U_j. Unless m is the last message sent in the current round, define $\tau'(m)$ to be the point at which U_j has already finished engaging in a secure two-party computation with X_i to compute the shares $\mu_0(U_j)$

and $\mu_1(U_j)$, and has just sent the encrypted version of $\mu_1(U_j)$ to X_i who has passed it along to $\mathcal{D}(X_i)$. If m is the last message of a round, define $\tau'(m)$ to be the point at which $\mathcal{D}(X_i)$ has finished redistributing the encrypted versions of $\sigma_1(X_i)$ to each of the neighbors of X_i through X_i. For the final message m that is passed, define $\tau'(m)$ to be the very end of the execution of Π_p.

We can then prove 2-faithfulness by showing that both conditions of Definition 6 are satisfied for $\ell = 2$ given this mapping.

Proof of Condition 1. We first show that for any node $X \in \mathcal{X}$, the view of X in Π_o at any time $\tau(m)$ can be perfectly reproduced by $\mathcal{S}_{p \to o}$ from the views of $\mathcal{N}_2(X)$ in Π_p at time $\tau'(m)$ and the random bits of $\mathcal{N}_2(X)$. (In fact, only the views of X and $\mathcal{N}(X)$ will be necessary; the views of $\mathcal{N}(\mathcal{N}(X))$ and random bits are not needed.) This immediately implies Condition 1 is satisfied by any set $\mathcal{Y}$, as the joint view of $\mathcal{Y}$ in Π_o can be perfectly reproduced by reproducing the view of each individual node in $\mathcal{Y}$ separately and combining the simulated views. To this end, it is sufficient to show inductively that any changes to the view of X in Π_o between time $\tau(m-1)$ and time $\tau(m)$ can be computed from the view of X and $\mathcal{N}(X)$ in Π_p at time $\tau'(m)$ along with the necessary random bits.

To do this, it is sufficient to consider the single node X_i who receives the mth message in Π_o; the views of the other nodes in Π_o do not change between $\tau(m-1)$ and $\tau(m)$. Suppose this message $\mu(U_j)$ is sent by $U_j \in \mathcal{N}(X_i)$. In the secure protocol, nodes U_j and X_i engage in an application S2PCin which X_i learns the value $\mu_0(U_j)$ and U_j learns the value $\mu_1(U_j)$. Because the simulator $\mathcal{S}_{p \to o}$ has access to the views of both X_i and U_j, it can learn both of their shares of $\mu(U_j)$ and calculate the value of $\mu(U_j)$ itself, which is precisely what is needed to update the view of X_i in Π_o.

Proof of Condition 2. We need to show that for any set $\mathcal{Y} \subseteq \mathcal{X}$ and any m, the view of $\mathcal{Y}$ in Π_p at time $\tau'(m)$ can be perfectly reproduced from the views of $\mathcal{N}_2(\mathcal{Y})$ in Π_o at time

$\tau(m)$ and the random bits used by $\mathcal{N}_2(\mathcal{Y})$ in Π_p by a simulator $\mathcal{S}_{o\to p}$ with no computational constraints. (Again, the views of $\mathcal{N}(\mathcal{N}(\mathcal{Y}))$ will actually be unnecessary, though the random bits will be needed.) It is again sufficient to show that this holds for individual nodes X, and as before, we prove this by induction on the messages sent.

The base case is trivially satisfied. Now suppose the hypothesis is true for time $\tau'(m-1)$. It is sufficient to show the view of X in Π_p from time $\tau'(m-1)$ to $\tau'(m)$ can be perfectly simulated from the view of $\{X\} \cup \mathcal{N}(X)$ at time $\tau(m)$ in Π_o.

Suppose $\nu(t) = i$. Assume without loss of generality that the neighbors of X_i are indexed according to the order in which they send messages to X_i in the current round, i.e. U_1 sends the first message, U_2 sends the second, and so on.

We first discuss how to simulate the basic changes in the views of each node between $\tau'(m-1)$ and $\tau'(m)$ that occur for *every* message m. We then go on to discuss the additional view changes that occur when m is the last message of the current round. For any message m that is *not* the last to be sent in the current round, only the views of of U_j, X_i, and $\mathcal{D}(X_i)$ change in Π_p between $\tau'(m-1)$ and $\tau'(m)$. We discuss how to simulate the view of each of these nodes in turn. Note that when $U_j = \mathcal{D}(X_i)$, it is necessary to combine the simulation techniques for neighbors U_j and the distinguished neighbor to produce the new simulated view.

The change in the view of U_j stems from the application of secure two-party computation with X_i, in which U_j provides input $\sigma_0(U_j)$, X_i provides input $\sigma_1(U_j)$, and both receive shares of the message $\mu(U_j)$. Since $\sigma_0(U_j)$ is part of the view of U_j in Π_p at time $\tau'(m-1)$, it is known inductively to the simulator. $\sigma(U_j)$ is also known, since this is essentially just the view of U_j in Π_o. Thus the simulator for U_j can obtain $\sigma_1(U_j) = \sigma(U_j) \oplus \sigma_0(U_j)$. Since it has access to the random bits used by both U_j and X_i in this application of S2PC, the view of U_j can be perfectly reproduced.

Now consider the change in the view of X_i. As in the previous case, the simulator for X_i has already computed $\sigma_1(U_j)$ (which is part of the view of X_i in Π_p at time $\tau'(m-1)$) and knows $\sigma(U_j)$ (which is essentially the view of U_j in Π_o). Since the simulator also has access to the random bits used by both X_i and U_j in Π_p, it can perfectly reproduce the views of both X_i and U_j in the S2PC application, and thus also $\mu_0(U_j)$ and $\mu_1(U_j)$. With the ability to calculate $\mu_1(U_j)$, the simulator for X_i can use the random bits of $\mathcal{D}(X_i)$ to compute its public key, enabling it to simulate the encrypted message that is passed from U_j to X_i as well.

$\mu_1(U_j)$, and can reproduce its appropriate encryption by first computing the public key of $\mathcal{D}(X_i)$ using its random bits.[5]

Now consider the case in which m is the last message of the current round. When this occurs, it is necessary to simulate the additional changes in the views of X_i and all $U_j \in \mathcal{N}(X_i)$ that occur when the new state of X_i is redistributed.

First, when the simulator is simulating either the view of X_i or the view of $\mathcal{D}(X_i)$, it must be able to simulate the appropriate view during the invocation of S2PC that computes the value of F_t. The simulators for both X_i and $\mathcal{D}(X_i)$ have access to the view of X_i in Π_o at time $\tau(m)$ which contains $\sigma(X_i)$ and $\mu(U_1), \ldots, \mu(U_d)$. As previously discussed, the simulator for X_i can compute both shares of the messages passed to X_i from each of its neighbors, while the simulator for $\mathcal{D}(X_i)$ can compute $\mu_1(U_1), \ldots, \mu_1(U_d)$, and thus can now also calculate $\mu_0(U_1), \ldots, \mu_0(U_d)$. By induction, $\sigma_0(X_i)$ and $\sigma_1(X_i)$ are known by the simulators of X_i and $\mathcal{D}(X_i)$ respectively since they belong to the respective views at time $\tau'(m-1)$. Given one of these values and $\sigma(X_i)$, the simulators can perfectly reproduce the views of both X_i and $\mathcal{D}(X_i)$ in the application of S2PC.

Next, when simulating the view of X_i, it is necessary to simulate the view when X_i is passed the new shares of its state from $\mathcal{D}(X_i)$, encrypted using the public keys of each of its other neighbors. Since we already know that the simulator for X_i is able to simulate $\sigma_0(X_i)$ and $\sigma_1(X_i)$, and since it is able to calculate the public key of U_j using the randomness of U_j,

[5] Here we are being somewhat imprecise about how the simulator for $\mathcal{D}(X_i)$ is able to calculate $\mu_1(U_j)$. This will depend on the precise details of how the shares $\mu_0(U_j)$ and $\mu_1(U_j)$ are calculated in the application of S2PC, which we have not discussed in detail here.

this case is straight-forward.

It remains to show that the simulator for any $U_j \in \mathcal{N}(X_i) \backslash \mathcal{D}(X_i)$ can simulate the appropriate change in view when this node receives its new share of X_i's state, encrypted using its own public key. Since the simulator can compute the new state $\sigma(X_i)$ of X_i from the view of X_i in Π_o, and has access to the random bits of X_i and $\mathcal{D}(X_i)$ that were used in the S2PC in which this new state was split into two shares, the simulator can reproduce the new value $\sigma_1(X_i)$.[6] Reproducing its encryption simply requires the random bits of U_j that were used to generate the public key. $\square$

5.5.1 The Impossibility Theorem

The Secure Propagation protocol requires each node to distribute its own public key to every node within two hops on the graph, and fails to guarantee privacy with respect to coalitions of size two or greater. We now present an impossibility result that shows that such requirements are unavoidable. Specifically, we show that any protocol that is ℓ-faithful to the general message-passing algorithm cannot be ℓ-private. Thus no 2-faithful protocol can guarantee security against coalitions of size 2, and furthermore it is not possible to create a 1-faithful protocol that is private even with respect to individual parties.

Theorem 10. *Any protocol Π_p that is ℓ-faithful to the original (insecure) general message-passing algorithm is not ℓ-private.*

Proof. We will first show that any protocol that is 1-faithful to the general message-passing algorithm is not private with respect to individual nodes, and then discuss how to extend this proof to the case of general ℓ-faithfulness.

Consider a graph with only three nodes, X_i, X_j, and X_k, and suppose that edges exist only between X_i and X_j and between X_j and X_k. Consider the following (insecure) message-passing protocol Π_o. Node X_i starts with input z, while X_j and X_k start with null input.

[6] Again, to be more precise here, we would need to go into more detail about how the shares are split in each S2PC based on the randomness of the nodes. This will be addressed in the full version of the paper.

In the first round, X_i passes z to X_j. In the second round, X_j passes z to X_k. Finally, X_k outputs the value z while X_i and X_j output null.

It is clear that if X_j learns the value of z during the execution of a protocol for implementing this algorithm, then the protocol is not 1-private; there cannot exist a simulator that consistently guesses the correct value of z after observing only the null input and output of X_j. We will show that during the execution of any 1-faithful protocol Π_p, X_j will learn the value of z, and therefore no 1-faithful protocol is private.

It will be useful to think explicitly about the *view* of each node at the end of the Π_p. Recall from Definition 4 that the the view of a party consists of the party's private input, private randomness, and all of the messages the party has been passed during the execution of the protocol. Let r_i, r_j, and r_k denote the random strings of X_i, X_j, and X_k respectively, and let $\vec{m}_{i \to j}$, $\vec{m}_{j \to i}$, $\vec{m}_{j \to k}$, and $\vec{m}_{k \to j}$ denote the sets of all messages passed between each pair of nodes.

Because we have assumed that Π_p is 1-faithful to Π_o, by definition of faithfulness it must be possible to perfectly reconstruct the view of X_i at the end of an execution of Π_p from z (which is the full view of X_i in Π_o), r_i, and r_j, given unbounded computation time. Consequently, it must be the case that any messages passed between X_i and X_j in Π_p can be constructed perfectly from z, r_i, and r_j, and thus we can assume without loss of generality that all interaction between X_i and X_j in Π_p occurs *before* any interaction between X_j and X_k.

At the point when X_i and X_j have finished interacting, the view of X_j consists of r_j and $\vec{m}_{i \to j}$, while the view of X_k is still simply r_k. Note that r_k is independent of r_j and $\vec{m}_{i \to j}$ since $\vec{m}_{i \to j}$ can be reconstructed from r_i, r_j, and z. Suppose that there exists an efficient protocol for X_j and X_k to exchange message in such a way that X_k is able to learn the value of z. Since r_k is simply a random string, this implies that X_j could *simulate* the role of X_k in this protocol and efficiently compute the value of z alone. Thus if Π_p allows X_k to learn the appropriate output, Π_p cannot be secure; X_j will be able to calculate the value of z too.

This proof can easily be extended to show that ℓ-faithfulness implies no privacy against coalitions of size ℓ or larger for any ℓ. The node X_j is now replaced by a chain of ℓ colluding nodes, $X_j^1, \ldots, X_j^\ell$. The message-passing algorithm Π_o now involves passing a single value z from X_i to X_j^1, who in turn passes it to X_j^2, and so on, until finally X_j^ℓ passes z to X_k. X_k outputs z, and all other nodes output null.

Assuming that the secure protocol Π_p is ℓ-faithful to Π_o, it must be the case that the view of X_i at the end of an execution of Π_p can be perfectly reconstructed with unbounded computational power from z (which is the full joint view of X_i and $X_j^1, \ldots, X_j^{\ell-1}$ in the original protocol), along with r_i and the randomness of the ℓ middle nodes. Thus we can again assume without loss of generality that any interaction between X_i and X_j^1 occurs before X_j^ℓ interacts with X_k.

As before, after the interaction with X_i is complete, if there is a protocol for $X_j^1, \ldots, X_j^\ell$ and X_k to exchange messages such that X_k learns the value of z, then $X_j^1, \ldots, X_j^\ell$ could together simulate the exchanges with X_k and learn the value of z themselves. Thus any ℓ-faithful protocol cannot be ℓ-private. □

Chapter 6

Summary of Contributions

This dissertation addresses several fundamental and important questions encountered in understanding the strategic and secure aspects of interactions in networks.

In our study of secure interactions in networks, we show how to make *any* kind of interaction over networks secure, while at the same time preserving the local and distribution nature of these interactions. We give a powerful 'compiler' that for each message-passing algorithm, produces a corresponding secure version that preserves exactly the same functionality and communication pattern. We also show a fundamental trade-off between preserving the local and distributed communication pattern of message-passing algorithms and the level of security that one can hope to achieve.

In our study of strategic interactions in networks, we examined two types of strategic interactions in networks that are of fundamental importance, *networked biased voting* and *networked bargaining*, using a variety of different techniques ranging from theoretical modeling and analysis, to behavioral experimentation.

The networked biased voting problem that we study aims to capture the tension between the expression of individual preferences and the desire for collective unity that is common in decision-making and voting processes, which often take place in social or organizational networks. In Chapter 2 we study this problem by modeling it as a biased opinion diffusion process, we then analyze the model to show that there exists networks where even a most minute amount of biased is enough to prevent the whole population from reaching a consensus in polynomial time, and demonstrate how this can be remedied by a carefully designed protocol that constitutes an approximate Nash equilibrium for the underlying biased voting game. In Chapter 3, we take this game to laboratory to conduct human subject behavioral experiments. We find that there are well-studied networks in which the minority preference consistently wins globally; and the presence of individuals with stronger preferences and the mixing of individuals of opposing preferences both reliably improve the chance of reaching a consensus. At the individual level, we find that striking the right balance of being stubborn in sticking to one's preference is highly correlated with one's earnings.

We continue our behavioral study of networked games in Chapter 4 with controlled human subject experiments on the networked bargaining game. Our experiments constitute the first large scale controlled experiments on bargaining game as a complementary to the large body of theoretical work on this subject found in the economics and sociology literature, where the most important premise for this line of study is that sheer topological differences of different location in a network play an important role in shaping bargaining power. Our experiments show that degree in general improves one's bargaining power whereas the limit on the number of deals one can get into does the opposite. As opposed to what many theoretical models suggest, we find that not only local structure, but also distant topology also affects one's bargaining power significantly in some networks. Our experiments also result in interesting findings that call for future theoretical development. For example, we find that network effect as disparity in bargaining power is largely muted when there is no limit imposed, i.e., when each individual can get into as many deals as his degree. Other examples include the observed positive correlation between inequality and social efficiency, and the fact that human subjects, through only local interactions, actually found significantly more efficient outcomes than what a centralized greedy algorithm can do.

Our study of strategic interactions in networks continue a recent line of research that aims to bring together economics, game theory, social science, and computer science to better understand how social and economic networks may play a role in shaping important strategic interactions in networks. In particular, the two sets of behavioral experiments that we describe in this dissertation are part of an extensive and continuing series that have been conducted at the University of Pennsylvania since 2005, in which collective problem-solving from only local interactions in networks has been studied on a wide range of tasks. The study of strategic interactions in networks is of significant importance because our modern life has become increasingly inter-connected in myriad ways, be it social, economic, technological and informational. And this is especially the case in the past fifteen years as the proliferation of the Internet has made the interaction of self-interested agents almost ubiquitous. Looking forward, we expect this line of investigation to continue to produce fruitful results that are of great interest to the economics, sociology and computer science community.

Appendix A

Some Tools from Probability Theory

A.1 Markov's Inequality

In probability theory, Markov's inequality gives an upper bound on the probability of a non-negative random variable X deviating from its expected value.

Theorem 11. (Markov's Inequality) *Let X be a random variable that only takes non-negative values, then for any $\lambda > 0$,*

$$P(x \geq \lambda) \leq \frac{\mathbb{E}(X)}{\lambda}.$$

A.2 Chernoff-Hoeffding Bound

In probability theory, Chernoff-Hoeffding bound provides an upper bound on the probability for the sum of random variables to deviate from its expected value. It is first published in [23] and [44].

Theorem 12. (Chernoff-Hoeffding Bound) *Let X_1, X_2, ..., X_n be independent and bounded random variables such that $X_i \in [a_i, b_i]$ for $i = 1, 2, ..., n$, let $X = \sum_{i=1}^{n} X_i$ be the* *sum of these random variables. Then*

$$P(X - \mathbb{E}(X) \geq tn) \leq \exp\left(-\frac{2t^2n^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right), \text{ and}$$
$$P(\mathbb{E}(X) - X \geq tn) \leq \exp\left(-\frac{2t^2n^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Note it is not required that X_1, X_2, ..., X_n be identically distributed, they only need to be independent. And when $X_i \in [0,1]$, the inequalities are simplified to

$$\begin{aligned} P(X - \mathbb{E}(X) \geq tn) &\leq e^{-2t^2n}, \text{ and} \\ P(\mathbb{E}(X) - X \geq tn) &\leq e^{-2t^2n}. \end{aligned}$$

A.3 A Convergence Theorem of Finite Markov Chains

In this section, we state a *convergence theorem* that quantifies the speed of convergence of finite Markov Chains. To this end, we need to be able to measure the distance between distributions. We consider a measure called *total variation distance.* The *total variation distance* between two probability distribution μ and ν over the same space Ω is defined by

$$||\mu - \nu||_{TV} = \max_{A \subset \Omega} |\mu(A) - \nu(A)|$$

The convergence theorem says that the total variation distance between the distribution of an irreducible and aperiodic Markov chain and its stationary distribution approaches 0 exponentially fast.

Theorem 13 (Convergence Theorem)**.** *Denote by P_x^t the probability distribution of a finite Markov chain over state space Ω at time t given it starts in state $x \in \Omega$. If the Markov chain is* irreducible *and* aperiodic, *with stationary distribution π. Then there exists constants*

$c_0 > 0$ *and* $c_1 \in (0, 1)$ *such that*

$$\max_{x \in \Omega} ||P_x^t - \pi||_{TV} \leq c_0 c_1^t$$

For detailed discussion on mixing properties of Markov chains, we refer readers to [3, 67, 66].

Appendix B

Additional Backgrounds for Chapter 5

B.1 Public-Key Encryption Schemes

A public-key encryption scheme consists of a triple (G, Enc, Dec) of probabilistic polynomial-time algorithms. G is a *key-generation algorithm* which outputs a random pair consisting of a public key pk and secret key sk, Enc is an *encryption algorithm*, and Dec is a *decryption algorithm*. We use $Enc_{pk}(x)$ to denote the encryption of a string x using public key pk, and $Dec_{sk}(y)$ to denote the decryption of a ciphertext y using secret key sk.

Informally, any public-key encryption scheme should satisfy the following properties. First, for any n-bit input x, $Dec_{\mathrm{sk}}(Enc_{\mathrm{pk}}(x)) = x$; that is, decryption is the inverse of encryption. Additionally, for any n-bit x, it is computationally hard for a party knowing only the public key pk and the encryption $Enc_{\mathrm{pk}}(x)$ to compute x. There are two fundamental ways to formalize this property [41]. At a high level, *semantic security* states that it is infeasible to obtain any information about the plaintext (decrypted message) from the ciphertext (encrypted message). In other words, anything one could efficiently compute about the plaintext from the ciphertext could also be computed efficiently *without* the ciphertext, given only the length of the plaintext. For the results presented in this paper, it will be more convenient to consider the second formalization of security, which requires that it is infeasible to distinguish between the encryptions of two distinct ciphertexts. In other words, the collections of encryptions of the ciphertexts are computationally indistinguishable, even if the public key is given.

Goldwasser and Micali [41] and Micali et al. [70] proved that these two definitions of security are equivalent. Thus for any semantically secure encryption system, the encryptions of two distinct ciphertexts are computationally indistinguishable. For more information on encryption schemes and security, see Chapter 5 of Goldreich [38].

B.2 Public-key Signature Schemes

A public-key signature scheme consists of a triple (G, Sg, Vf) of probabilistic polynomial-time algorithms. G is a *key-generation algorithm* that outputs a random pair consisting of a signing key sgk and verification key vfk; the verification key can be made publicly known while the signing key is kept secret. Sg is an *signing algorithm*, and Vf is a *verification algorithm*. We use $Sg_{sgk}(x)$ to denote the signed message produced from string x and signing key sgk, and $Vf_{\text{vfk}}(x, y) \in \{0, 1\}$ to denote whether y is a properly signed message produced from x and the signing key corresponding to vfk.

A public-key signature scheme is useful because it satisfies the following properties. First, for any input x of the appropriate length, $Pr[Vf_{\text{vfk}}(x, Sg_{\text{sgk}}(x)) = 1] = 1$. Second, it is computationally infeasible for any party without knowledge of sgk to forge a properly signed message, even if the party has access to vfk and messages already signed with sgk.

For detailed discussion on public-key signature scheme, we refer readers to Chapter 6 of Goldreich [38].

B.3 The Malicious Model

Until this point, we have been interested only in the semi-honest model in which it is assumed that all parties obediently follow the protocol and only potentially "cheat" by attempting to learn more information than they should from the messages they receive. This assumption cannot be enforced in many settings. However, standard tools from cryptography, such as zero-knowledge proofs of knowledge, can be used to ensure that any party who deviates from the protocol is *caught*.

At a high level, an interactive proof system is *zero-knowledge* if the prover is able to convince the verifier to accept a statement with high probability if and only if the statement is true, and furthermore if the proof gives away no additional information other than the truth of the statement. Goldreich et al. [40] showed that it is possible to construct a zero-knowledge proof system for every language in NP; in other words, it is possible to construct a zero-knowledge proof of any statement for which there exists a short, efficiently verifiable proof or *certificate*. Furthermore, efficient zero-knowledge proofs exist whenever the prover is in possession of the certificate.

We say that a protocol is secure in the malicious model if it is privacy-preserving and any party who attempts to deviate from semi-honest behavior is immediately caught. It is assumed that once a cheater is caught, the protocol halts.[1] A privacy result analogous to Theorem 7 can be proved in the malicious model, stating that under standard cryptographic assumptions, for any m-input, m-output functionality, there exists a (centralized) protocol that securely computes the functionality in the malicious model.

In order to extend the Secure Propagation protocol to the malicious model, we make use of both public-key encryption and public-key signature schemes. As before, we first need nodes to be able to distribute their keys (both pk and vfk) to their neighbors and the neighbors of their neighbors. However, since we are now in a malicious model, it is no longer the case that a node cannot rely on its immediate neighbor to relay such information to neighbors two hops away. Therefore, a "preprocessing" key-distribution phase involving some communication mechanism extraneous to the given network is needed to achieve this. For example, this can be implemented by setting up a one-time direct communication between nodes two hops away. Such communication extraneous to the network is possibly expensive,

[1] The assumption that it is possible to halt when a party is caught cheating may seem strange in our distributed setting, but is often justifiable, for example if there exists an "expensive" broadcast mechanism that could be used to notify everyone on the network to halt the execution if a cheater is detected.

but is needed *only* for the key-distribution phase; the main protocol we describe below limits communication to parties connected by an edge on the underlying network.

In the semi-honest model, each time a S2PC is invoked between two parties, say X_i and U_j, the resulting outputs are distributed between them so that their shares sum up to some meaningful information (i.e a message or a state) in the original protocol. We then rely on U_j to honestly encrypt its share of the information and distribute it to the other neighbor(s) of X_i via X_i itself. In the malicious model, we cannot expect U_j to follow this honestly, nor can we trust that X_i will relay the true encryptions as opposed to alternate forged messages. Indeed, we will require both public-key encryption and public-key signature to be part of the S2PC so that X_i in addition obtains as output an appropriate signed and encrypted version of U_j's share of the information, and this signed and encrypted messages is then sent to X_i's other neighbor(s). This way, we prevent U_j from sending messages that are encrypted from anything other than his share of the information and we also prevent X_i from forging such encryptions.

Along with the above modifications, we need both parties in the S2PC to prove that any inputs they supply to the S2PC are the legitimate ones. This is enforced by applying zero-knowledge proofs in a standard way.

Bibliography

[1] Srinivas M. Aji and Robert J. McEliece. The generalized distributive law. *IEEE Transactions on Information Theory*, 42(2):325–343, 2000.

[2] Susanne Albers, Stefan Eilts, Eyal Even-Dar, Yishay Mansour, and Liam Roditty. On nash equilibria for a network creation game. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 89–98, 2006.

[3] David J. Aldous and James A. Fill. Reversible markov chains and random walks on graphs. Manuscript, 1994.

[4] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 295–304, 2004.

[5] Elliot Anshelevich, Anirban Dasgupta, Éva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. *Theory of Computing*, 4(1):77–109, 2008.

[6] Yossi Azar, Benjamin E. Birnbaum, L. Elisa Celis, Nikhil R. Devanur, and Yuval Peres. Convergence of local dynamics to balanced outcomes in exchange networks. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 293–302, 2009.

[7] Albert László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[8] Xiaohui Bei, Wei Chen, Shang-Hua Teng, Jialin Zhang, and Jiajie Zhu. Bounded budget betweenness centrality game for strategic network formations. In *Proceedings of the 17th Annual European Symposium on Algorithms*, pages 227–238, 2009.

[9] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 1–10, 1988.

[10] Elisa Jayne Bienenstock and Phillip Bonacich. The core as a solution to exclusionary networks. *Social Networks*, 14:231–244, 1992.

[11] Ken Binmore. *Game Theory and the Social Contract, Volume 2: Just Playing*. The MIT Press, 1998.

[12] Lawrence E. Blume, David Easley, Jon Kleinberg, and Éva Tardos. Trading networks with price-setting agents. *Games and Economic Behavior*, 67(1):36–50, 2009.

[13] Béla Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[14] Stephen Boyd, Arpita Ghosh, Salaji Prabhakar, and Devavrat Shah. Gossip algorithms: Design, analysis, and applications. *IEEE Transactions on Information Theory*, 42(6):2508–2530, 2006.

[15] Norman Braun and Thomas Gautschi. A nash bargaining model for simple exchange networks. *Social Networks*, 28(1):1–23, 2006.

[16] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures & Algorithms*, 27(2):201–226, 2005.

[17] C. Camerer. *Behavioral Game Theory*. Princeton University Press, 2003.

[18] George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46:167–174, 1992.

[19] Tanmoy Chakraborty, Stephen Judd, Michael Kearns, and Jinsong Tan. A behavioral study of bargaining in social networks. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, pages 243–252, 2010.

[20] Tanmoy Chakraborty, Michael Kearns, and Sanjeev Khanna. Network bargaining: algorithms and structural results. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 159–168, 2009.

[21] G. Charness, M. Corominas-Bosch, and G. Frechette. Bargaining and network structure: An experiment. *Journal of Economic Theory*, 136(1):28–65, 2007.

[22] David Chaum, Claude Crépeau, and Ivan Damgård. Multi-party unconditionally secure protocols. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 11–19, 1988.

[23] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23(4):493–509, 1952.

[24] Peter Clifford and Aidan Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581–588, 1973.

[25] Karen S. Cook and Richard M. Emerson. Power, equity, and commitment in exchange networks. *American Sociological Review*, 43(5):721–739, 1978.

[26] Karen S. Cook, Richard M. Emerson, Mary R. Gillmore, and Toshio Yamagishi. The distribution of power in exchange networks: Theory and experimental results. *The American Journal of Sociology*, 89(2):275–305, 1983.

[27] Karen S. Cook and Toshio Yamagishi. Power in exchange networks: A power-dependence formulation. *Social Networks*, 14(3-4):245–265, 1992.

[28] Rina Dechter. *Constraint Processing.* Morgan Kaufmann, 2003.

[29] P. Erdös and A. Rényi. On random graphs. I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[30] Paul Erdös and Alfréd Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[31] Éva Tardos and Tom Wexler. *Algorithmic Game Theory*, chapter Network Formation Games and the Potential Function Method, pages 487–516. Cambridge University Press, 2007.

[32] Eyal Even-Dar and Michael Kearns. A small world threshold for economic network formation. In *Advances in Neural Information Processing Systems 19*, pages 385–392, 2006.

[33] Eyal Even-Dar, Michael Kearns, and Siddharth Suri. A network formation game for bipartite exchange economies. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 697–706, 2007.

[34] Alex Fabrikant, Ankur Luthra, Elitza N. Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of the 22nd Annual ACM Symposium on Principles of Distributed Computing*, pages 347–351, 2003.

[35] Noah E. Friedkin. An expected value model of social power: Predictions for selected exchange networks. *Social Networks*, 14(3-4):213–230, 1992.

[36] Douglas M. Gale and Shachar Kariv. Trading in networks: A normal form game experiment. *American Economic Journal: Microeconomics*, 1(2):114–132, 2009.

[37] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[38] Oded Goldreich. *Foundations of Cryptography*, volume 2. Cambridge University Press, 2004.

[39] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game – a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 218–229, 1987.

[40] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.

[41] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Science*, 28(2):270–299, 1984.

[42] Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[43] Yair Halevi and Yishay Mansour. A network creation game with nonuniform interests. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*, pages 287–292, 2007.

[44] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[45] Richard A. Holley and Thomas M. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *Annals of Probability*, 3:643–663, 1975.

[46] Matthew O. Jackson. The economics of social networks. In *Advances in Economics and Econometrics, Theory and Applications: Ninth World Congress of the Econometric Society*, volume 1. Cambridge University Press, 2005.

[47] Ramesh Johari, Shie Mannor, and John N. Tsitsiklis. A contract-based model for directed network formation. *Games and Economic Behavior*, 56(2):201–224, 2006.

[48] J. Stephen Judd and Michael Kearns. Behavioral experiments in networked trade. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 150–159, 2008.

[49] Sham Kakade, Michael Kearns, John Langford, and Luis E. Ortiz. Correlated equilibria in graphical games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 42–47, 2003.

[50] Sham M. Kakade, Michael J. Kearns, Luis E. Ortiz, Robin Pemantle, and Siddharth Suri. Economic properties of social networks. In *Advances in Neural Information Processing Systems 17*, 2004.

[51] Yashodhan Kanoria, Mohsen Bayati, Christian Borgs, Jennifer T. Chayes, and Andrea Montanari. A natural dynamics for bargaining on exchange networks. *CoRR*, abs/0911.1767, 2009.

[52] Michael Kearns. *Algorithmic Game Theory*, chapter Graphical Games, pages 159–178. Cambridge University Press, 2007.

[53] Michael Kearns, Stephen Judd, Jinsong Tan, and Jennifer Wortman. Behavioral experiments on biased voting in networks. *Proceedings of the National Academy of Sciences*, 106(5):1347–1352, 2009.

[54] Michael Kearns, Michael L. Littman, and Satinder Singh. Graphical models for game theory. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*, pages 253–260, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[55] Michael Kearns and Siddharth Suri. Networks preserving evolutionary equilibria and the power of randomization. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, pages 200–207, 2006.

[56] Michael Kearns, Siddharth Suri, and Nick Montfort. An experimental study of the coloring problem on human subject networks. *Science*, 313(5788):824–827, 2006.

[57] Michael Kearns and Jinsong Tan. Biased voting and the democratic primary problem. In *Proceedings of the 4th International Workshop on Internet and Network Economics*, pages 639–652, 2008.

[58] Michael Kearns, Jinsong Tan, and Jennifer Wortman. Network-faithful secure computation. 2007.

[59] Michael Kearns, Jinsong Tan, and Jennifer Wortman. Privacy-preserving belief propagation and sampling. In *Advances in Neural Information Processing Systems 20*, 2007.

[60] Michael Kearns and Jennifer Wortman. Learning from collective behavior. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 99–110, 2008.

[61] Jon Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[62] Jon Kleinberg. *Algorithmic Game Theory*, chapter Cascading Behavior in Networks: Algorithmic and Economic Issues, pages 613–632. Cambridge University Press, 2007.

[63] Jon Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 295–304, 2008.

[64] Jon Kleinberg, Siddharth Suri, Éva Tardos, and Tom Wexler. Strategic network formation with structural holes. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pages 284–293, 2008.

[65] Nikolaos Laoutaris, Laura J. Poplawski, Rajmohan Rajaraman, Ravi Sundaram, and Shang-Hua Teng. Bounded budget connection (bbc) games or how to make friends and influence people, on a budget. In *Proceedings of the 27th Annual ACM Symposium on Principles of Distributed Computing*, pages 165–174, 2008.

[66] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[67] László Lovász. Random walks on graphs: A survey. 2:1–46, 1993.

[68] Robert D. Luce and Howard Raiffa. *Games and Decisions*. Wiley, 1957.

[69] Barry Markovsky, John Skvoretz, David Willer, Michael Lovaglia, and Jeffrey Erger. The seeds of weak power: An extension of network exchange theory. *American Sociological Review*, 5:197–209, 1993.

[70] Silvio Micali, Charles Rackoff, and Bob Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal on Computing*, 17:412–426, 1988.

[71] Stephen Morris. Contagion. *Review of Economic Studies*, 67(1):57–78, 2000.

[72] Thomas Moscibroda, Stefan Schmid, and Roger Wattenhofer. On the topologies formed by selfish peers. In *Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing*, pages 133–142, 2006.

[73] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1996.

[74] Moni Naor and Kobbi Nissim. Communication preserving protocols for secure function evaluation. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 590–599, 2001.

[75] John Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[76] Luis E. Ortiz and Michael Kearns. Nash propagation for loopy graphical games. In *Advances in Neural Information Processing Systems 15*, pages 793–800, 2002.

[77] C. Papadimitriou. *Algorithmic Game Theory*, chapter The Complexity of Finding Nash Equilibria, pages 29–52. Cambridge University Press, 2007.

[78] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[79] Everett M. Rogers. *Diffusion of Innovations*. Free Press, 5th edition, 2003.

[80] Tim Roughgarden. *Selfish Routing and the Price of Anarchy*. The MIT Press, 2005.

[81] Thomas C. Schelling. *Micromotives and Macrobehavior*. W. W. Norton, 1978.

[82] John Skvoretz and David Willer. Exclusion and power: A test of four theories of power in exchange networks. *American Sociological Review*, 58(6):801–818, 1993.

[83] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[84] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 162–167, 1986.

[85] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2003.

[86] Jeff Zeleny. Working together, obama and clinton try to show unity. *New York Times*, 2008. Available online at http://www.nytimes.com/2008/06/28/us/politics/28unity.html.