UNIVERSITY OF CALIFORNIA, SAN DIEGO

Algorithmic Modeling of Decision Making over Networks

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Andrea Vattani

Committee in charge:

Professor Ramamohan Paturi, Chair
Professor Sanjoy Dasgupta
Professor Massimo Franceschetti
Professor Alex Snoeren
Professor Joel Sobel

2012

The dissertation of Andrea Vattani is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

____

____

Chair

University of California, San Diego

2012

DEDICATION

To my family

EPIGRAPH

Don't be such a hush-hush.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ALGORITHMS

ACKNOWLEDGEMENTS

I would first like to thank my parents who have always supported me in my long journey as a student from Italy to San Diego.

I wish to thank my advisor Mohan Paturi for his enthusiasm and energy, which has brought a lot of inspiration into my research. He gave me the freedom and time to explore research areas even outside my main PhD focus. I would also like to especially thank Sanjoy Dasgupta for his optimism, assurance and support. His positivity shined through since my very first year at UCSD, when he introduced me to the subject of clustering — an area that I still remarkably enjoy. I am thankful to Amos Israeli who visited UCSD in 2009: my meetings with him were extremely enjoyable and I had the opportunity to explore the power (and limitations) of distributed algorithms with an expert of the field.

During my PhD program I had the pleasure to collaborate with several wonderful people that I consider friends. Starting with my UCSD colleagues whom I've worked closely with: Manuel Cebrian, Lorenzo Coviello, Massimo Franceschetti, Moshe Hoffman, Daniel Lokshtanov, Petros Mol, Daniel Ricketts, Matus Telgarsky, Panos Voulgaris. Needless to say, all the merit is mine and all mistakes are theirs ... oops, or is it the other way around? It was very rewarding to work with all of you — as well as lots of fun.

I've spent two wonderful summers at Yahoo! Research, where I've met great people as well. First I would like to thank Ravi Kumar for being a great mentor. You've encouraged me to explore different areas of research and thus, have introduced me to several other researchers. Among them, a special thanks goes to Sergei Vassilvitskii who I deeply enjoy working with. I am also thankful to my fellow interns whom I've shared many great outings with during my two summers at Y! Research: Bahman, Ben, Javad, Markus, Silvio, and William.

I would next like to thank Flavio Chierichetti for being my constant contact for advice and with whom I spent so many hours thinking about problems with, as well as trying to convince him to eat meat. I am also thankful to Silvio Lattanzi for the time we spent together while at Yahoo!.

Last but not least – by any means – I would like to thank all of my friends,

both here and in Italy, who have made my life as enjoyable as it can get. A special thank you to Petros Mol for your dear friendship and sense of humor throughout our journey at UCSD. Another special thank you to Lorenzo Coviello who joined the club later, but fit right in. I've appreciated our friendship as well as your dinners. Finally, I'd like to thank Dan Ricketts for the nice time we've spent together on our shredder challenge and project. The most important thanks goes to my girlfriend, Jenny, who followed all my years at UCSD closely and supported me and made me happy more than anybody can ever want.

Chapter 2, in part, is a reprint of the paper "Low memory distributed protocols for 2-Coloring" co-authored with Amos Israeli, Mathew D. McCubbins and Ramamohan Paturi, published in the proceedings of the 12th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS 2010 [IMPV10]. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the paper "Human matching behavior in social networks: an algorithmic perspective" co-authored with Lorenzo Coviello, Massimo Franceschetti, Mathew D. McCubbins and Ramamohan Paturi, published in PLoS ONE, Volume 7, Number 8, August 2012 [CFM$^+$12]. The dissertation author and Lorenzo Coviello were the primary investigators and authors of this paper.

Chapter 4, in part, is a reprint of the paper "Finding red balloons with split contracts: robustness to individuals' selfishness" co-authored with Manuel Cebrián, Lorenzo Coviello, and Panagiotis Voulgaris published in the proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012 [CCVV12]. The dissertation author was the primary investigator and author of this paper.

Chapter 5, in full, is a reprint of the paper "The Effects of Diversity in Aggregation Games" co-authored with Petros Mol and Panagiotis Voulgaris published in the proceedings of the 2nd Symposium on Innovations in Computer Science, ICS 2011 [MVV11]. The dissertation author was the primary investigator and author of this paper.

Chapter 6, in full, is a reprint of the paper "Hiring a secretary from a poset" co-authored with Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii in the proceedings of the 12th ACM Conference on Electronic Commerce, EC 2011 [KLVV11]. The dissertation author was the primary investigator and author of this paper.

VITA

2007	Laurea in Computer Science Sapienza University of Rome, Italy
2012	Doctor of Philosophy in Computer Science University of California, San Diego, CA, USA

ABSTRACT OF THE DISSERTATION

Algorithmic Modeling of Decision Making over Networks

by

Andrea Vattani

Doctor of Philosophy in Computer Science

University of California, San Diego, 2012

Professor Ramamohan Paturi, Chair

The modeling and prediction of *collective* human behavior has been one of the key challenges of social sciences for several decades. In this dissertation, we use an algorithmic approach to study the behavior of *multiple* agents that interact with each other. A typical scenario considers a group of agents driven by both individual and collective incentives who communicate with each other through a *network* whose links represent potential interactions among them. We consider both *coordination* tasks, where the incentives of the agents are aligned, and non-coordination tasks, where their incentives are conflicting. The tasks we consider include social differentiation, selection of a reciprocating partner, and information aggregation. One of the key questions is whether the agents can solve these tasks. To address this question, we model agent behaviors algorithmically and analyze both individual and collective outcomes. For each task, we assess which algorithmic models will explain observed human behavior and which are more efficient in accomplishing the task. Finally, we study how the network structure affects outcomes and performance.

Chapter 1

Introduction

The modeling and prediction of collective human behavior has been one of the key challenges of social sciences for several decades. As early as 1947, Herbert Simon argued that *information processing* constitutes the core of human decision-making [SB76]. A corollary of his argument is that human decision-making processes can be modeled *algorithmically.* This also means that human computing abilities cannot overcome the computational barriers of the problem being solved. This was the view proposed by Valiant [Val84, Val00], who also argued that, despite the inherent complexity of human decision-making, it is possible to isolate basic principles and formulate simple mathematical models using an algorithmic approach. Such simple abstractions of human behavior can then lead to the design of computing devices and systems.

In this dissertation, we use an algorithmic approach to study the behavior of *multiple* agents that interact with each other. A typical scenario considers a group of agents driven by both individual and collective incentives who communicate with each other through a *network* whose links represent potential interactions among them. We consider both *coordination* tasks, where the incentives of the agents are aligned, and non-coordination tasks, where their incentives are conflicting. One of the key questions is whether the agents can solve these tasks. To address this question, we model agent behaviors algorithmically and analyze both individual and collective outcomes. For each task, we assess which algorithmic models will explain observed human behavior and which are more efficient in accomplishing the task.

Finally, we study how the network structure affects outcomes and performance.

Modeling a network of agents is challenging due to its complex nature. The complexity manifests itself on two different levels: first, the underlying structure of the network – who is connected to whom; second, the behavior of the agents. Even if the *individual* behavior is very simple, the *collective* behavior can be complex since the actions of an agent may have consequences for the actions of possibly every other agent in the network. Furthermore, in deciding his or her own actions, an agent may take into account that the outcome will depend on the network topology. As a result, modeling such networks may also involve the study of strategic behavior and reasoning of the agents.

It is often the case that the agents are unaware of the structure of the underlying network except for their immediate neighbors and perhaps the "friends of their friends". This means that the *decision* about which action to take often depends only on the local view of an agent. Nonetheless, the *effect* of an action can easily spread throughout the whole network and influence later decisions of agents faraway in the network. In other cases, the agents may not know the exact structure of the network but may have, as prior knowledge, a probability distribution over possible network topologies. In these cases, a strategic agent evaluates his or her own actions not only with the expectation that the collective behavior will depend on what he or she decides to do, but also with knowledge of the possible topologies surrounding him or her.

In analyzing the possible strategies the agents may adopt and the collective behavior they produce, it is important to distinguish the case where the agents are trying to *coordinate*, from the case where the agents have individual incentives that may conflict with each other. Even coordination may be difficult to achieve when agents have only a local view of the network. This limited knowledge about the topology may cause seemingly appropriate agent strategies to fail in achieving coordination efficiently. We will see an example of this phenomenon in Chapter 3, where each agent wants to select a mutually reciprocating partner among his or her neighboring parties.

There are cases in which the network of interest does not represent social or communication ties among the agents, yet it affects the behavior of the agents by providing, for instance, spatial constraints. A popular example is the problem of choosing a route through a network of highways. In this example, each agent wants to minimize the time to get to his or her destination with the expectation that the traffic congestion arising on a particular route is a result of the choices of all agents.

The objective of this dissertation is to provide an algorithmic perspective for networks of agents for the diverse situations and goals mentioned thus far. For each case, we provide algorithmic abstractions modeling the individual behavior of the agents. We assess whether their interaction by means of a network leads to the desired outcomes. Furthermore, we investigate what properties are required for a successful and efficient completion of the task. In particular, we explore the roles of the topology and of the incentives, and analyze how performance scales with the size of the network.

Chapters 2, 3, and 4 consider algorithmic models motivated and inspired by experiments that have shown the success of human subjects in solving a variety of tasks over a network. In these chapters, the network represents the social ties among the agents as well as their communication medium. Chapter 2 provides algorithmic models for the network coloring problem, which can be viewed as an abstraction of social differentiation. Our models are inspired by the human subject experiments reported in [KSM06, JKV10, MPW09, EMPW11]. Chapter 3 considers the matching problem, in which each agent needs to select a mutually reciprocating partner among the neighboring parties. Chapter 4 provides an algorithmic model of a large-scale experiment conducted in 2009 to assess the power of social networks in retrieving a rare piece of information. The scenario is competitive as agents have conflicting interests; however, we will show that the competitive aspect does *not* hinder the accomplishment of the task. In Chapter 5, the network represents a territory on which the agents move guided by their individual incentives. We consider the task of aggregation and show how the diversity of the agents is a necessary requirement to prevent negative effects. Finally, in Chapter 6 we study an instance of sequential decision making; a directed network among the agents defines who is preferred to whom and an external decision-maker needs to select one of the best agents as defined by the network.

An overview of the models and results is provided in Section 1.2. Before proceeding we review basic concepts and terminology.

1.1 Central fields

This dissertation uses ideas and concepts spread across different disciplines including distributed computing, game theory and graph theory. A reader familiar with such fields can proceed directly to Section 1.2.

Distributed computing

The field of distributed computing considers a network of autonomous agents that interact with each other to compute a *common* task. Different models of distributed computing have been proposed. We will focus on the message-passing model, where the agents can interchange messages through the links of the network and share no memory. The goal is for the agents to *collaboratively* compute some function of the communication graph. Common examples include problems such as computing a coloring, an independent set, or a matching of the communication network. The collaborative aspect of this field makes distributed algorithms amenable to model scenarios where the agents of the network are trying to achieve a common goal.

The efficiency of distributed algorithms is quantified in terms of the *communication* needed to compute the solution. Specifically, in the *synchronous* model, time is discretized into *communication rounds*, and in each round every agent can receive messages from the neighbors, perform some local computation, and send messages to the neighbors. The efficiency of a distributed algorithm is then defined as the *number of rounds* required to complete the task.

The number of rounds required by a distributed algorithm is often compared with the diameter D of the underlying network, i.e. the largest distance between any pair of agents. When no constraints are imposed on the algorithm, the challenge is to complete the desired task within a number of rounds that is much less than D. This means that the agents have to collectively compute a solution of the given problem with no information from distant parts of the network. For instance, in Chapter 3 we will demonstrate how a simple distributed algorithm can compute a maximal matching of any bounded-degree network of n nodes in only $O(\log n)$ communication rounds (regardless of the diameter D of the network, which could be as large as n).

It is easy to see that when no constraints on the algorithms are imposed, any computable problem can be solved in at most $2D$ communication rounds by gathering the whole topology of the network into a single node which can then locally compute the solution and broadcast it throughout the network. This approach however requires the algorithm to exchange large (poly(n) bits) messages and requires a large amount of memory for the agents (again poly(n) bits). When the size of the messages or the memory of the agents is restricted (usually to be either constant or $O(\log n)$), then devising a distributed algorithm that computes the desired solution in $O(D)$ communication rounds may be a challenge or even impossible. A famous example is the $\Omega(D + \sqrt{n}/\log n)$ lower bound on the number of communication rounds required to compute the minimum spanning tree of a network of diameter $D = \Omega(\log n)$ [PR00] when the message size is bounded by $O(\log n)$ bits.

Distributed algorithms and the techniques to analyze them will appear in many places throughout this dissertation, especially in the ideas and proofs of the algorithms presented in Chapters 2 and 3. For example, Chapter 2 explores different distributed algorithms for the network coloring problem. It is possible to show that $\Omega(D)$ communication rounds are necessary for any distributed algorithm that 2-colors a bipartite network. For this problem, we will provide round-optimal algorithms for rings and general bipartite networks which use constant and $O(\log n)$ bits respectively, for both memory and messages.

Game theory

Game theory provides models to study and predict the behavior of rational agents when the outcome not only depends on what actions they individually choose among several options, but also on the choices made by the other agents. A *game* models situations where two or more agents (usually called *players*) must simultaneously commit to a strategy and will receive a *payoff* that depends on the strategies chosen by all other players. The strategy of a player is said to be *best response* to a set of strategies if it produces the most favorable outcome for that player given the strategies of the other players.

A fundamental concept is the notion of *Nash equilibrium*: a collection of strategies, one per player, is a Nash equilibrium if no player has incentive to unilaterally change his or her own strategy knowing the strategies of all other players. Another way of expressing the same concept is that the players choose strategies that are best responses to each other. As we have already mentioned, we will consider scenarios in which agents are connected through a network but only have a prior distribution of the actual topology. In this case, the information that the players have about the game is *incomplete*. In such games, a collection of strategies is an equilibrium (more precisely, a Bayesian Nash equilibrium) if no player can increase his or her *expected payoff* by changing his or her action for any information — here the expectation is taken over the randomness of their prior distributions.

We will also consider sequential games. A sequential game differs from a classic game in that the decisions of the players are not taken simultaneously but sequentially. In this case, the decision of a player may depend on the past actions of the other players. The *best-response dynamics* with respect to a predetermined order of turns for the players is obtained when, for every $t \geq 0$, the player at round t chooses the strategy that would produce the most favorable outcome for the next round. This dynamics is often called *myopic* best response, as players do not consider what consequences choosing a strategy would have on future rounds in the game.

Finally, we will look into the more recent concepts of *price of anarchy* and *price of stability*. These concepts measure how the efficiency of a system degrades due to selfish behavior of its agents and are related to the notion of (in-)efficiency of equilibria [DR89, KP99, KP09]. Given a set of strategies, the *social welfare* refers to the sum of the payoffs of all agents. The *optimum welfare* is obtained when the social welfare is optimized; however, this may not be an equilibrium. The price of anarchy is then defined as the ratio between the optimum social welfare and the social welfare obtained in the *worst* equilibrium (worst with respect to its social welfare). The price of stability on the other hand is the ratio between the optimum social welfare and the social welfare obtained in the *best* equilibrium.

Graph theory

As we have mentioned, the network *structure* plays a fundamental role in the outcome of the system. Categorizing networks in terms of how “efficient” they are is a challenging task and such categorization also depends on the nature of the problem of the system.

We refer the reader to any graph theory book for basic notions of graphs. More specific concepts, such as graph generative models, are self-contained in this dissertation.

1.2 Central topics

This dissertation is divided into two parts. The first includes Chapters 2, 3 and 4, and the second includes Chapters 5 and 6. In the first part we focus on models inspired and motivated by experiments conducted on human subjects to solve a specific task. For all systems considered in this part, the network represents the social and communication ties among the agents. In the second part, we look at models for other phenomena: aggregation of entities and the decision-making in hiring a candidate. In the former, the network specifies spatial constraints for the agents, whereas in the latter the network defines preferences between agents.

1.2.1 Part I: Models for Networked Experiments

In the last few decades, there have been numerous experiments involving human subjects in order to better understand human behavior [Mil63, Mil74, Mil77, Zim08] as well as collective human capabilities in solving a task when incentivized to do so [Mil67, TM69, CE78, CEG83, LL96, DMW03, KSM06, JK08, KJTW09, MPW09, CJKT10, EMPW11, TCG$^+$11, PPR$^+$11].

For instance, the small-world experiment [Mil67, TM69, DMW03] revealed that individuals who only know the locations of their direct acquaintances can still, collectively, construct a short chain to a given target. An algorithmic perspective of such experiment was provided by Kleinberg [Kle00a, Kle00b] and led to new insights about social networks as well as efficient algorithms for distributed routing [Kle00b, Kle01, LNNK$^+$05, FG09, FG10, GS11]. Another example is the network bargaining experiments conducted by Cook et al. [CE78, CEG83], for which an algorithmic model of behavior [KT08a] and subsequent efficient distributed algorithms were developed [ABC$^+$09, Kan10, KBB$^+$11b].

We take a similar approach where we provide algorithmic models for experiments that have shown the success of human subjects in solving a variety of tasks over a network. This approach has several benefits. First, the proposed models of human behavior can be validated using the experimental data or by conducting subsequent experiments. Second, it allows analysis of the scalability of the proposed algorithmic models: such analysis suggests whether human subjects are able to solve the task at larger scales; in addition, it suggests which network topologies hurdle (and which are more amenable to) the accomplishment of the task. Furthermore, when the behavioral model is not efficient from an algorithmic perspective, it raises the question whether efficient algorithms exist at all for the problem under consideration — this question, when answered in the affirmative, can lead to new methods improving over the current ones.

Chapters 2 and 3 focus on highly controlled human subject experiments for the problems of network coloring and matching, respectively. Chapter 4 provides an algorithmic perspective on the DARPA network challenge, a large-scale experiment conducted in 2009 to assess the power of social networks in retrieving a “rare” piece of information. We proceed with an overview of each of these topics.

Coloring of networks

A coloring of a graph is an assignment of colors to nodes such that no two neighboring nodes are given the same color. The graph coloring problem can be viewed as an abstraction of social differentiation in which each agent prefers to distinguish his or her behavior from that of the neighboring parties. Examples are numerous and include selecting an expertise different from your nearby peers in an organization, selecting a phone ring-tone that differs from your friends' and colleagues', choosing rooms for overlapping activities.

The work in Chapter 2 is influenced by the human subject experiments conducted in [KSM06, JKV10, MPW09, EMPW11] in the context of distributed network coloring. They report that human subjects connected over a network can collectively obtain a coloring of the underlying network. In these experiments, each participant is a node of a virtual network and can interact with his or her neighbors through a computer interface. The subjects can only view the colors of their neighbors and change their colors as many times as they want. Subjects are paid if a coloring of the entire network is reached within an allocated time. The efficiency in reaching a coloring depends both on the network topology [KSM06, JKV10, EMPW11] and the incentives for the agents [JKV10, MPW09].

Chapter 2 provides an algorithmic perspective for these experimental results. We focus on 2-coloring of bipartite networks and coloring of preferential attachment graphs with minimum number of colors. Such networks encompass all topologies considered in the experiments of [KSM06, MPW09]. The main property of the model from a computational point of view a low-memory constraint on the algorithms. This constraint is justified since the participants of the experiments could only rely on their individual memorization skills in solving the task. We consider two algorithms that can be viewed as candidates for modeling human behavior for even rings and general bipartite networks — similar heuristics were also considered in [KSM06, JKV10] but no analysis was provided. Our analysis exhibits quadratic running times for such heuristics in the worst case but also suggests that the performance improves as the network is more uniformly connected. One one side of the spectrum, the ring requires quadratic time. On the other side of the spectrum, random r-regular graphs requires almost-linear time, for $r \geq 3$. Adversarially chosen dense graphs can lead to performance even poorer than the ring topology.

The worst-case performance of these algorithms raises the question whether low-memory efficient algorithms for 2-coloring exist. We answer this question in the affirmative providing (i) an optimal linear-time constant-memory coloring algorithm for rings that can also elect a leader, improving over previous leader election algorithms; (ii) a near-optimal algorithm for general bipartite networks. Finally, we provide a simple low-memory distributed algorithm for coloring preferential attachment graphs on n nodes within $\tilde{O}(\sqrt{n})$ many rounds and conjecture its actual worst-case running time to be only $O(\text{poly}(\log n))$.

Matching in networks

A matching in a network is a subset of the edges such that no two edges share a node. The matching problem is a natural abstraction of many human and organizational problems in which each agent needs to select a mutually reciprocating partner among the neighboring parties. As a specific scenario, consider the problem faced by faculty members pairing up with graduate students, with each student having to select exactly one advisor and each faculty member having to select exactly one student. We can view students and faculty members as the nodes of a (bipartite) network, with an edge between a faculty member and a student if such a pair shares research interests. The objective here is to find an assignment from faculty members to students sharing their interests so that number of matched pairs is maximized, thus a matching problem.

The work in Chapter 3 looks at an extensive series of matching experiments on human subjects through an algorithmic lens. The major difference with Chapter 2 — in addition to the task being solved, matching rather than coloring — is the fact that we were able to conduct the experiments ourselves which allowed us to validate the prediction of our model. We conducted two sets of experiments for a total of over 250 experiments over 80 networks with up to 24 nodes each, ranging from simple networks to more complex stochastic models including preferential attachment and small-world networks. As in the coloring experiments, the participants were connected over a virtual network and each participant could interact only with his or her neighbors through a computer interface that allowed him or her to repeatedly propose and accept partnerships with one of the neighbors. The participants were given an equal monetary reward for each game where a maximum matching was reached within the allotted time.

We isolated a simple behavioral property that we call "prudence" from the data collected in the first set of experiments. This property states that individuals do not break existing matched pairs unless they receive an alternative proposal by an unmatched neighbor. We developed an algorithmic model of human behavior based on this property. A complexity analysis of the proposed distributed algorithm highlights how an approximate maximum matching is reached quickly whereas reaching a maximum matching can take exponentially many rounds. Specifically, a $1/2$-approximation is reached in only $O(\log n)$ rounds for any bounded-degree graph, while for general graphs a $(1 - \epsilon)$-approximation is reached in polynomially many rounds (linearly many rounds are sufficient for any constant $\epsilon > 0$ in bounded-degree graphs). This is in agreement with the experimental data, which shows that human subjects always find a good matching quickly. We also provide a family of graphs, which we call "bad" graphs, on which the proposed model takes exponentially many rounds to reach a maximum matching. Computer simulations of our model also show that small-world networks are easier to solve than preferential attachment networks.

We conducted a second set of experiments to validate our algorithmic model. In particular, we tested the participants on the bad graphs, small-world networks and preferential attachment networks. The games on the bad graphs were never solved, consistent with the prediction of exponentially slow convergence. We remark that the bad graphs would have been solved easily, had the subjects used a different simple strategy. This further corroborates the thesis that the prudence property plays a fundamental role in the subject's decisions. Finally, we observed that the games over preferential attachment were much harder to solve than small-world networks, which is again consistent with the predictions of our algorithmic model and also in agreement with the coloring experiments of [KSM06].

Query information networks

Can a social network help retrieving the answer to a difficult query when given a limited budget? The Defense Advanced Research Projects Agency (aka DARPA), a research organization of the United States Department of Defense, designed in 2009 a so called "Network Challenge" that answered this question in the affirmative. The experiment consisted of locating ten red balloons placed at ten undisclosed locations in the United States. A $40,000 cash prize was allocated for the first participant to submit the correct coordinates of all ten balloons within the contest period. The common strategy was to recruit a team and use an adequate structure of economic incentives to encourage participation and coordination.

The MIT Media Laboratory team won the competition in less than 9 hours, adopting a recruitment scheme based on recursive incentives which succeeded in recruiting more than 4000 people [TCG$^+$11, PPR$^+$11]. Chapter 4 considers a generalization of such a recruiting scheme which we call "split-contracts", and analyzes it from a game-theoretical and computational point of view. The algorithmic and networking model we consider are an extension of the query incentive model introduced by Kleinberg and Raghavan [KR05] to study strategic games in decentralized information networks. Kleinberg and Raghavan considered a "fixed-payment" incentive scheme, as opposed to a recursive incentive scheme.

The model is specified by a network defined by a branching process, the rarity of answers to a specific question, and the desired probability of success. We ask: how much reward, as a function of the rarity of the answers, does the root node need to invest in order to retrieve an answer with the desired probability, when all of the nodes are playing strategically? For binomial branching processes and constant probability of success, Kleinberg and Raghavan showed that the "fixed-payment" scheme causes the reward function to exhibit a threshold behavior that depends on the branching factor of the network. In particular the root must invest a reward that is exponentially larger than the optimal one when the branching factor is too small.

In Chapter 4, we show that split contracts, unlike fixed-payments, are robust in a strategic environment in the sense that the reward the root must invest at equilibrium is at most a constant factor of the optimum. In other words, the price of anarchy of the game with split contracts is constant. Also, unlike the results in [KR05], our results hold for any branching process, and we are able to characterize the dependence of the investment with respect to the parameters of the branching process and the success accuracy.

1.2.2 Part II: Models for Other Phenomena

We now turn our attention to other kinds of decision-making over networks. In this part of the dissertation the network does not represent the social ties among the agents. In Chapter 5, the network's nodes represent locations and the edges represent vicinity between locations. In this case, the network plays the role of a shared resource among the agents and the combined actions of the agents populate such a network. In Chapter 6, we are given a directed network over n candidates encoding a partial ordering among them and an external decision-maker wants to hire an optimal candidate by interviewing them on a one-by-one basis.

Spatial-constrained systems

Aggregation of entities is a widely observed phenomenon in sociology, economics, biology and other fields, with examples including urban agglomeration, economies of agglomeration, insect swarms, and bacterial colonies. The Las Vegas phenomenon is another example of this; the city saw a rapid growth since the late 1940 when a few gaming centers moved there from Texas and Arkansas [DT90].

Chapter 5 provides a game-theoretical model for aggregation phenomena. Our formalization models a population of t agents inhabiting a world — represented as a network with n nodes — and measures aggregation by the number of edges induced by the nodes occupied by the agents. In other words, our model considers a competitive version of the densest t-subgraph problem. The game we consider is a sequential game with best-response dynamics, where in each round one of the agent can decide to move to a different location based on his or her payoff function.

We consider natural classes of strategies for the agents. In particular we consider "leader" agents — who have a tendency to "invest" by moving to high degree nodes in hope that other players will follow — and followers — who are more prudent and look for immediate rewards. We mainly focus on the price of anarchy of our games; that is, we compare the efficiency (i.e. aggregation) of the worst equilibrium to the aggregation provided by the optimum densest t-subgraph. Our analysis highlights the advantages of populations with diverse strategies (*heterogeneous* populations) over populations where all individuals share the same strategy (*homogeneous* populations). In particular, we prove that a simple heterogeneous population composed of *leaders* and *followers* achieves asymptotically lower price of anarchy compared to any homogeneous population, no matter how elaborate its strategy is.

Sequential decision-making

The secretary problem is a perfect example of sequential decision-making under uncertainty that has been studied in the fields of applied probability, statistics, decision theory and auction theory. The classical model considers a decision-maker willing to hire the best candidate among n rankable applicants for a secretary position. The applicants are interviewed sequentially in uniform random order and right after an interview, the decision-maker can either hire or reject the applicant. The decision is irrevocable and can only depends on the relative ranks of the applicants interviewed so far. The optimal algorithm is to skip the first $^n/_e$ applicants and hire the next arriving applicant who is the best seen so far, and achieves success probability of $^1/_e$.

Implicit in the classical setting is the assumption that there is a total order on the applicants, but this assumption rarely holds in real life since applicants often have incomparable attributes. Chapter 6 considers a generalization of the classical secretary problem in which the total order between the applicants is relaxed to a partial order among them, and the goal is to select any of the "maximal" applicants.

In other words, there is a directed acyclic network among the applicants which is initially unknown to the decision-maker. The interview of an applicant reveals the relationships between that applicant and all applicants interviewed so far and again the decision to hire or reject is irrevocable. We provide an algorithm that can be seen as an extension of the algorithm for the classical model which improves over previous work. It recovers a success probability of $^1\!/_e$ for any poset with a single maximal element while improving to a success probability of at least $k^{-1/(k-1)}(1 - O(\frac{\log^2 k}{k}))$ for posets with k maximal elements. We complement this result by showing that no algorithm can achieve better than $k^{-1/(k-1)}$ success probability.

Part I

Models for Networked Experiments

Chapter 2

Coloring Experiments

Social situations and tasks often require coordination among the agents. It is an interesting question how agents, who act rationally based on local considerations, can solve a task collectively. It is not obvious how to achieve coordination even between two cooperating agents as symmetry makes it difficult to break the tie. The apparent difficulty of coordination is amplified when we consider networks with hundreds, and even thousands of agents. Several experimental studies [Mil67, TM69, LL96, KSM06, JKV10, KJTW09, MPW09, EMPW11] have been conducted with human subjects in order to understand how agents can collectively solve a global task over a network.

In this chapter we will focus on a line of work that shows that human subjects connected over a network can successfully achieve coordination in the form of obtaining a *coloring* of the underlying network [KSM06, JKV10, MPW09, EMPW11]. Kearns, Suri and Montfort [KSM06] show that subjects in an experimental setting can indeed solve a distributed version of the graph coloring problem using only local information when they are incentivized to work collectively. Further they have clearly demonstrated that greater network connectivity will speed up the coordination. In these experiments, each subject is a node of the underlying network and can see the current colors of the neighbors. Each subject can change his or her own color as many times as he or she wants, and the subjects are paid a fixed amount if a proper coloring of the entire network is reached within an allocated time. Additionally progress of the coloring task as the percentage of edges properly colored is displayed to all subjects. The network topologies considered include bipartite (i.e., 2-colorable) graphs (for example, even rings with or without chords, two-leader graph, two rings connected as a cylinder), and preferential attachment graphs with parameter $k \in \{2, 3\}$ (with 2 or 3 neighbors initially added to each node).

McCubbins, Paturi and Weller [MPW09] extend the work of Kearns *et al.* by considering *asymmetric incentives*. Specifically, they consider 2-colorable graphs where one of the two colors is distinguished in that if a proper coloring is achieved, subjects with the distinguished color are paid an additional bonus. This setup can be viewed as a networked version of the two-player battle of the sexes game. The authors show that network connectivity is crucial for achieving coordination when cooperation (common incentives) is coupled with conflicting interests (asymmetric incentives). Agents over weakly connected networks were *not* able to solve the coloring problem with *asymmetric* incentives whereas they could solve the problem with symmetric incentives over the same network.

In a subsequent experimental study, Enemark, McCubbins, Paturi and Weller [EMPW11] consider 3-colorable graphs and show that the complexity of the underlying graph instance is an important determinant of the probability of solving the coloring problem. In particular, they have shown that adding edges to a 3-colorable graph while maintaining 3-colorability decreases the probability of solving the coloring game until the point the underlying graph has a unique coloring (up to isomorphism) and then additional edges increase the probability of solving the coloring problem.

This line of results is somewhat surprising given that human agents in the network have limited resources: in terms of memory, they can only rely on their individual memorization skills; in terms of knowledge of the network and communication, they just have a local perspective and can only interact with their immediate neighbors in the network. In this chapter, we seek to provide an algorithmic perspective for the experimental results. We ask whether there are "natural" strategies for solving the coordination problem in a feasible amount of time.

We are concerned with algorithms to color rings, general bipartite graphs, and preferential attachment graphs with minimum number of colors. All algorithms use low (often constant) memory and reach a solution in feasible (polynomial rounds) and sometimes optimal time. All the algorithms also have short message length and use a broadcast communication strategy. Our contributions include two simple algorithms RingGuess and GraphCoalescing for rings and general bipartite graphs, which can be viewed as candidates for natural human strategies. However, the quadratic time bounds for these algorithms call their scalability into question. We present two other algorithms RingElect and GraphElect which are optimal or nearly optimal in terms of the number of rounds (proportional to the diameter of the graph) but require somewhat more complex strategies. The question of finding simple algorithms in the style of RingGuess and GraphCoalescing that run in time proportional to diameter is open. Our algorithms provide a plausible basis for an argument that coordination in large distributed, social networks is feasible. As we leverage on common distributed computing strategies such as leader election or token management scheme, low memory algorithms for these problems arise.

2.1 Results

Our goal is to provide an algorithmic explanation for the success of the human agents in solving the coordination coloring problem over bipartite networks. We ask whether there exist algorithms that are natural in the sense of closely representing the experimental conditions. In particular, we ask whether there exist algorithms that only rely on local information, use small amount of memory, have no knowledge of the size of the underlying network, broadcast messages, and employ simple logic so they can be viewed as candidates for human strategies. We further ask whether such algorithms converge in feasible or optimal amount of time. More generally, what are the optimal algorithms for the graph coloring problem (for bipartite graphs and preferential attachment graphs) that require a small amount of memory?

Our results consist of constant/low memory algorithms to solve the 2-coloring problem on (even) rings as well as on general bipartite graphs, and to compute optimal colorings of preferential attachment graphs. We remark that these graphs encompass all the the topologies considered in the experiments of [KSM06, MPW09, EMPW11]. For the ring, we present two constant-memory algorithms. The first one RingGuess is extremely simple and converges in time quadratic in the size of the network. RingGuess can be viewed as a candidate for a natural human strategy. The second algorithm RingElect achieves optimality in terms of time (linear number of rounds) while only requiring constant memory. In addition, this algorithm can elect a leader in optimal time. To the best of our knowledge, optimal leader election algorithms, both in terms of time and memory, were not known for our setting[1]. We also model and investigate the slow-down caused by the asymmetric incentives in the experiments of [MPW09].

For general bipartite graphs with n nodes and m edges, we present an algorithm GraphCoalescing which can be viewed as a generalization of RingGuess. Each node u uses $O(\log \delta_u)$ bits of memory, where δ_u denotes the degree of u. GraphCoalescing is also very simple and is a candidate for a human strategy. For general graphs GraphCoalescing computes a 2-coloring in $O(nm^2 \log n)$ many rounds. For Δ-regular graphs the bound improves to $O(\Delta n^2)$. For random Δ-regular graphs the algorithm converges in only $O(n \log n)$ many rounds, suggesting the importance of a "globally connected" network. We then present the algorithm GraphElect which employs a leader election strategy as a tool to obtain a 2-coloring of the network. GraphElect requires up to $O(\log n)$ memory without requiring the knowledge of n. We show that GraphElect computes 2-coloring in $O(\log n + D)$ (essentially optimal) rounds where D is the diameter of the graph. Finally, for bipartite graphs, we discuss how asymmetric incentives can slow down the convergence of the algorithms.

For general graphs it is well-known that the k-coloring problem is NP-hard [GJ79] for $k > 2$. It is also hard to approximate [Kho01] even for centralized algorithms. Therefore, we cannot hope for efficient algorithms to color general

[1] See Section 6.3 of [San06] for a summary of leader election algorithms in the different settings. Also see Section 2.3 for details about our setting.

graphs for $k > 2$. However, for a class of graphs (generated by a preferential attachment rule) which are studied extensively in the context of social networks, we present an algorithm that converges in $O(\sqrt{n} \log^2 n)$ rounds where each node u uses $O(\log \delta_u)$ bits of memory.

2.2 Related work

The experimental study of human strategic behavior over networks is a topic of great current interest in the literature. The work by Kearns and others on network coloring and consensus games [KSM06, KJTW09, MPW09, JKV10, EMPW11] has been particularly influential. Judd et al. [JKV11] investigated how subjects choose between playing either a dominant or a submissive role in a network game, documenting the importance of fairness. Kearns et al. [KJV12] performed experiments on network formation games when there is a cost for creating links. Suri and Watts [SW11] conducted experiments in which individuals connected over networks play local public good games. Wang et al. [WSW12] studied multiplayer prisoner's dilemma games in which subjects can propose and delete links to other players, showing that partner selection increases cooperation. Brautbar and Kearns [BK11] introduced a network formation game in which players need to maximize their clustering coefficients.

As social interaction naturally induces strategic behavior, our work is also closely related to game theory. Indeed, several authors proposed game theoretical models of human interaction over social networks. Topics vary from diffusion and contagion over networks [MS09, GK12, AOY11] to strategic information retrieval [KR05, CCVV12], models of segregation [BIKK12] and bargaining over networks [KBB$^+$11a], to mention a few. The main element that distinguishes our work from the game theory literature is that we focus on the algorithmic processes involved in strategic thinking and the ensuing collective dynamics rather than on equilibria. Moreover, our algorithmic model is motivated and supported by experimental data.

The question of finding low-memory distributed algorithms has not been explored in the field of distributed computing and only very recently has received some attention [MS10, ESW12].

Recently, Mossel and Schoenebeck [MS10] have presented low memory algorithms for solving the consensus problem and a variation of it called the majority coordination problem. They analyze the consensus problem in social networks and parameterize social network computation by the amount of memory allocated to each node. Although their work is similar in spirit to ours in terms of the focus on low memory algorithms, their model diverges from ours in several aspects (see Section 2.3). Our focus on natural strategies calls for simplest possible algorithms.

The work by Chaudhuri, Graham and Jamall [CGJ08] is also motivated by the coloring experiments in [KSM06], but their setting is entirely different in that nodes never change their color. They show that a greedy strategy properly colors a network with high probability if the number of colors is at least $\Delta + 2$, where Δ is the maximum degree of the network.

2.3 Model

We consider the classical model of a synchronous message-passing distributed network. As in the experimental settings of [KSM06, MPW09], we consider anonymous and uniform networks. That is, nodes in the network do not have distinct identifiers and do not have knowledge of the size of the network (or other parameters such as diameter). Moreover, they all run the same algorithm. We note that, on the contrary, the models presented in [MS10] integrate the notion of advice[2], which correspond to some prior knowledge given to the nodes (usually a network parameter such as size of the network, diameter, their product, etc.).

Given our assumptions of anonymity and uniformity of the network, there exists no (even randomized) algorithm that computes a coloring of the network and *terminates* (this impossibility result about termination is famous for the leader

[2] In [MS10], advice is presented as a read-only memory, which does not count towards the memory requirements of the node. Computations involving advice that would typically require memory proportional to the size of the advice are viewed as constant memory operations (e.g., when taking actions with probability $\frac{1}{a}$ where a is the advice).

election problem, e.g. [AW04], and it extends easily to the coloring problem). However, we observe that the experiments in [KSM06, MPW09] do not require the human subjects to terminate but only to obtain a proper coloring within the allocated time. The algorithms in [MS10] are also not concerned with guaranteeing termination.

The choice of a synchronous network is motivated by the following discussion. The asynchronous setting in distributed computing is usually adversarial (to model communication delays) and focuses on the number of messages used by the algorithms rather than their time to converge. The asynchronous models introduced in [MS10] assign a rate to each edge (or node) of the network which determines the frequency at which the nodes operate[3]. While the asynchronous model in [MS10] is appealing since it embeds a notion of broadcast time, we believe that these models do not suit our goal of modeling plausible human strategies. A simple reason is that the frequency at which a node (human subject) operates does not solely depend on the edge of the network it is adjacent to, but rather on the execution of the algorithm itself, especially considering that nodes are not aware of the global structure. For example, a subject whose neighborhood changes is more likely to take an action than a subject whose neighborhood did not change. In the synchronous setting, every node is given the opportunity to take an action at every step, which is a better model for the distributed coloring experiments. In addition, synchronicity properly models the situation when two adjacent subjects take actions almost at the same time and need to break the tie.

We recall that for synchronous networks the running time of an algorithm is given by the number of communication rounds. In each round a processor receives messages from the neighbors, performs local computation, and sends messages to the neighbors. Finally, we remark that all our algorithms work in a simple broadcast model, where a node sends the same message to all the neighbors. The broadcast model is more suitable for capturing the setting of the coloring experiments in [KSM06, MPW09, EMPW11]. Indeed, in coloring experiments, the subjects are not allowed to communicate with each other except that they are allowed to

[3] Specifically, an edge “rings” with a probability that depends on its rate, and when it rings the nodes incident on it can operate and coordinate via shared randomness.

change their color and observe the changes in the color of a neighbor. Thus, a subject can only communicate with the neighbors by broadcasting a color change.

2.4 Ring networks

Rings are among the most studied topologies in distributed coloring experiments as well as in distributed computing. In this section we analyze two constant-memory algorithms for the ring topology. The first algorithm, RingGuess, is natural and is a plausible candidate for subject strategies in rings. A slight variant of this algorithm is also used in [KSM06] as a comparative tool with respect to human subject performance [4]. We show that RingGuess converges to a 2-coloring in $\Theta(n^2)$ rounds (in expectation) in a ring with n nodes. The algorithm does not involve any explicit message passing except that the each node has access to the color of its neighbors. Its message complexity defined as the total number of color changes by all the nodes is bounded by $O(n^2 \log n)$.

Our analysis of RingGuess raises the question whether there exists a constant memory algorithm that converges in linear number of rounds, which is clearly optimal for the ring. We present a new algorithm, RingElect, to 2-color a ring which uses constant memory and converges in $O(n)$ rounds. RingElect employs a leader election strategy, and also elects a leader within the same resource bounds.

At the end of this section, we discuss how asymmetric incentives will slow down the algorithms.

2.4.1 A natural algorithm

Consider a situation that frequently occurs in the experiments of [KSM06, MPW09, EMPW11] when at some point during the game a subject sees a neighbor choosing the same color as his/hers. In this situation the subject may change color

[4] In [KSM06] the algorithm bears the name *distributed heuristic* and works for any number c of colors. When restricted to the case $c = 2$ is essentially RingGuess but is used asynchronously. They simulate this algorithm on the networks used in human subject experiments to compare the steps required by the algorithm with the time to solve the problem by the subjects. No algorithmic analysis is provided in [KSM06].

or wait for the neighbor to change. One could conceivably use timing strategies to make the decision. However it is not possible to implement timing strategies in bounded memory and without the knowledge of the size of the ring. As such, a natural action is probably to change color with some probability, reverting the change if a neighbor changed its color as well.

With this motivation in mind, we introduce the RingGuess algorithm. Any node which has the same color as one of its neighbors repeatedly executes the following 2-round algorithm:

1. Change color with probability $p = \frac{1}{2}$, while remembering the old color and the colors of the two neighbors.

2. If any of the neighbors changes its color during the first round, restore the previous color.

Algorithm 2.1 presents pseudo-code for RingGuess.

Analysis

We now proceed with the analysis of the RingGuess algorithm. Let a *conflict* be an edge with nodes of the same color, and the distance between two conflicts be the minimum number of edges that separates them. We observe that since a node with no conflicts does not change color, and one with conflicts ends the 2-round algorithm with a different color only if its neighbors did not change color, the number of conflicts never increases. The 2-round RingGuess algorithm 'moves' the conflicts (clockwise or counterclockwise) with some probability. Also, when two conflicts have a node in common (i.e., 3 consecutive nodes have the same color), there is a probability of $p^3 = \frac{1}{8}$ that the two conflicts vanish – this happens when the middle node is the only one flipping its color.

The convergence proof of the algorithm will make use of random walks. The following lemma bounds the number of steps for a random walk to terminate.

Lemma 1. *Let $\mathcal{W}_k = (X_0, X_1, \ldots)$ be a unidimensional random walk on a path of nodes $1, 2, \ldots, k, \ldots$ starting on the the first node (i.e. $X_0 = 1$) and terminating*

Algorithm 2.1: RingGuess Algorithm

```
oldColor, oldColor$_{left}$, oldColor$_{right}$ $\leftarrow \perp$
hasConflict $\leftarrow$ $true$
color $\leftarrow$ random bit                              ▷ Begin with a random color
Broadcast color to neighbors                               ▷ Broadcast color at round 0

function RingGuess($p$, $t$, msg$_l$, msg$_r$)             ▷ Called at every round $t \geq 1$
    if hasConflict then
        if $t$ is odd then
            oldColor$_{left}$ = msg$_l$
            oldColor$_{right}$ = msg$_r$
            Flip color with probability $p$
        else
            if msg$_l$ $\neq$ oldColor$_{left}$ or msg$_r$ $\neq$ oldColor$_{right}$ then
                color $\leftarrow$ oldColor
            else
                hasConflict $\leftarrow$ (color $\in$ {msg$_l$, msg$_r$})
            Broadcast color to neighbors
```

when the k-th node is reached or surpassed. For $\delta \in \{1, 2\}$ and $j \geq 1$, consider the following transition probabilities:

$$\begin{aligned}
\mathbf{P}_{j\to j-\delta} &= \begin{cases} q_\delta & \textit{if } j-\delta > 0 \\ 0 & \textit{if } j-\delta \leq 0 \end{cases} \\
\mathbf{P}_{j\to j+\delta} &= q_\delta \\
\mathbf{P}_{j\to j} &= 1 - \sum_{\delta\in\{1,2\}} (\mathbf{P}_{j\to j-\delta} + \mathbf{P}_{j\to j+\delta})
\end{aligned}$$

where $\mathbf{P}_{i\to j} = \Pr[X_{t+1} = j | X_t = i]$, $q_1, q_2 > 0$ and $2(q_1+q_2) \leq 1$. Then the expected time for $\mathcal{W}_k$ to terminate is at most $\frac{k^2}{q_1+4q_2}$.

Proof sketch. Let h_i be the expected value of the random variable representing the number of steps to reach (or surpass) state k from state i. Then the following system $\mathcal{S}^*$ of equations holds

$$
\begin{aligned}
&h_1 = c_1 + q_1 h_2 + q_2 h_3 + (1 - q_1 - q_2) h_1 \\
&h_2 = c_2 + q_1(h_1 + h_3) + q_2 h_4 + (1 - 2q_1 - q_2) h_2 \\
&h_k = 0 \\
&h_{k+1} = 0 \\
&h_j = c_j + q_1(h_{j-1} + h_{j+1}) + q_2(h_{j-2} + h_{j+2}) + (1 - 2q_1 - 2q_2) h_j, \quad 3 \le j \le k-1
\end{aligned}
$$

with $c_j = 1$, for $1 \le j \le k-1$.

Recall that our goal is to show that $h_1 \le \frac{k^2}{q_1+4q_2}$. Let $h_1 = x_1^*, h_2 = x_2^*, \ldots, h_{k+1} = x_{k+1}^*$ be the solution of this system S^*, and let $h_1 = \tilde{x}_1, h_2 = \tilde{x}_2, \ldots, h_{k+1} = \tilde{x}_{k+1}$ be the solution of the system $\tilde{\mathcal{S}}$ obtained by setting $c_1 = 1$, $c_j = 2$, for $2 \le j \le k-1$, and replacing equation $h_k = 0$ with $h_k = \frac{k^2-(k-1)^2}{q_1+4q_2}$. We observe that it has to be $\tilde{x}_j \ge x_j^*$ for any $1 \le j \le k+1$. Using induction, we show $\tilde{x}_j = \frac{k^2-(j-1)^2}{q_1+4q_2}$ for $1 \le j \le k+1$. From this, we conclude that $h_1 = x_1^* \le \tilde{x}_1 = \frac{k^2}{q_1+4q_2}$. □

We apply Lemma 1 to bound the number of rounds required for two conflicts in a ring to come close to each other. This will be the main ingredient to establish the main theorem.

Lemma 2. *Consider any 2-coloring with m conflicts such that no conflicts are at distance less than 2. Then, the expected number of rounds for two conflicts to be at a distance less than 2 is at most $\frac{n^2}{p^2 m^2}$.*

Proof. We observe that after an execution of the 2-round algorithm, any conflict (independently of the others) will move one edge clockwise with probability p^2, one edge counterclockwise with probability p^2, and it will not change position with probability $1 - 2p^2$.

Fix two consecutive conflicts and let D be a random variable representing the distance between them. After an execution of the 2-round algorithm, D will (a) increase by 2, as well as decrease by 2, with probability $(p^2)^2 = p^4$; (b) increase by 1, as well as decrease by 1 with probability $2p^2(1-2p^2) = 2p^2 - 4p^4$; (c) not change with probability $1 - 2(p^4 + (2p^2 - 4p^4)) = 1 - (4p^2 - 6p^4)$.

As $D \leq n$, we have that the expected number of rounds for D to be less than 2 is no larger than the expected time for the random walk $\mathcal{W}_n$ from Lemma 1 to terminate when $q_1 = 2p^2 - 4p^4$ and $q_2 = p^4$. Lemma 1 assures that the expected number of rounds for D to become less than 2 in at most $\frac{n^2}{q_1+4q_2} = \frac{n^2}{p^2}$.

To prove the lemma, we will show that the expected number of rounds for two conflicts among m to be at a distance less than 2 is no larger than the expected time for a random walk $\mathcal{W}_{\lfloor n/m \rfloor}$ to terminate. Consider the Markov chain $\bar{D}_0, \bar{D}_1, \ldots$, with $\bar{D}_t = (D_t^{(1)}, D_t^{(2)}, \ldots, D_t^{(m)})$, where $D_t^{(i)}$ is the random variable representing the distance between the i-th and $(i+1)$-st conflict on the ring at time t. $D_t^{(m)}$ represents the distance between the last and the first conflict at time t. We couple this Markov chain with another one, $M_t = \min_i D_t^{(i)}$, that keeps track of the distance between the closest pair of conflicts for $t \geq 0$. Now observe that $M_t \leq \left\lfloor \frac{n}{m} \right\rfloor$ and that M_t will take on a value less than 2 at least as fast as the random walk $\mathcal{W}_{\lfloor n/m \rfloor}$ terminates. This observation along with Lemma 1 concludes the proof. □

We are now ready to prove the main theorem for this section.

Theorem 3 (RingGuess). *The expected number of rounds for* RingGuess *to compute a 2-coloring of the ring is $\Theta(n^2)$. Its expected message complexity is $O(n^2 \log n)$.*

Proof. For $i = 1, \ldots, n$, let X_i be the random variable denoting the number of rounds between the moment the $(i-1)-th$ conflict is resolved and the moment the $i-th$ conflict is resolved.

Lemma 2 along with the observation that there is a constant probability that two conflicts at distance less than 2 vanish, imply that $E[X_i] \leq c\frac{n^2}{p^2(n-(i-1))^2}$, for some constant $c > 0$.

We are interested in the number of rounds to resolve all conflicts, that is $X = \sum_{m=1}^{n} X_i$. We conclude that the expected number of rounds to resolve all the conflicts is bounded by

$$E[X] = \sum_{i=1}^{n} E[X_i] \leq \sum_{m=1}^{n} c\frac{n^2}{m^2} = O(n^2).$$

Analogously the number of messages (or color changes) is bounded by $\sum_{m=1}^{n} 2cm\frac{n^2}{m^2} = O(n^2 \log n)$ since the expected number of color changes in a configuration with m conflicts is $2m$. □

We remark that the analysis is tight, as the expected number of rounds it takes for two conflicts initially at distance $\Omega(n)$ to get a distance less than 2 is $\Theta(n^2)$.

2.4.2 An optimal algorithm

The quadratic time bound obtained for RingGuess calls its scalability into question. It is natural to ask whether we can even achieve the optimal in both worlds, that is whether there exists an algorithm that obtains a 2-coloring of a ring in linearly many rounds using only constant memory.

In this section we answer this question in the affirmative with the algorithm RingElect. The algorithm elects a leader and a 2-coloring of the ring easily follows. Hence, we also provide an optimal algorithm to elect a leader in a ring using constant memory and linearly many rounds.

We now proceed with the description of the algorithm. For simplicity, we describe RingElect in a more restrictive model, where local orientation is assumed. Specifically, a node is capable of sending a message only to a specific neighbor, and on reception of a message can distinguish which of its neighbors sent that message. Later we will explain how this assumption can be removed. The high-level picture of the algorithm RingElect is the following. We begin in a configuration where all nodes are leaders. Each leader plays a leader election on each of the segments of the ring connecting it to its clockwise-next and counterclockwise-next leaders. A leader losing in both its segments becomes slave and sends a message notifying that it is conceding the election to the leaders at the end of its segments. A concede message has the extra function of aborting all the election messages encountered whilst traversing the segment.

A detailed description of the algorithm follows. As mentioned before, we consider two types of messages, *concede* and *contest*. Each message `msg` has a field `msg.type` to denote its type. Concede messages are sent by nodes who became slaves. Contest messages are generated by leaders (a) during the first round, (b) on reception of a concede message, and (c) on reception of a contest message. A contest message `msg` carries a bit `msg.b` indicating the position of the leader who sent it and a "history" $\texttt{msg.h} \in \{\star, 0, 1\}$ indicating what generated it: in cases (a) and (b) the history is set to $\star$; in case (c) the history is set to the election bit contained in the received contest message.

Each node has the following local variables: $\texttt{status} \in \{\textit{start,leader,slave}\}$ (initialized with *start*); for $i \in \{l, r\}$, $\texttt{msg}_i$ to remember the latest message received from direction i; $\texttt{b}_i$ to store the random bit used for the election on the segment in direction i; $\texttt{losing}_i$ to remember whether it is "losing" the election on the segment in direction i. Every node runs the algorithm described by Algorithm 2.2.

Analysis

We say that the portion of the ring between nodes u and v is a *segment* at round t if at round t, u and v are leaders and all other nodes between u and v are slaves. At the beginning we have exactly n segments. As the algorithm progresses, the number of segments goes down and the segments get larger. When all segments vanish, only one leader is left.

A message `msg` is *crucial* if $\texttt{msg.type} = \textit{contest}$, $\texttt{msg.b} = 1$ and $\texttt{msg.h} = 0$. Similarly, a message variable $\texttt{msg}_i$ of a leader node is *crucial* at the end of round t if it holds a crucial message at the end of round t. For a segment with left-leader u and right-leader v, the variables of the segment are the variable $\texttt{msg}_r$ of u and the variable $\texttt{msg}_l$ of v. By construction of the algorithm, a leader becomes a slave only if both its variables are crucial.

We will say that a message m is on a segment at the end of round t, if during round t a node u of that segment sent m to another node v of that segment. The direction of a message is defined by who sent the message and who received it. We say that two messages m from u to v and m' from u' to v' on a segment are *converging* (resp. *diverging*) if a path $u, v, \ldots, v', u'$ (resp. $v, u, \ldots, u', v'$) is in the segment. Finally, for a segment of leaders u and v, and for a message m on the segment directed toward u, any message in between m and v is said to be *behind*

Algorithm 2.2: RingElect Algorithm

```
status $\leftarrow$ $leader$, $\mathtt{b}_l \leftarrow \bot$, $\mathtt{b}_r \leftarrow \bot$, $\mathtt{msg}_l \leftarrow \bot$, $\mathtt{msg}_r \leftarrow \bot$
for $i \in \{l, r\}$ do                              ▷ In the first round, start the election
    Election($i$)
function RingElect($t$, $\mathtt{new}_l$, $\mathtt{new}_r$) ▷ Messages received at round $t$, may be $\bot$
    if status = $leader$ then
        for $i \in \{l, r\}$ do
            if $\mathtt{new}_i \neq \bot$ then
                $\mathtt{msg}_i \leftarrow \mathtt{new}_i$
                $\mathtt{losing}_i \leftarrow (\mathtt{msg}_i.\mathtt{type} = contest$ and $\mathtt{msg}_i.\mathtt{b} > \mathtt{b}_i$ and $\mathtt{msg}_i.\mathtt{h} = 0)$
        if $\mathtt{losing}_l$ and $\mathtt{losing}_r$ then            ▷ Election is lost, concede
            status $\leftarrow$ $slave$
            Send concede message to both neighbors
        else                  ▷ Election is still on, reply with new contest messages
            for $i \in \{l, r\}$ do
                if $\mathtt{new}_i \neq \bot$ then
                    Election($i$)
    else if status = $slave$ then      ▷ Concede messages abort contest messages
        Forward $\mathtt{new}_l$ to $r$ unless $\mathtt{new}_l.\mathtt{type} = contest$ and $\mathtt{new}_r.\mathtt{type} = concede$
        Forward $\mathtt{new}_r$ to $l$ unless $\mathtt{new}_r.\mathtt{type} = contest$ and $\mathtt{new}_l.\mathtt{type} = concede$
function Election(i)
    $\mathtt{b}_i \leftarrow$ random bit
    if $\mathtt{msg}_i = \bot$ or $\mathtt{msg}_i.\mathtt{type} = contest$ then
        $\mathtt{h} = \star$
    else
        $\mathtt{h} = \mathtt{msg}_i.\mathtt{b}$
    Send new msg to $i$ with $\mathtt{msg.type} := contest$, $\mathtt{msg.b} := \mathtt{b}_i$, $\mathtt{msg.h} = \mathtt{h}$
```

m.

We now proceed with a detailed characterization of the behavior of the algorithm. We start with a couple definition.

Definition 4 (Safe configuration). *A segment is in a safe configuration at the end of round t if the following properties hold at the end of round t.*

(i) *There are one or two concede messages on the segment.*

(ii) *No variable of the segment is crucial.*

(iii) *Every crucial contest message on the segment is converging to a concede message on the segment.*

(iv) *If there are two concede messages on the segment then they are diverging and no other message is in between them. If there is only one concede message on the segment, then there can be only one message behind it. This message is non-crucial and is traveling in the same direction as the concede message.*

Definition 5 (Contest configuration). *A segment is in a contest configuration at the end of round t if the following properties hold at the end of round t.*

(a) *There are exactly two contest messages and no concede messages on the segment.*

(b) *At most one variable of the segment is crucial.*

(c) *Crucial messages on the segment travel in the same direction. Also if a variable of the segment is crucial, crucial messages travel toward the leader holding that variable.*

The following lemma offers a characterization of the behavior of the algorithm.

Lemma 6. *At the end of any round $t \geq 1$, any segment is either in a safe configuration or in a contest configuration.*

Proof. We prove the lemma by induction on t. During round $t = 1$ every node sends a non-crucial contest message per segment, therefore at the end of the round there will be two non-crucial contest messages (history of messages is set to $\star$) per segment, and no leader has crucial variables. Hence, at the end of round $t = 1$ each segment is in a contest configuration.

By induction suppose that the lemma holds at the end of round t, and consider any segment. First suppose that the segment is in a safe configuration at the end of round t. If no concede messages are received by any of the two leaders at the beginning of round $t + 1$, there will be a safe configuration at the end of round $t + 1$ as no other messages are sent. Otherwise a leader receiving a concede message at the beginning of round $t + 1$, will send a non-crucial contest message on the segment (the history of the message is set to $\star$). Therefore at the end of round $t + 1$ the segment will be either in a safe configuration (if at least a concede message is present) or in a contest configuration (if no concede messages are left).

Now suppose that the segment is in a contest configuration at the end of round t. Consider first the case when no leader of the segment becomes a slave during round $t + 1$. We will show that at the end of round $t + 1$ there will be a contest configuration. Note that property (a) holds at the end of round $t + 1$ because a slave receiving a contest message will forward it, while a leader receiving a contest message will send another contest message on the segment. Property (b) holds at the end of round $t + 1$ because property (c) guarantees that only one leader can receive a crucial message at the beginning of round $t + 1$, and again by (c) the other leader has no crucial variables. Property (c) holds at the end of round $t + 1$ because a leader that receives a crucial message at the beginning of round $t + 1$, will send a non-crucial contest message on the segment (the history of the message will be 1 since the received message is crucial).

Finally, consider the case when a leader of the segment becomes a slave during round $t + 1$. A leader becomes a slave only if both its variables are crucial. Hence, property (b) implies that only one leader of the segment can become a slave during round $t + 1$. Let u be this node, and let v and w be the other leaders of the two segments of u. If $v = w$ then we are left with only one leader and no segments.

Thus, assume $v \neq w$. Since u becomes a slave during round $t+1$, it must be the case that both segments of u are in a contest configuration at the beginning of round $t+1$ (if not, one of u's variable would not be crucial). We will prove that at the end of round $t+1$ the new segment defined by the leaders v and w will be in a safe configuration. Property (i) and (iv) are trivial since u will send two concede messages on its two sides. By property (b) both v and w have non-crucial variables at the end of round t, and since all crucial contest messages are traveling toward u by (c), properties (ii) and (iii) will hold at the end of round $t+1$. □

From Lemma 6, property (ii) of safe configurations and properties (b)-(c) of contest configurations, we obtain the following corollary.

Corollary 7. *There is always at least one leader.*

We are now ready to prove our main theorem.

Theorem 8 (RingElect)**.** *The algorithm* RingElect *elects a leader and computes a* 2*-coloring in* $O(n)$ *many rounds and* $O(n \log n)$ *bit-complexity, both quantities in expectation.*

Proof. Consider any round and let $\ell \geq C$ be the number of leaders, where C is a sufficiently large constant. For a leader u, let the *scope* of u be the union of the two segments of u (notice that scopes are not disjoint and each scope contains exactly 3 leaders.) Let S be a maximal set of node-disjoint scopes. Then it has to be $|S| \geq \lfloor \frac{\ell}{3} \rfloor \geq \frac{\ell}{4}$, where the last step follows if C is large enough.

Observe that at least $|S| - \frac{\ell}{8} \geq \frac{\ell}{8}$ scopes of S are *short*, that is they have length at most $\frac{8n}{\ell}$. If not, the remaining scopes of T would contain more than n distinct nodes. Consider any such short scope. After a phase of $\frac{16n}{\ell}$ rounds either at least one of the three leaders has become a slave or all the nodes have drawn new bits for the elections on the two segments of the scope. Also, in each of these phases there is a constant probability q that one of the leaders of the scope becomes a slave. Fix any $m \geq 1$. For $1 \leq j \leq \frac{\ell}{8}$, let Z_j be the indicator random variable of the event that the j-th short scope has no leader become a slave after m phases. Let Z be the number of short scopes such that no leader becomes a slave after m phases, and L be the number of leaders left in total after m phases. As each scope has exactly 3 leaders, we have $L \leq \ell - (\frac{1}{3}\frac{\ell}{8} - 3Z)$. Then $E[Z_j] = \Pr[Z_j = 1] = (1-q)^m$, and $E[\sum_j Z_j] = \sum_j E[Z_j]\frac{\ell}{8}(1-q)^m \leq \frac{\ell}{8}e^{-m}$. Also, as the scopes are node-disjoint, the random variables Z_j are independent. By Chernoff bound, $\Pr(Z \geq (1+\delta)E[Z]) \leq \exp(-\delta^2 E[Z]/3)$. By choosing m as a sufficiently large constant and δ as a sufficiently small constant, we have that $L \leq \frac{\ell}{c}$ with probability at least 1/2, for some constant $c > 1$.

For $0 \leq i \leq \log_c(n)$, let E_i be the event that at most n/c^i leaders are left. The above argument shows that given E_i at some round t, the probability of E_{i+1} at round $t + m\frac{16n}{(n/c^i)} = t + 16mc^i = t + O(c^i)$ is at least 1/2. Let X_{i+1} be the number of phases of length $16mc^i$ until E_{i+1} happens. Then $E[X_i|E_i] = \sum_{j=0}^{\infty} j \Pr(X_i = j|E_i) \leq \sum_{j=1}^{\infty} j(1/2)^{j-1} = O(1)$. We can conclude that, in expectation, the number of rounds to reduce the number of leaders down to a constant is at most $E[\sum_{i=0}^{\log_c(n)}(16mc^i)X_i] = O(\sum_{i=0}^{\log_c n} c^i) = O(n)$. It is easy to see that in expectation linearly many rounds are sufficient to reduce the number of leaders from a constant to one, as every $O(n)$ rounds a leader has a constant probability to become a slave.

A similar analysis shows that the expected number of messages is $O(n \log n)$.

Finally, we explain how a 2-coloring can be achieved. In the first round every node chooses a color for itself. Every time that a leader receives a concede message, it will start to propagate its coloring. A slave who receives two non-compatible colorings by its two neighbors will not propagate any of the two. When only one leader is left, the coloring of this leader will be propagated through the entire network (in linearly many rounds). □

Broadcast model

We will briefly describe how to modify the algorithm RingElect so that only broadcast is used. We will still assume that when a node receives a message, it can distinguish which neighbor broadcast it[5].

[5] This is a natural assumption. For example, this assumption holds for the coloring experiments in [KSM06, MPW09, EMPW11].

We will use the key property is the fact that a slave never receives two messages coming from the same neighbor in two consecutive rounds. (This property can be shown to hold inductively.) Using this property we can modify the algorithm as follows. A slave node u will accept (and therefore broadcast) a message m broadcast from a slave neighbor v iff at least one of these conditions holds: (i) in the previous round u did not accept m from the other neighbor; (ii) v is broadcasting two messages (this happens only if in the previous round v accepted the message m from u and a message m' from its other neighbor; in this case, if $m \neq m'$, u knows what message to ignore, otherwise u will accept any of the two). A slave node u will accept a message m broadcast from a leader neighbor w iff in the previous round u broadcast a message received by the other neighbor. Similar rules can be used for leader nodes. The only major modification is the following: when a leader accepts two messages in the same round (coming from the two different segments) and does not become a slave, it will draw only one bit and use (broadcast) it for both segments. This modification of the election process does not affect the performance of the algorithm which will still converge in linear time.

2.4.3 Slow-down with asymmetric incentives

As mentioned in the introduction, the experiments in [MPW09] introduces asymmetric incentives for bipartite graphs in the following way. If a proper coloring is achieved, the participants ending with a specific color are paid more than the participants ending with the other color. The effect during the game is that participants are reluctant to leave the “special” color for the other one.

We wish to quantify the influence of these asymmetric incentives on our algorithms. We model “selfish” participants in the following way. We say that a node is Byzantine if, when supposed to give up the special color or become a slave, it does so with probability q strictly less than one. Now consider a ring with (at least) two Byzantine nodes at odd distance $\Omega(n)$. (Note that if we place two Byzantine nodes at random on the ring, this situation will happen with constant probability.) Then, with proofs similar to the ones presented, it is possible to show that the convergence time of the algorithms gets slower in the following way:

RingGuess will converge in $\Theta(\frac{n^2}{q^2}+\frac{1}{q^3})$ time; and RingElect will converge in time $O(\frac{n}{q})$.

An interesting aspect caused by the requirement on constant memory, is that detection of Byzantine nodes is impossible for the other nodes.

2.5 General bipartite graphs

We now turn our attention to general bipartite graphs. First we present a simple algorithm that computes a 2-coloring of any bipartite graph in poly(n) time. Each node v uses an amount of memory proportional to the logarithm of its degree. Secondly, we show that using a little more memory per node, namely $O(\log n)$ bits, we can obtain essentially optimal convergence time.

2.5.1 A natural algorithm

Without loss of generality let the color palette be $\{0, 1\}$. Consider the following simple algorithm that we call GraphCoalescing.

In the first round every node chooses a random color b and sends $\bar{b}$ to a random neighbor. In every round $t \geq 2$, each node u receiving at least one bit performs the following operations:

(a) chooses at random a bit b among the received bits;

(b) colors itself with b; and

(c) with probability $\frac{\delta_u}{\delta_u+1}$ sends $\bar{b}$, the complement of b, to a random neighbor and with the remaining probability sends b to itself.

Observe that each node u in the algorithm uses $O(\log \delta_u)$ bits of memory to select a random neighbor.

The idea of this algorithm is that every node proposes a coloring. Each proposal takes a random walk on the graph, and when two (or more) proposals meet, only one of them will survive (the proposals will *coalesce* into one) and continue its random walk. Now suppose that at some point only one proposal is left:

then it will walk randomly through the network, and will lead to a proper coloring once all nodes have been visited. Viewing proposals as tokens, it follows that the algorithm can also be used to provide a token management scheme (See [IJ90] for a similar approach).

The following theorem borrows heavily from the literature.

Theorem 9 (GraphCoalescing)**.** *The algorithm* GraphCoalescing *computes a* 2*-coloring in* $O(m^2 n \log n)$ *many rounds for general bipartite graph with* m *edges,* $O(\Delta n^2)$ *many rounds for* Δ*-regular bipartite graphs, and* $O(n \log n)$ *many rounds for random* Δ*-regular bipartite graphs.*

Proof. In the proof we refer to each proposal as a particle. Let G be a (bipartite) graph. We observe that part (c) of the algorithm implies that each particle is performing a random walk on the graph G' that is obtained from G by adding self-loops to each node. Therefore, since G' is not bipartite, the random walk of each particle is aperiodic. The expected number of rounds required for coloring the graph is bounded by the expected number T_{coalesce} of rounds for the particles to coalesce to a single particle, plus the cover time T_{cover} (that is, the expected number of rounds for a single particle to visit all the nodes). A classic result in [AKL$^+$79] shows that the cover time of a graph is $T_{\text{cover}} = O(mn)$. By [AF99, Section 14.3], we have that $T_{\text{coalesce}} = O(\Delta n^2)$ for Δ-regular graphs, and $T_{\text{coalesce}} = O(T_{\text{cat\&mouse}} \log n)$ for general graphs, where $T_{\text{cat\&mouse}}$ is the time required for two random walks to meet. For non-bipartite graphs it is well-known that $T_{\text{cat\&mouse}} = O(m^2 n)$. The bound for random Δ-regular bipartite graphs follows from the bound in [CFR09] for coalescing particles and the bound on the cover time in [CF05]. □

We observe that the algorithm (as it is) is not suitable for a broadcast model because nodes must be able to send messages to a specific neighbor. This issue can be addressed using the isomorphism between the coalescing particles process and the voter model. In the voter model each node starts with an opinion (a proposal of a coloring in our case). As time passes, nodes modify their opinions in the following way. At each step, each node changes its opinion to the opinion of a random neighbor or stick to its opinion where all the options are equally probable.

It is known that the expected time for only one opinion to survive (the number of opinions can only decrease with time) is the same as the expected time for all the particles to coalesce (e.g. see [CFR09]). This observation easily leads to a broadcast algorithm with the same guarantees.

2.5.2 An optimal algorithm

In this section we present GraphElect, an algorithm that uses $O(\log n)$ memory in expectation and computes a 2-coloring of any bipartite graph in $O(D + \log n)$ expected number of rounds, where n and D are size and diameter of the graph respectively. Any distributed algorithm that 2-colors general bipartite graphs requires $\Omega(D)$ rounds: therefore GraphElect is optimal for graphs of diameter at least $\Omega(\log n)$.

We now describe the algorithm GraphElect. At any given stage, a processor can be either a leader or a slave. At the beginning of the algorithm all processors are leaders. Each processor presents to its neighbors a variable `Leading-Rank` (initial value 0), and its color (initial value either 0 or 1). In addition, each processor keeps locally a variable `Rank` (initial value 0). At the beginning of each round, a processor reads the state variables of all its neighbors and computes the maximal leading rank among all its neighbors. The processor holding that maximal leading rank is the processor's leading neighbor. If there is more than a single processor holding the maximal leading rank, the leading neighbor is elected arbitrarily from among the leading processors. If the maximal leading rank is larger than the processor's own rank, the processor adjusts its color to be the opposite color of its leading neighbor, and becomes a slave (if it was a leader). A slave keeps doing this simple routine forever and never gets a chance to become a leader again.

In addition to all aforementioned variables, a leader also holds a timer whose initial value is zero. The nodes counts down from timer value to zero. When the count goes to 0, if the processor is still a leader, it increments its rank and leading rank by 1. Then it updates its timer value to be twice the old timer value, plus a random value in $\{0, 1\}$.

We now proceed with the analysis of GraphElect.

Lemma 10. *Let u and v two nodes in the graph. If, at the beginning of a certain round, u is still a leader and its rank is greater than the rank of v, then for the rest of the computation there will be some node (possibly u) with rank greater than the rank of v.*

Proof. Let $W_k^{\langle u\rangle}$ be the value of the timer of u right after the k-th update. $W_k^{\langle u\rangle} = 2\cdot W_{k-1}^{\langle u\rangle} + B_k^{\langle u\rangle}$ (as long as u is a leader), where the $B_k^{\langle u\rangle}$'s are i.i.d. random variables taking values from $\{0,1\}$. We have that $W_k^{\langle u\rangle} = \sum_{i=0}^{k} 2^{k-i} B_k^{\langle u\rangle}$. Now consider the first round t^* when the rank of u is greater than the rank of v. During round t^*, u must have updated its timer, and let this one be its k-th update. It must be the case that $W_j^{\langle u\rangle} = W_j^{\langle v\rangle}$ (and therefore $B_j^{\langle u\rangle} = B_j^{\langle v\rangle}$) for all $j < k-1$, and $W_{k-1}^{\langle u\rangle} < W_{k-1}^{\langle v\rangle}$ (and therefore $B_{k-1}^{\langle u\rangle} < B_{k-1}^{\langle v\rangle}$). Now consider the k-th update for u and v. We have that

$$W_k^{\langle u\rangle} = 2W_{k-1}^{\langle u\rangle} + B_k^{\langle u\rangle} \leq 2W_{k-1}^{\langle u\rangle} + 1 \leq 2(W_{k-1}^{\langle v\rangle} - 1) + 1 < 2W_{k-1}^{\langle v\rangle} \leq 2W_{k-1}^{\langle v\rangle} + B_k^{\langle v\rangle},$$

which equals $W_k^{\langle v\rangle}$. Therefore, $W_k^{\langle u\rangle}$ rounds after t^*, u will increase its rank to $k+1$, while the rank of v can only increase it to $k+1$ after at least $1 + W_k^{\langle v\rangle} > W_k^{\langle u\rangle}$ many rounds from t^*.

By induction, as long as u is a leader, u will have a rank greater than the rank of v. If at some point u loses its leadership status, it must be that another node u' has a rank greater than the rank of u, and thus greater than the rank of v. □

Theorem 11 (GraphElect)**.** *The algorithm* GraphElect *elects a leader and computes a 2-coloring of bipartite graphs of diameter D in $O(D + \log n)$ rounds in expectation.*

Proof. We will compute the expected time to have only one leader left in the network. Once this event happens, the coloring propagated by the leader will be adapted by each node eventually producing a 2-coloring. Let $t(k)$ be the minimum round such that there exists a node that is updating its timer for the k-th time. In other words $t(k)$ is the round during which one or more nodes achieve the rank k for the first time and k is the largest rank at round $t(k)$. Also let L_k be the set of nodes with the largest rank k at round $t(k)$. By Lemma 10, we have that $L_{k+1} \subseteq L_k$ for any k. We want to compute the expected k^* such that $|L_{k^*}| = 1$. At the beginning L_0 contains all the nodes in the graph, so $|L_0| = n$. At round $t(k)$ we have L_k nodes which will select i.i.d. random numbers in $\{0, 1\}$: observe that only the nodes that select 0 will be in L_{k+1}. Therefore, in expectation, $|L_{k+1}| = \frac{1}{2}|L_k|$. We conclude that in $O(\log n)$ expected rounds there will be only one node with the largest rank.

At this point we have only one node of largest rank, call it u^*. However we can still have multiple leaders: for instance, u^* might be very far from some other leader w, and by the time the leading rank is propagated from u^* to w, w might have increased its own rank. However, this cannot happen when the timer length of u^* (hence, of all the other leaders) is at least D. Since after $O(\log n)$ rounds the u^*'s timer value will be more than 0 with high probability and the timer value doubles at each update, we have that after at most $O(\log n + D)$ rounds from round $t(k^*)$ the vale of u^*'s timer will be at least D. Thus, after $O(\log n + D)$ rounds there will be only one leader. □

2.6 Discussion: Leader election vs 2-coloring

Some of our algorithms are based on a leader election strategy. Observe that once a leader is elected, a 2-coloring of an entire bipartite network can be performed easily in D rounds, where D is the diameter of the network – the leader chooses a color, its neighbors use the other color, and so on. This argument unfortunately breaks down if the leader cannot be sure of being the only leader in the network[6] – the issue is that multiple leaders might initiate non-compatible colorings. Even using randomization, we cannot detect the existence of a unique leader in a model where nodes are not aware of the size (or diameter) of the network [AW04]. Nonetheless our leader election algorithms can be extended to compute a 2-coloring of the network in our model.

This discussion suggests the following questions. Is electing a leader always

[6] This problem is known as “termination detection” in distributed computing.

the (asymptotically) optimal choice to compute a 2-coloring of a bipartite graph? Also, is electing a leader as hard as 2-coloring a network? When considering specific classes of graphs the answer is negative, because network topology can play a major role. As a simple example, consider connecting n pendant nodes to the nodes of a $2n$-ring in an alternating fashion. Then an algorithm where a node colors itself red if it has degree 3 and blue otherwise would compute a 2-coloring in just one step, while a leader election instead requires $\Omega(n)$ rounds. Similarly, consider connecting exactly one pendant node to an even ring. Then an algorithm where a node declares to be a leader iff it has degree 3 would elect a leader in one step (while a 2-coloring requires $\Omega(n)$ rounds.)

2.7 Preferential attachment graphs

Preferential attachment graphs (henceforth, PA graphs) were introduced by Barabási and Albert [BA99] to model complex real-world networks. PA graphs capture the property that the degree sequence of many real-world networks has a “scale-free” power law distribution[7]. PA graphs with parameter $m \geq 1$ are generated by adding one node at a time. When a new node is added, it is connected to m neighbors chosen randomly with probability proportional to their degree. This well-known generative model creates the popular “rich-get-richer” effect, as high-degree nodes are more likely to be chosen as neighbors of new nodes.

The preferential attachment model was later formalized by Bollobás et al. [BRST01]. Fix any integer $m \geq 1$. Then, G_m^n is generated starting with the graph G_m^1 consisting of one node and m self-loops, and applying the following process $n-1$ times: G_m^{i+1} is obtained by G_m^i by adding a new node u and, sequentially, m edges $e_j = (u; v_j)$, $j = 1, 2, \ldots, m$, where $v_j \neq u$ is chosen with probability $\frac{\deg(v_j)}{2(mi+j)-1}$, and $v_i = u$ is chosen with probability $\frac{\deg(u)+1}{2(mi+j)-1}$. Here $\deg(u)$ and $\deg(v_j)$ denote the degrees of u and v_j before e_j is added (self-loops count twice towards the degree). Note that the definition of G_m^n allows for multiple edges and self-loops. When talking about coloring, we will implicitly refer to the graph obtained

[7] Specifically, the fraction of nodes with degree d is proportional to $d^{-\gamma}$, where γ is a constant independent of the size of the network.

by removing self-loops from the generated graph.

In this section we present an algorithm, PAcolor, that computes an $(m+1)$-coloring of any PA graph with parameter m. Each node u in the network uses $O(\log \delta_u)$ bits of memory, where δ_u denotes the number of distinct neighbors $v \neq u$ of u (note that $\delta_u \leq d(u)$). The number of colors used by PAcolor is in general optimal as, for any fixed $m \geq 1$, the first $m+1$ nodes of the PA graph will form a clique with constant probability. We will show that PAcolor computes an $(m+1)$-coloring of the graph in $O(\sqrt{n}\log^2 n)$ many rounds with high probability.

We now proceed with the description of the algorithm. Every node is initially uncolored and active. At every round, every active node u whose number of active neighbors is at most m, proposes to become inactive with some constant probability $0 < p < 1$. If u is the only proposing node in its neighborhood then it stores the set A_u of currently active neighbors in memory and becomes inactive (as $|A_u| \leq m$, the memory needed to store such set is $O(m \log \delta_u)$). Otherwise, the node does nothing in the current round. Every inactive node u waits until all of its neighbors in A_u are colored, and then colors itself with one of the available colors. We will see that the neighbors of u that are not in A_u will not be colored before u. Hence, as $|A_u| \leq m$, there is always at least one color available for u.

2.7.1 Analysis of the algorithm

We will now establish the correctness of the algorithm. First we claim that all nodes will eventually become inactive. By contradiction suppose the algorithm terminates with some active nodes. Then consider the active node u that was added last in the sequential process that generated the graph. Since all other active nodes were already added, u has at most m active neighbors at the time of termination. This leads to a contradiction. To argue that no node u has to wait indefinitely on its stored set A_u of neighbors before it gets a chance to color itself, we will consider the order dependencies between nodes and argue that they form a directed acyclic graph. Specifically consider the directed graph that has an edge from u to v iff $v \in A_u$. Suppose by contradiction that there is a cycle in this graph. Let u be the node of the cycle that became inactive first, and let v be the predecessor node of u in the cycle. By construction of the algorithms no two adjacent nodes become inactive in the same round. Therefore, it must be the case the $v \in A_u$ and $u \notin A_v$. However, since v is a predecessor of u in the dependency graph we have $u \in A_v$, which leads to a contradiction. Observe that the coloring process will start from the sinks of this directed acyclic graph, and will propagate up to the sources.

Observe that the number of rounds required for the algorithm to converge is at most twice the time it takes for all the nodes to become inactive. Therefore, since at each round at least a node will become inactive with constant probability (for any fixed m), it follows that the expected convergence time is $O(n)$. However we show the following stronger bound.

Theorem 12 (PAcolor). *Fix any integer $m \geq 1$ and let G_m^n be an n-node PA graph generated with parameter m. With high probability with respect to the randomness used for generating G_m^n and the randomness of the algorithm,* PAcolor *computes an $(m+1)$-coloring of G_m^n in $O(\sqrt{n}\log^2 n)$ rounds.*

Proof. Consider the nodes of G_m^n in the order determined by the PA process that generated G_m^n. We wish to analyze the number of steps required by the following process to terminate: in each step, remove the maximum number of consecutive nodes starting from the last that form an independent set. We claim that this process terminates in $O(\sqrt{n}\log n)$ with high probability.

Before proceeding with the proof of this claim, we will show how we can use it to bound the number of rounds needed by PAcolor to color G_m^n. Consider the maximum number of consecutive nodes starting from the last that form an independent set — call S this set. All the nodes in S have degree at most m and therefore are candidates for becoming inactive. We want to bound the number T of rounds required by PAcolor to make all the nodes in S inactive. Consider the subgraph of G_m^n induced by the nodes in S and their neighbors with degree at most m (the other neighbors are not candidate for becoming inactive). This subgraph has bounded degree, as each node has degree at most m. Also each node has a constant probability of becoming inactive, and the probability that $u \in S$ becomes inactive is independent from the probability that $v \in S$ becomes inactive if the distance between u and v is more than 2. Given the bounded degree, there are at least a constant fraction of nodes in S at distance more than 2 each other. Thus a constant fraction of nodes in S will become inactive in every round with probability at least $1 - \exp(\Omega(|S|))$ by a Chernoff bound argument. Therefore, we can conclude that $T \leq t$ for some $t = O(\log n)$ with probability at least $1 - O(n^{-c})$, for any $c > 0$. Using the claim, we can conclude that PAcolor converges in at most $O(t\sqrt{n}\log n) = O(\sqrt{n}\log^2 n)$ rounds with high probability.

We now proceed with proving the claim. Consider the following variation of the process. For $j = 1, \ldots, \log\sqrt{n}$, we define phase j as follows: as long as there are at least $\frac{n}{2^j}$ nodes left, in each step remove the largest independent S of consecutive nodes starting from the last. When the last phase is completed remove the last $\sqrt{n}$ nodes one by one.

First observe that the number of steps required for this new process to terminate is an upper bound for the number of steps required by the original process. Now consider any phase j. Let $s > \frac{n}{2^j}$. The probability that the last $t < s - \frac{n}{2^j}$ nodes of G_m^s do *not* form an independent set is at most

$$\sum_{i=1}^{t-1} \frac{2mi}{2m(s-t+i)+1} < \sum_{i=1}^{t-1} \frac{i}{s-t+i} \leq \sum_{i=1}^{t-1} \frac{i}{s-t} = \frac{t(t-1)}{2(s-t)} \leq 2^{j-1}\frac{t^2}{n},$$

where for the last step we used the fact that $s - t > \frac{n}{2^j}$. Therefore, at least $\sqrt{n/2^j}$ nodes will form an independent set with probability at least $p = 1/2$. Let E_t be the event that the t-th step of the j-th phase is such that $|S| > \sqrt{n/2^j}$. We know that for every step t in the j-th phase, $\Pr[E_t] \geq 1/2$ (no matter what happened in the previous phases and steps). Therefore the probability that the j-th phase lasts for more than $8\sqrt{n}$ steps is at most the probability of less than $(n/2^{j+1})/\sqrt{n/2^j} = 2\sqrt{n}$ heads in $8\sqrt{n}$ independent tosses of a fair coin, which can be bounded by $\exp(-\sqrt{n})$ using Chernoff bound. Therefore we can conclude that the process will be completed within $(1+\log\sqrt{n})\sqrt{n}$ with probability at least $1 - (\log n)\exp(-\sqrt{n})$, which proves the claim. □

2.8 Conclusions

Motivated by the coloring experiments in [KSM06, MPW09, EMPW11], we have proposed "natural" algorithms to compute a 2-coloring of even rings and general bipartite graphs. The notion of natural algorithms in our context is closely coupled with the concept of memory, in the sense that a low-memory constraint can lead to algorithms that are good candidates for modeling human strategies. As the existing literature on distributed computing does not focus on low-memory algorithms (except for [MS10, ESW12]), and simple algorithms such as RingGuess and GraphCoalescing have quadratic (or worse) running time, we faced the challenge of developing time-optimal algorithms while using low memory. For ring networks, we achieved the best of both world showing that RingElect computes a 2-coloring in linear time while using constant memory. The ideas in such algorithms also yield a new optimal algorithm for the leader election problem, improving over previous leader election algorithms. One drawback of RingElect is that, unlike RingGuess, is not self-stabilizing. It would be interesting to understand the limits of self-stabilizing low-memory algorithms.

For general bipartite graphs, we proposed GraphElect that computes a 2-coloring in $O(D + \log n)$ many rounds using only $O(\log n)$ bits of memory. The question whether there exists an algorithm that can 2-color a general bipartite graph in $O(D + \text{poly}(\log n))$ many rounds using constant memory is still open. We conjecture that such algorithm does not exist: specifically, we conjecture that any algorithm that uses only constant memory requires at least $\Omega(D^{1+\epsilon})$ many rounds to compute a 2-colorings of bounded-degree bipartite graphs with diameter polynomial in n.

Finally, we have focused on coloring of preferential attachment graphs as such topologies were also used in the coloring experiment in [KSM06]. We have proposed a simple algorithm, PAcolor that colors a preferential attachment graph in $\tilde{O}(\sqrt{n})$ many rounds. We believe that the analysis of such algorithm can be improved — simulations suggest that PAcolor terminates in $O(\text{poly}(\log n))$ many rounds but a formal proof of this fact is missing. Preferential attachment graphs have been shown to be quick networks in terms of spreading of information (see for example [CLP09, Gia11, DFF12] and references therein). We believe their high conductance and degree distribution properties will suffice even in solving efficiently the coloring problem. However, an efficient algorithm for such problem will require some amount of coordination among the nodes of the network. For instance, the natural dynamic suggested and used as a comparative tool in [KSM06] will most likely require quadratic (or worse) many rounds to compute a coloring of a preferential attachment graph. Formal proofs of such conjectures would be very interesting.

Chapter 2, in part, is a reprint of the paper "Low memory distributed protocols for 2-Coloring" co-authored with Amos Israeli, Mathew D. McCubbins and Ramamohan Paturi, published in the proceedings of the 12th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS 2010 [IMPV10]. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Matching Experiments

This chapter argues that despite the inherent complexity of human social interactions, it is possible to isolate basic behavioral principles, formulate mathematical models, and predict collective dynamics, using an algorithmic approach. As a simple example of this approach, in the context of a distributed coordination game on networks (i.e., the *maximum matching game*), we present an algorithmic model of human behavior that is based on *simple* principles of local interaction and that is able to capture *complex* collective coordination.

Our approach is similar in spirit to the one in physics where particle systems and cellular automata described by simple rules are known to generate complex behaviors, such as phase transitions and universal computability [Lig85, Bov06, VNB66, Coo04]. However, our algorithmic modeling approach embeds individual interaction behavior as part of a distributed computing system and leads to computational complexity analysis.

Our work is influenced by the work of Kearns et al. [KSM06] who studied the effect of network topology on subjects' ability to color a graph, and by subsequent work in the context of distributed coloring and consensus games [KJTW09, MPW09, JKV10, EMPW11]. However, our focus is on algorithmic modeling and analysis, rather than on observing the effect of network topology on performance.

We have conducted over 250 experiments with human subjects on a pool of over 80 networks with up to 24 nodes each, ranging from simple networks to more complex stochastic models including preferential attachment [BA99, BRST01] and

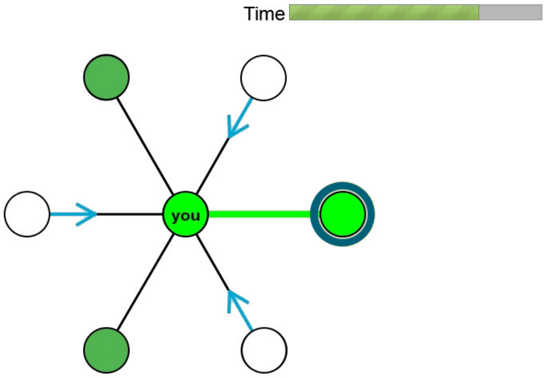

Figure 3.1: **Computer interface.** The subject is matched with the node on the right and is being requested by three unmatched nodes.

small-world networks [WS98]. Our experimental set-up is simple. Subjects are represented by nodes of a network with edges representing potential matches. In our experiments, human subjects are connected over a virtual network and interact with their neighbors through a computer interface, see Figure 3.1. Subjects can form and destroy pairs with their neighbors, and each subject can be part of a single pair at a time. Subjects are given only local information about their immediate neighbors, and can only interact with them. They are able to propose to match with a neighbor and accept a proposal from a neighbor. While matched, a subject can also make a proposal to or accept a proposal from another neighbor; in both cases, the existing match would automatically be broken. Moreover, a subject can only have a single outstanding proposal at a time. Therefore, at any time, a subject can either be part of a matched pair, or not be matched and have at most a single outstanding proposal. Subjects are equally incentivized to achieve a *maximum matching*, namely to form the maximum number of disjoint mutual pairs, without regard to whom is matched with whom. Specifically, they are given an equal monetary reward for each game where a maximum matching is found within the allotted time.

To better understand this setup, consider the following metaphor: imagine that incoming graduate students are pairing up with faculty members. Further imagine that every member of the department prefers every graduate student to

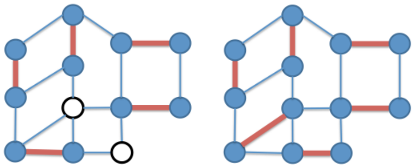

Figure 3.2: **Approximate and maximum matching.** Left: an approximate maximum matching of size 5 on a network with 12 nodes (matching edges are represented in bold red, matched nodes are colored, unmatched nodes are white). Right: a maximum matching of size 6 on the same network (note that the maximum matching is also a perfect matching, as all nodes are matched).

have one adviser and every adviser to have one graduate student, and only certain faculty and graduate students share interests. Communication is limited so that individuals can only tell if members with whom they share an interest are already matched. Each member of the department is now a node, the edges represent shared interest, and individuals can then propose to work with members with whom they share an edge.

Our algorithmic model is based on a simple property that we call "prudence" and that emerges from the analysis of a first set of experimental data. This property states that *individuals do not break existing matched pairs unless they receive an alternative proposal by an unmatched neighbor.* Based on this property, we propose a simple distributed algorithm, analyze its performance, validate the model with additional experimental results, and predict outcomes. The prudence property is reminiscent of the notion of risk aversion, a relevant topic in the economics literature [Pra64, KT79].

3.1 Results

We now briefly summarize our findings. Throughout the paper we use the graph-theoretic terminology, according to which a matching is a set of edges without common nodes. The size of a matching is the number of edges in it. A maximum matching is a matching with the largest size. For $0 < c \leq 1$, a matching is a c-approximate maximum matching if its size is within a factor of c from that

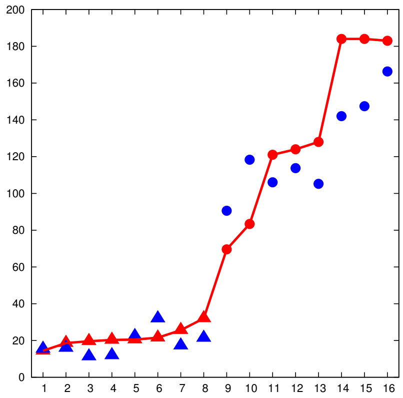

Figure 3.3: **Affinity between humans' and algorithm's performance, 16-node networks.** The performance of the human subjects (red points joined by continuous line) and of the algorithm (blue points) over eight bipartite 16-node networks (triangles) and eight non-bipartite 16-node networks (circles) are plotted. The experiment was run multiple times on each network and the average behavior is reported. The x-axis shows the indexes of the networks sorted by increasing average time required to reach a maximum matching. Bipartite networks are labeled from 1 to 8, while non-bipartite networks are labeled from 9 to 16. The y-axis shows the average time (in seconds) required to reach a maximum matching for humans, while the average number of rounds of the algorithm is scaled by a constant factor.

of a maximum matching. A matching M is maximal if it is not a proper subset of any other matching, i.e., for any new edge added to it, it is no longer a matching. Figure 3.2 depicts an approximate and a maximum matching of a network. We show that the convergence time to the maximum matching in computer simulations of the prudence algorithm fits well the experimental data (after scaling by a constant factor), see Figures 3.3 and 3.4. By computer simulations we also predict that convergence to a maximum matching is slower on preferential attachment networks than on small-world networks, see Figure 3.5. This prediction is validated by our experiments with human subjects. It is also in agreement with the experimental results by Kearns et al. [KSM06] regarding the coloring problem, and with the theoretical results by Montanari and Saberi [MS09] regarding the spread of innovation in networks. On the theoretical side, we analyze the dynamics of the prudence algorithm and show that for all graphs of bounded degree a 1/2-*approximate maximum* matching is reached quickly, on average in $O(\log n)$

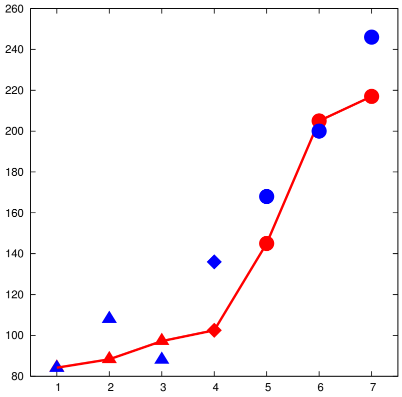

Figure 3.4: **Affinity between humans' and algorithm's performance, 24-node networks.** The performance of the human subjects (red points joined by continuous line) and of the algorithm (blue points) over different 24-node networks are plotted. In particular, small-world networks (triangles), a ring network (diamonds), and preferential attachment networks (circles) were tested. The experiment was run multiple times on each network and the average behavior is reported. The x-axis shows the indexes of the networks sorted by increasing average time required to reach a maximum matching. The y-axis shows the average time (in seconds) required to reach a maximum matching for humans, while the average number of rounds of the algorithm is scaled by a constant factor.

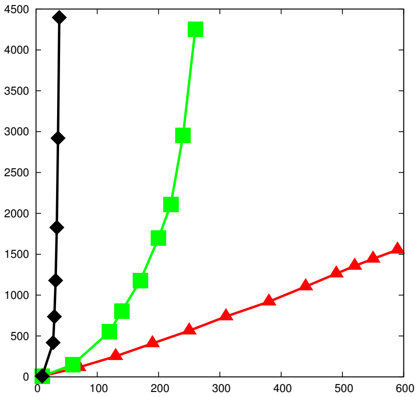

Figure 3.5: **Algorithm's asymptotic performance.** PRUDENCE algorithm's performance with respect to the network's size for the "bad" graph G_n (black diamonds), for preferential attachment model (green squares), small-world model (red triangles). For each generative model and network size we generated 100 networks and run the algorithm 1000 times on each. The average behavior is reported. The x-axis shows the network size, and the y-axis shows the average number of rounds required by the algorithm to converge to a maximum matching.

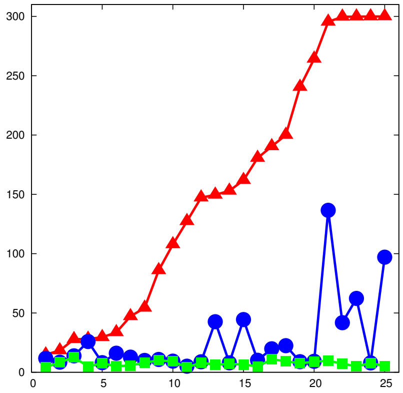

Figure 3.6: **Experimental performance, 24-node networks.** Performance of the experimental subjects on networks of 24 nodes. The plot shows the time to reach a perfect matching of size 12 (red), an approximate matching of size 11 (a 0.92–approximate matching, in blue) and a matching of size 6 (a 1/2–approximate matching, in green). Results for single games are reported. The x-axis shows the indexes of the games sorted by increasing solving time, while the y-axis shows the time in seconds. The right-most four games on the red plot did not converge to a maximum matching and correspond to three instances of the "bad" graph G_n and to one instance of the preferential attachment network.

rounds, where n refers to the number of nodes in the network (Theorem 16); and for all graphs a $(1-\epsilon)$-*approximate maximum* matching is reached in polynomially many rounds with high probability (Theorem 17). We also show that there are instances (called "bad" graphs) for which reaching a *maximum* matching requires exponential time with high probability when starting from a set of configurations (called "bad" matchings) which constitute almost all possible configurations (Theorems 20 and 21). These results show that in the worst case there is an exponential gap between reaching a good matching (i.e., an approximate maximum matching whose cardinality is close to a maximum matching) versus the best (i.e., perfect) matching. The experimental data shows (consistently with the theoretical analysis) that human subjects always find a "good" matching quickly, while they can take much longer to improve the solution to a maximum matching, see Figure 3.6. In particular, on the bad graph, human subjects could not converge to a maximum matching in the allotted time.

We point out that our simplified setup constitutes a simplification of the richness and heterogeneity of the ties in real social networks, as the subjects have no preference over each other, all the ties are equivalent, and interaction has no cost. However, such a simplified model leads to a tractable analysis and to the formulation of a general principle of collective behavior.

3.2 Related work

We refer the reader to Section 2.2 for work related to the experimental study of human strategic behavior over networks.

Matching over networks has been extensively studied from both theoretical and algorithmic points of view, since the work on marriage markets by Gale and Shapley in the early sixties [GS62]. An exhaustive game-theoretic formulation of one-to-one and many-to-one matching can be found in the book by Roth and Sotomayor [RS92]. Recently, Hatfield and Kominers [HK10] proposed a centralized algorithm for the general case of bilateral contracts in many-to-many markets.

Several studies have been devoted to the design of parallel and distributed algorithms for matching. For the parallel setting, it is known that a perfect matching can be found in polylogarithmic time on a PRAM [KUW86, MVV87]. As for the distributed setting, Lotker et al. [LPSP08] proposed an algorithm that finds a $(1 - \epsilon)$-approximate matching in logarithmic time – improving upon the classical result about distributed *maximal* matching by Israeli and Itai [II86] – but is not suitable to model human strategies.

Another problem closely related to matchings is the assignment problem. In the fifties, Kuhn [Kuh55] presented a centralized algorithm that solves it in polynomial time, known as the Hungarian algorithm. Bertsekas et al. [BC91] proposed parallel algorithms for network flow problems, of which the assignment problem is a special case. Zavlanos et al. [ZSP08] extended the work in [BC91] by superimposing a communication network over the assignment graph.

Our approach of modeling collective human behavior is related to Markov chains over matchings. Sasaki and Hajek [SH88] analyzed a Metropolis-like algorithm and proved that it is not guaranteed to find a maximum matching of a general graph in polynomial time.

3.3 The matching games

The experiments included the interaction of the participants through a computer interface, and were conducted in accordance with the ethical standards specified in the 1964 declaration of Helsinki. Written consent was granted before participation in the experiments. Our institutional review boards approved this study (UCSD IRB approval 111213SX, US Army Human Research Protection Office ARO-HRPO Log A-17038).

Before formulating our algorithmic model, we conducted four sessions of experiments, each with a different pool of sixteen undergraduate students connected over a virtual network. Subsequently, to validate our model, we ran an additional session of experiments with a pool of twenty four subjects on a set of networks that included small world and preferential attachment networks. In each of the first four sessions the subjects were asked to solve the matching game on a pool of over 70 networks. All networks admitted a *perfect matching*, namely a maximum matching with no unmatched nodes. We considered networks classified into four groups: bipartite networks admitting a unique perfect matching, bipartite networks admitting multiple perfect matchings, non-bipartite networks admitting a unique perfect matching, non-bipartite networks admitting multiple perfect matchings. Within each group, networks were randomly generated. As a remark, a bipartite network is a network whose nodes can be divided into two disjoint sets V_1 and V_2 such that every edge connects a vertex in V_1 to one in V_2. If this property does not hold, we say that the network is non-bipartite. Subjects sat in front of workstations for the entire two-hour duration of the session and had no eye-contact with each other. For each matching game, a network was chosen, subjects were randomly assigned to its nodes, and each subject interacted with its neighbors by making or accepting proposals to form matched pairs using the interface shown in Figure 3.1. Each subject could control the node in the center of the screen and could only see its neighbors and, among those, distinguish which of them were currently matched (marked in dark green). A subject could make proposals or accept proposals by selecting a neighbor with a mouse click, and could only have one outstanding proposal at a time to form a matched pair (circled in yellow). While subjects knew whether a neighbor is matched or unmatched, they did not have direct knowledge of any outstanding requests to their neighbors other than their own. If two neighbors selected each other, a pair was formed (a bright green link appeared between them) which could be broken when one of the partners selected another neighbor. As a remark, since a pair was formed when two subjects selected each other and each subject could make a single selection at a time, each subject could be part of a single pair at a time (with one of its neighbors).

If a perfect matching was found within the time limit of five minutes, the game was declared solved and each participant was rewarded by \$.50 or \$1 depending on the session, otherwise the game ended with no reward. The number of games in an experimental session was not fixed, but games were run for the two-hour duration of the session. Therefore, the number of games and the cumulative reward in a session depended on the performance of the participants, providing an additional incentive to coordinate.

The networks used in this first set of experiments can be divided into four classes: bipartite, non bipartite, unique perfect matching, multiple perfect matchings. Two one-tailed Welch's t-tests confirmed the hypotheses that it is harder for humans to complete the matching game on non-bipartite than on bipartite networks (p-value < 0.001); and that non-bipartite networks with unique perfect matching are more difficult to solve than non-bipartite networks with multiple perfect matchings (p-value < 0.001). No statistically significant difference was found between the completion time of bipartite networks with unique and with multiple perfect matchings.

3.4 The algorithmic model

One of the main behavioral properties that emerged from the experimental data is that matched players *may* break their current matching *only if* they have

Algorithm 3.1: PRUDENCE algorithm for node u

```
if unmatched then
    $f(u) \leftarrow \textsc{UnmatchedChoose}(u)$
else if matched and $\exists$ neighbor $v$ such that $f(v) = u$ then
    $f(u) \leftarrow \textsc{MatchedChoose}(u)$
```

been requested by an unmatched neighbor. In particular, in 30% of the games no player ever violated this rule at any time during the game. In the remaining games, over 93% percent of the moves were in agreement with this rule. Therefore, this property led to the following modeling assumption:

Assumption 13 (Prudence)**.** *A matched node does not break its current matched pair if it does not receive any request from other neighbors.*

Two remarks are in order. First, note that this behavioral rule is peculiar to the matching problem since each player needs to choose a partner but also needs to be chosen. Second, notice that a matched subject with unmatched neighbors has some incentive to behave non-prudently and break the current match, because the subject can infer from the status of its neighbors that the perfect matching is not reached yet. However, experimental data shows that this rarely happens.

For each node u, let $f(u)$ indicate u's current preference. In other words, $f(u)$ is the unique node to which u has currently proposed to. $f(u)$ will be null if u does not have a current proposal. If two neighbors u and v currently prefer each other (i.e., $u = f(v)$ and $v = f(u)$), then consider them matched and the edge $e = \{u, v\}$ as part of the matching. Assume that each node knows if a neighbor is matched or unmatched.

Given the prudence property, we model the algorithmic behavior of humans using the PRUDENCE algorithm (see Algorithm 3.1). The algorithm is specified by the implementation of two functions, called MATCHEDCHOOSE(u) and UNMATCHEDCHOOSE(u), which are placeholders for the behavior that node u would follow depending on whether u is matched or unmatched. We consider a synchronous setting, in which time is divided into rounds, and at the beginning of each round each node observes its status and the status of its neighborhood and then decides on an action to take.

In the following we provide a canonical implementation of the functions UNMATCHEDCHOOSE(u) and MATCHEDCHOOSE(u) consistent with the prudence property. UNMATCHEDCHOOSE(u) does not change the current value of $f(u)$ with probability p, while with probability $1 - p$ accepts the proposal from a neighbor uniformly at random from among the neighbors v with $f(v) = u$ if any; if there is no neighbor v with $f(v) = u$, then it proposes to a node uniformly at random from among the unmatched neighbors if any; otherwise it proposes to a node uniformly at random from among all the matched neighbors. In other words, unmatched nodes prefer neighbors who requested them over other unmatched neighbors, and unmatched neighbors over matched neighbors. As for matched nodes, MATCHEDCHOOSE(u) accepts a proposal from a neighbor uniformly at random from among the neighbors v with $f(v) = u$ (note that u's current partner is one of them). We remark that the simulations' performance and the fit with the experimental data was practically insensitive to the value of p chosen in the run of the algorithm.

3.5 Analysis

In this section we present our analytical results regarding the convergence behavior of the PRUDENCE algorithm. In particular, our results describe how well the algorithm performs in finding a large matching and the time it takes in terms of the number of rounds required. Due to space constraints, we only present proof sketches here. Complete details of the proofs are deferred to the SI.

We define a *matching* at round t as the set of matched edges at the beginning of round t of the algorithm. We first claim that the prudence property implies that the size of the matching does not decrease with time. The proof is immediate and it is omitted.

Claim 14. *The size of the matching at round t is non-decreasing as t increases.*

We then observe that the behavior of the PRUDENCE algorithm can be described by a Markov chain over matchings. A transition from a matching M

to a matching M' is made by selecting an edge $e = \{u, v\}$ such that at least one among u and v is unmatched, and setting $M' = M + e$ if u, v are both unmatched, and $M' = M + e - e'$ if exactly one of u and v is matched in M and e' is the matching edge. This Markov chain is reversible when restricted to matchings of the same size. Since the Markov chain is memory-less and has positive probability of reaching a maximum matching, we conclude that the Prudence algorithm enjoys self-stabilization.

Claim 15. *The* Prudence *algorithm is a self-stabilizing algorithm.*

Our first theorem says that a 1/2-approximate matching will be reached quickly in networks with bounded degree.

Theorem 16. *In any bounded-degree graph on n nodes, the expected number of rounds for the* Prudence *algorithm to reach a 1/2-approximate matching is $O(\log n)$.*

The key idea of the proof is to show that, in expectation, the "distance" in terms of number of matched pairs to the smallest *maximal* matching shrinks by a constant factor in each round of the Prudence algorithm. Since it is well known that any maximal matching is a 1/2-approximation of the maximum matching, the result then follows.

Proof of Theorem 16. For ease of presentation, we assume $p = 0$, and remark that this result holds for all choices of $0 \leq p < 1$. Let G be a graph of n nodes and maximum degree Δ. Let m be the number of matched nodes in the smallest maximal matching of G. For $t \geq 0$, denote by W_t the set of nodes of G which are unmatched and have at least an unmatched neighbor at the beginning of round t, and let $|W_t|$ be the cardinality of W_t. Also, let M_t be the matching of G obtained by the Prudence algorithm at the beginning of round t and N_t be the number of nodes matched by M_t. For $t \geq 0$, define the random variable

$$D_t = m - N_t.$$

We devote the rest of the proof to showing that

$$E[D_t] \leq (1 - (\Delta + 1)^{-3})^t E[D_0] \tag{3.1}$$

The theorem then follows by the observations that $E[D_0] \leq n$ and that any maximal matching is at least a 1/2-approximation of the maximum matching.

To prove (3.1), we will first show that $E[B_t(W_t)|W_t] \geq (\Delta+1)^{-3}|W_t|$, where $B_t(W_t)$ is the number of nodes in W_t that match with nodes in W_t during round t (here the expectation is taken over the randomness of the algorithm during round t). For $u \in W_t$, let $Z_t(u)$ be the indicator random variable that takes value 1 if and only if u gets matched with a node in W_t during round t. By linearity of expectation, we have that

$$E[B_t(W_t)|W_t] = \sum_{u \in W_t} E[Z_t(u)] = \sum_{u \in W_t} \Pr(Z_t(u) = 1).$$

Let A_t be the set of nodes $u \in W_t$ such that (i) u has no incoming or outgoing request to nodes in W_t, and (ii) all neighbors $v \in W_t$ of u have an incoming request. Let $\bar{A}_t = W_t \setminus A_t$. For $u \in A_t$, we have that $\Pr(Z_t(u) = 1) = 0$, as unmatched nodes prefer neighbors who requested them over other unmatched neighbors. On the other hand, for $u \in \bar{A}_t$, we have $\Pr(Z_t(u) = 1) \geq \Delta^{-2}$. To see this, note that a pending request involving u (if any) will be honored with probability at least Δ^{-2}; if no such request exists, the co-occurrence of the event of u requesting a neighbor with no incoming request and of that neighbor requesting u happens with probability at least Δ^{-2}. By definition of A_t, no two nodes in A_t can be neighbors. Also, by definition of W_t, every node $u \in W_t$ has at least one neighbor in W_t. These two facts imply that $|\bar{A}_t| \geq (\Delta + 1)^{-1} W_t$. We can conclude that $E[B_t(W_t)|W_t] \geq (\Delta + 1)^{-3}|W_t|$.

We now relate D_{t+1} to $B_t(W_t)$. First, note that $D_{t+1} \leq D_t - B_t(W_t)$. By itself, this bound is not strong as W_t can be small. However, when W_t is small, the current matching must be close to a maximal matching. Indeed, by considering the union of M_t and any maximal matching of W_t, we have that $m \leq N_t + |W_t|$. This implies that $D_t = m - N_t \leq |W_t|$ and hence $D_{t+1} \leq D_t - B_t(W_t) \leq |W_t| - B_t(W_t)$. Therefore, we have

$$\begin{aligned} D_{t+1} &\leq D_t - B_t(W_t), \\ D_{t+1} &\leq |W_t| - B_t(W_t). \end{aligned}$$

By taking the expectations with respect to the randomness of the algorithm during round t, we get

$$\begin{aligned} E[D_{t+1}|W_t, D_t] &\leq D_t - E[B_t(W_t)|W_t] \leq D_t - (\Delta+1)^{-3}|W_t|, \\ E[D_{t+1}|W_t, D_t] &\leq |W_t| - E[B_t(W_t)|W_t] \leq |W_t| - (\Delta+1)^{-3}|W_t| \\ &= (1-(\Delta+1)^{-3})|W_t|. \end{aligned}$$

Now, by taking the expectation with respect to the randomness of the algorithm during rounds up to t, we obtain

$$\begin{aligned} E[D_{t+1}] &\leq E[D_t] - (\Delta+1)^{-3}E[|W_t|], \\ E[D_{t+1}] &\leq (1-(\Delta+1)^{-3})E[|W_t|]. \end{aligned}$$

Letting $d_t = E[D_t]$, $w_t = E[|W_t|]$, and $\alpha = (\Delta+1)^{-3}$, the bounds above can be rewritten as

$$d_{t+1} \leq \min\left\{d_t - \alpha w_t, (1-\alpha)w_t\right\}.$$

To conclude the proof of (3.1), we show by induction that $d_t \leq d_0(1-\alpha)^t$. For $t=0$, as $d_0 \leq w_0$, we have $d_1 \leq d_0 - \alpha w_0 \leq (1-\alpha)w_0$. Now, let us consider any $t \geq 1$ and distinguish between the cases of $w_t \leq d_0(1-\alpha)^t$ and $w_t > d_0(1-\alpha)^t$. If $w_t \leq d_0(1-\alpha)^t$, we have $d_{t+1} \leq (1-\alpha)w_t \leq d_0(1-\alpha)^{t+1}$. Otherwise, if $w_t > d_0(1-\alpha)^t$, using the induction hypothesis, we have that

$$d_{t+1} \leq d_t - \alpha w_t \leq d_0(1-\alpha)^t - \alpha w_t \leq d_0(1-\alpha)^t - d_0\alpha(1-\alpha)^t = d_0(1-\alpha)^{t+1},$$

which completes the proof. □

We remark that the assumption of having bounded degrees is necessary as there are unbounded degree graphs in which a polynomial number of rounds is required with high probability to achieve a 1/2-approximation. However, in this case, a polynomial number of rounds is also enough to achieve *any* constant approximation: indeed, as the next theorem states, the Prudence algorithm provides a PTAS (polynomial time approximation scheme) for the maximum matching problem. Given a graph G, Δ denotes its maximum degree.

Theorem 17. *For any graph G of n nodes, $\epsilon > 0$ and $c \geq 1/2$, the* Prudence *algorithm reaches a $(1-\epsilon)$-approximate matching in $\frac{c}{\epsilon} n\Delta^{2/\epsilon}$ rounds with probability at least $1 - \exp(-c\epsilon^2 n/2)$.*

The theorem implies that, for any constant $\epsilon > 0$, a matching whose size is within a $(1 - \epsilon)$ fraction of the size of the maximum matching is reached in polynomial time. For bounded-degree graphs, this result also holds for $\epsilon = \Omega(1/\log n)$, implying that in this case a maximum matching can be reached in polynomial time.

To prove the theorem, we track the progress of the algorithm towards an approximate maximum matching, using the concept of an *augmenting path.* An augmenting path is a path of odd length which alternates between matched and unmatched edges and whose extreme edges are unmatched. It turns out that there is a close connection between the size of a shortest augmenting path in a matching and how close the matching size is to the size of a maximum matching. More specifically, we use the following lemma due to Hopcroft and Karp [HK71].

Lemma 18. *Consider any matching M that does not admit augmenting paths of odd length k or smaller. Then, the size of M is at least a fraction $\frac{k+1}{k+3}$ of the size of a maximum matching.*

Hence, to prove Theorem 17, we need to show that short augmenting paths (for a suitably chosen k) are solved in a short amount of time. It is useful to consider a particle analogy to understand the process that eliminates short augmenting paths. We consider each unmatched node as a particle. Particles move around the graph from node to node as nodes change their status between matched and unmatched states dictated by the random choices in the algorithm. There are exactly two particles along an augmenting path, situated at the extreme nodes. To understand how an augmenting path gets shorter and eventually vanishes, we consider how the two particles move closer to each other along the path.

Let $u_0, u_1, u_2, \cdots, u_\ell$ denote a *shortest* augmenting path. If the extreme unmatched node u_0 proposes to u_1 and u_1 accepts the proposal breaking the current match with u_2, then the particle moves from u_0 to u_2. A similar argument applies to the other end of the path. Also, the minimality of the path guarantees that the internal nodes do not change their current matching as they have no unmatched neighbor. It follows that the particles become closer to each other and the augmenting path gets shorter. Using this approach, we can prove that with suitable probability the length of the *shortest* augmenting path shrinks after each round. When an augmenting path becomes an edge (that is, a path of length one), and if the extreme unmatched nodes select each other as partners, the particles and the path vanish, yielding an increment to the size of the matching. Hence, a key step of our proof is to lower bound the probability that an augmenting path of length k vanishes, and then to apply Lemma 18 to relate the existing augmenting paths and the matching size.

Proof of Theorem 17. For ease of presentation, we assume $p = 0$, and remark that this result holds for all choices of $0 \leq p < 1$. Let G be a graph of n nodes, maximum degree Δ, and maximum matching of size OPT. We will consider the unmatched nodes as *particles* randomly moving on the nodes of the network as per the algorithm choices. To see how the particle move, consider the particle positioned at any unmatched node u. If u requests a matched neighbor v and v accepts the requests, then the particle will move to v's old partner (which is left unmatched). If u requests an unmatched neighbor z and z accepts the request, then both the particles at u and z will dissolve. Note that when two particles dissolve the size of the matching increases by one.

An augmenting path is a path of odd length which alternates matched and unmatched edges and whose extreme edges are unmatched. Observe that by switching each unmatched edge of an augmenting path into a matched edge, and viceversa, the size of the matching increases by one.

We split the rounds into epochs of $\lfloor 1/\epsilon \rfloor$ rounds each. We claim that if at the beginning of any epoch the size of the matching is less than a $(1-\epsilon)$OPT, then the size of the matching increases by at least one by the end of that epoch with probability at least $\Delta^{-2/\epsilon}$. To prove the claim, consider the first round of any epoch and let $u_0, u_1, \ldots, u_\ell$ be any *shortest* augmenting path at the beginning of that round. It must be that $\ell < 2(\epsilon^{-1} - 1)$, otherwise Lemma 1 would imply that the size of the matching is at least a $\frac{\ell+1}{\ell+3} \geq 1-\epsilon$ fraction of OPT. For $\ell = 1$, u_0 and u_1 will match with each other during the first round with probability at least Δ^{-2}, hence the claim is true. For $\ell = 3$, u_0 and u_3 will request respectively u_1 and u_2 with probability at least Δ^{-2} during the first round of the epoch, and these requests will be accepted in the second round with probability at least Δ^{-2} — hence, the size of the matching increases by one within 2 rounds with probability at least Δ^{-4}. Now consider $5 \leq \ell < 2(\epsilon^{-1}-1)$. We have that two particles occupy the nodes u_0 and u_ℓ at the extremes of the augmenting path. With probability at least Δ^{-2}, u_0 requests to match with u_1 during the first round and u_1 accepts in the second round, making the corresponding particle move from u_0 to u_2. A similar argument yields that the particle at u_ℓ moves to $u_{\ell-2}$ within two rounds with probability at least Δ^{-2}. Moreover, as the augmenting path under consideration is a shortest augmenting path, nodes $u_2, \ldots, u_{\ell-2}$ have no unmatched neighbors at the beginning of the first round and hence do not receive any matching request during that round. Therefore, with probability at least Δ^{-4}, at the end of the second round the nodes u_2 and $u_{\ell-2}$ are unmatched whereas nodes $u_3, \ldots, u_{\ell-3}$ did not change their partner. That is, the length of the shortest path at the beginning of the third round of the epoch is at most $\ell - 4$ with probability at least Δ^{-4}. By means of the same argument, we can conclude that with probability at least $(\Delta^{-4})^{\ell/4} > \Delta^{-2/\epsilon}$, all nodes in an augmenting path are matched within $\ell/2 \leq \lfloor 1/\epsilon \rfloor$ rounds, which proves the claim.

For any epoch i, we now associate a binary random variable X_i which takes on value 1 with probability $p = \Delta^{-2/\epsilon}$. The claim guarantees that the size of the matching after T epochs is at least $\min\{(1-\epsilon)\text{OPT}, \sum_{i=1}^{T} X_i\}$. Also, as successive rounds of the algorithm are independent, the X_i's are independent random variables. For any $0 < \delta \leq 1$, the Chernoff bound states that

$$\Pr\left[\sum_{i=1}^{T} X_i < (1-\delta)Tp\right] < \exp(-Tp\delta^2/2).$$

For any $c \geq 1/2$, by setting $T := cn\Delta^{-2/\epsilon}$ and $\delta := \epsilon$, the above yields that the size of the matching after T epochs (i.e., after $T\lfloor 1/\epsilon \rfloor \leq \frac{c}{\epsilon} n\Delta^{2/\epsilon}$ rounds) is at least $\min\{(1-\epsilon)\text{OPT}, (1-\epsilon)cn\} = (1-\epsilon)\text{OPT}$ with probability at least

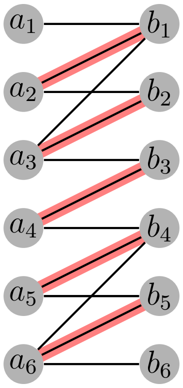

Figure 3.7: **The bad graph.** The “bad” graph G_n for $n = 3$. One of the “bad” matchings of Theorem 20 is highlighted in red.

$1 - \exp(-c\epsilon^2 n/2)$. □

We remark that the random process governing the movement of the particles in the network is not a classical random walk over the nodes of the graph. Indeed, if that were the case, a *maximum* matching would always be reached in polynomial time by a simple cat-and-mouse argument. Instead, a random move of a particle depends on the current matching, which in turn changes when the particle moves. This modest difference can lead to an exponential time gap between convergence to an approximate matching and convergence to a maximum matching. Indeed, exploiting the dependence of the particles' movements on the current matching, we show that there is a family of graphs for which the PRUDENCE algorithm takes exponentially many rounds with high probability to reach a maximum matching starting from a set of configurations that cover almost all possible cases. This family of “bad” graphs is defined as follows (see also Figure 3.7).

Definition 19 (Bad graph G_n)**.** *The bipartite graph $G_n = (A \cup B, E)$ has $4n$ nodes $A = \{a_1, \ldots, a_{2n}\}$ and $B = \{b_1, \ldots, b_{2n}\}$, and its edges are (a_{n+1}, b_n), (a_i, b_j) for all $1 \leq i \leq n$ and $1 \leq j \leq i$, and (a_i, b_j) for all $n+1 \leq i \leq 2n$ and $n+1 \leq j \leq i$.*

Note that the set of “horizontal” edges (a_i, b_i), for $1 \leq i \leq 2n$ is the unique perfect matching for G_n.

Theorem 20. *The* Prudence *algorithm requires* $2^{\Omega(n/\log^2 n)}$ *many rounds with high probability to reach the perfect matching when starting from any* $(2n-1)$*-matching in which the two unmatched nodes are in opposite sides of* G_n*.*

The main idea of the proof is to track the positions of the unmatched nodes throughout the course of the algorithm and to lower bound the number of rounds needed before they meet as an adjacent pair.

We first prove a one-to-one correspondence between the Markov process of the state evolution between matchings and a classical random walk on a tree (represented in Figure 3.8) whose size is exponential in n. We show that this classical random walk takes exponential time to reach the root of the tree starting at any one of its nodes, thus providing a lower bound on the convergence time of the Prudence algorithm.

We say that a matching M of G_n of size $2n-1$ is *bad* if the Prudence algorithm requires exponentially many rounds with high probability to converge to the perfect matching when starting from M. Observe that all matchings considered by Theorem 20 are bad. The following theorem states that *almost all* matchings of size $2n-1$ are bad.

Theorem 21. *The ratio between the number of "bad" matchings and the number of all* $(2n-1)$*-matchings of* G_n *is* $1-O(2^{-n})$*.*

Theorems 20 and 21 show that the Prudence algorithm requires exponentially many rounds to converge to the perfect matching of G_n when starting from a set of configurations (the bad matchings) constituting almost all possible cases (the matchings of size $2n-1$).

3.6 Prediction and validation of the model

Figure 3.3 compares the performance of the human subjects (red) with that of simulations (blue) on a set of 16-node networks (8 bipartite networks and 8 non-bipartite networks) with unique perfect matchings. The networks are sorted by increasing average completion time, and as a result bipartite networks are labeled from 1 to 8, while non-bipartite networks are labeled from 9 to 16. Each of these networks was tested at least 6 times over all sessions. The vertical axis represents the time (in seconds), and the numerical values of the convergence time of the algorithm are scaled by a constant factor to best match the experimental data.

In an additional experimental session, we tested twenty four subjects connected over small-world, preferential attachment and ring networks as well as over the "bad" graph G_n. The games on the bad graph were never solved, consistent with the prediction of exponentially slow convergence. Furthermore, we found that preferential attachment networks were more difficult to solve than small-world networks (one-tailed Welch's t-test, p-value < 0.01). Figure 3.4 shows the affinity between humans' (red) and algorithm's (blue) performance, on this set of 24-node networks: small-world networks (triangles), ring network (diamonds), preferential attachment networks (circles). The x-axis shows the indices of the networks sorted by increasing average time to find the perfect matching, and the y-axis shows the average time.

Figure 3.5 shows, by simulation, that the algorithm scales linearly in the size of the network in the case of small-world networks [WS98], while it scales polynomially for preferential attachment networks [BA99, BRST01], and exponentially on the "bad" graph G_n. These results closely resemble the experimental data of the coloring games performed by Kearns et al. [KSM06], where preferential attachment networks resulted in the worst performance among all tested networks, while small-worlds networks appeared to be much easier to solve.

Figure 3.6 shows the performance of the experimental subjects on networks of 24 nodes, each admitting a perfect matching. In particular, it reports results for single games, and it compares the time to reach a perfect matching of size 12 (red), an approximate matching of size 11 (a 0.92-approximate matching, in blue) and a matching of size 6 (a 1/2-approximate matching, in green) in each game. The x-axis shows the indexes of the games sorted by increasing solution time, while the y-axis shows time in seconds. The plot shows (consistent with the theoretical analysis) that a 1/2-approximate matching is reached almost immediately in all games, an almost maximum matching is reached quickly, while reaching a perfect matching can take a large amount of time.

3.7 Conclusions

While it is challenging to characterize the strategies used by humans in performing even simple social tasks, as they may depend on diverse individual cognitive and psychological attitudes, we argue that it is possible to isolate simple behavioral invariants of individual behavior, which are useful for algorithmic modeling, analysis and prediction of collective dynamics of coordination.

To illustrate our approach, we have focused on a simple matching game over networks and presented a combination of theoretical, experimental, and simulation results. From the experiments, we identified the prudence property as a common behavioral invariant of human subjects when they coordinate to find a maximum matching. We proposed an algorithm as model of human behavior and showed that it can successfully predict dynamics of coordination.

We have shown that our approach is able to uncover basic behavioral properties that may not be apparent from off-line surveys. Indeed, when subjects were asked to report on their strategies in post-experimental surveys, we obtained a list of diverse strategies, including: choose a partner and never disengage from it, always accept proposals from neighbors, try to change partner if the game is not solved for a while. Moreover, our results demonstrate that algorithmic modeling and the mathematical analysis of algorithms can be useful in systematically predicting the aggregate behavior and in deriving results that hold for any graph, or for a large family of graphs. This general conclusions cannot be derived rigorously form experimental observations and computer simulations.

Our work suggests further research in several directions. A natural question is whether non-prudent behavior by a subset of the nodes can help. In a preliminary investigation, we have evaluated the performance of a variant of our algorithm where a subset of nodes behave non-prudently with a positive probability. In our simulations, these populations do not offer significant improvement in terms of finding a maximum matching. Furthermore, populations entirely composed of non-prudent nodes seem to perform poorly. In other words, a group of aggressive and risk-taking individuals might not achieve coordination easily.

Our PRUDENCE algorithm is memoryless. It is an interesting question as to what extent human subjects use memory in distributed games, and how memory could be incorporated in modeling human strategies. In an initial attempt to study this, we implemented a variant of the PRUDENCE algorithm in which a node remembers its recent history and gives less preference to neighbors who recently rejected it. In simulations on preferential attachment and small world networks, memory did not result in significant improvement over the memoryless case. Furthermore, simulations show that making decisions based on events in a distant past (that is, tracking events that happened in a distant past) might hurt performance. A careful investigation of the role of memory in human strategies in distributed games is of fundamental interest.

Regarding the incentives, in our matching games each subject obtains the same reward when a maximum matching is reached, regardless of the chosen partner. How does the introduction of preferences affect the overall coordination? Preferences could be “enforced” for example by rewarding subjects based on the partners they match with. There is likely to be a trade-off between the collective task of finding a maximum matching and the individual profit maximization.

As a final remark, the proposed PRUDENCE algorithm constitutes a *possible* reasonable explanation of human coordination behavior in the distributed matching game. Apart from the simple variations mentioned above, we did not test how well other alternative algorithmic models could fit the experimental data.

3.8 Appendix: analysis for Theorems 20–21

We say that the nodes $\{a_i : 1 \leq i \leq n\} \cup \{b_i : 1 \leq i \leq n\}$ constitute the upper half of G_n, and the remaining ones constitute the lower half of G_n. Let $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$ be the set of all matchings of G_n of size $2n-1$, where $\mathcal{M}_1$ is the set of matchings of size $2n-1$ in which the two unmatched nodes are in opposite halves of G_n, and $\mathcal{M}_2 = \mathcal{M} \setminus \mathcal{M}_1$ are the remaining ones.

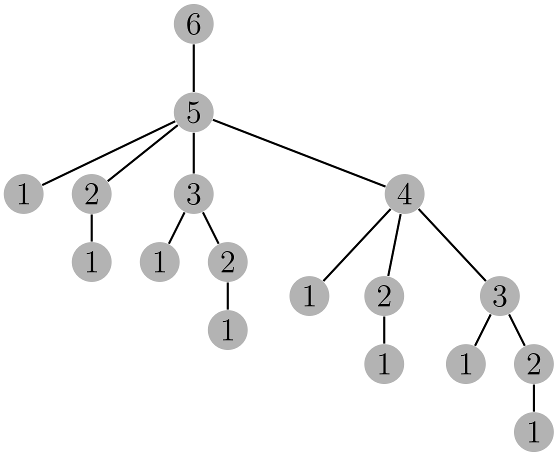

Figure 3.8: **Tree T_n^*.** Tree T_n^* with labels, for $n = 6$. This is used in the proof of Theorem 20.

Our goal is to show that the Prudence algorithm requires $2^{\Omega(n/\log^2 n)}$ rounds with high probability to reach the perfect matching of G_n when starting from any matching in $\mathcal{M}_1$. We first prove certain properties for the matchings in $\mathcal{M}_1$. We then establish a correspondence between the Markov chain over matchings induced by the Prudence algorithm and a classical random walk on the tree T_n^*. In particular, we show that the hitting time of the root of T_n^* is a lower bound on the number of rounds to reach the perfect matching of G_n.

3.8.1 Properties of matchings in $\mathcal{M}_1$.

We begin by characterizing the matchings in $\mathcal{M}_1$.

Lemma 22. *Consider any matching $M \in \mathcal{M}_1$, and let a_k, b_ℓ be the unmatched nodes in the upper and lower half of G_n, respectively. Then, the following properties hold:*

1. *For all $i < k$ and $i > \ell$, the matching M contains the edges (a_i, b_i).*

2. *If $k < n$, M contains the edge (a_n, b_j) for some $1 \leq j < n$. Similarly, if $\ell > n + 1$, M contains the edge (a_i, b_{n+1}) for some $n + 1 < i \leq 2n$. That is, the nodes a_n and b_{n+1} can be matched only through non-horizontal edges.*

3. *If in its upper half M contains a pair of edges $(a_{i_1}, b_{j_1}), (a_{i_2}, b_{j_2})$ with $i_1 \neq j_1$, $i_2 \neq j_2$, and $1 \leq i_1 < i_2 \leq n$, then $1 \leq k \leq j_1 < i_1 \leq j_2 < i_2 \leq n$. Similarly, if in its lower half M contains a pair of edges $(a_{i_1}, b_{j_1}), (a_{i_2}, b_{j_2})$ with $i_1 \neq j_1$, $i_2 \neq j_2$, and $n+1 \leq j_1 < j_2 \leq 2n$, then $n+1 \leq j_1 < i_1 \leq j_2 < i_2 \leq \ell \leq 2n$. That is, non-horizontal matching edges do not cross.*

Proof. To prove the first property, we show that $(a_i, b_i) \in M$ for all $i < k$ (the claim for $i > \ell$ can be proven in the same way). We show by induction on $1 \leq j \leq k-1$ that $(a_i, b_i) \in M$ for all $i \leq j$. For $j = 1$, we have that a_1 must be matched to b_1 (its only neighbor), and therefore the claim holds true. Suppose the claim holds true for some $j < k-1$. By the inductive assumption we have that $(a_i, b_i) \in M$ for all $i \leq j$. As $(a_{j+1}, b_i) \in E$ if and only if $i \leq j+1$, a_{j+1} must be matched to b_{j+1}, and therefore the claim holds for $j+1$.

The second property follows by observing that $M \in \mathcal{M}_1$ implies that the bridge edge (a_{n+1}, b_n) is in M, and therefore a_n cannot be matched to b_n, and a_{n+1} cannot be matched to b_{n+1} in M. To see this, suppose by contradiction that $(a_{n+1}, b_n) \notin M$. Then, b_n must be matched to a_n (its only neighbor besides a_{n+1}), and a node in $\{a_1, \ldots, a_{n-1}\}$ is unmatched. Then, each of the $n-1$ nodes in $\{b_1, \ldots, b_{n+1}\}$ must be matched with one of the $n-2$ matched nodes in $\{a_1, \ldots, a_{n-1}\}$, generating a contradiction. This implies that $(a_{n+1}, b_n) \in M$.

To prove the third property, assume that, in its upper half, M contains edges $(a_{i_1}, b_{j_1}), (a_{i_2}, b_{j_2})$ with $i_1 \neq j_1$, $i_2 \neq j_2$, and $1 \leq i_1 < i_2 \leq n$. Then, it must be that $j_1 < i_1$ and $j_2 < i_2$. Moreover, Property 1 implies that $k \leq j_1$. Therefore, it only remains to show that $i_1 \leq j_2$. Suppose by contradiction that $i_1 > j_2$. As $i_1 > j_1 \geq k$ and $j_1 \neq j_2$, it must be that $i_1 \geq k+2$. As b_{j_2} is matched to a_{i_2} and $i_2 > i_1$, we have that each of the $i_1 - k \geq 2$ nodes in $\{a_{k+1}, \ldots, a_{i_1}\}$ must be matched to one of the $i_1 - k - 1$ nodes in $\{b_k, \ldots, b_{i_1 - 1}\} \setminus \{b_{j_2}\}$, generating a contradiction. This implies that $i_1 \leq j_2$ and therefore $1 \leq k \leq j_1 < i_1 \leq j_2 < i_2 \leq n$. The claim in Property 3 regarding the lower half of M is similarly proved. □

It follows from Lemma 22 that a matching $M \in \mathcal{M}_1$ can be uniquely reconstructed by specifying the two unmatched nodes and the nodes in $\{a_1, \ldots, a_n\} \cup \{b_{n+1}, \ldots, b_{2n}\}$ whose matching edges are *non-horizontal.* To see this, consider the upper half of G_n: assume $a_{j_0} \neq a_n$ is the unmatched node and $S = \{j_1, \ldots, j_m\}$, with $1 \leq j_0 < j_1 < j_2 < \ldots < j_m = n$, is the set of indexes of the left nodes whose matching edges are non-horizontal. (Note that $n \in S$ by Lemma 22.) Then, $j_0 < j_1$ and $(a_i, b_i) \in M$ for all i such that $i \notin S \cup \{j_0\}$ and $1 \leq i \leq n$. Hence, it necessarily holds that $(a_{j_i}, b_{j_{i-1}}) \in M$ for all $1 \leq i \leq m$. This completes the construction of the matching in the upper half of G_n. A similar argument can be applied to the lower half. These two arguments imply the following lemma.

Lemma 23. *There exists a bijection ψ between matchings in $\mathcal{M}_1$ and elements of $\mathcal{P} \times \mathcal{P}'$, where*

$$\begin{aligned}\mathcal{P} &= \big\{(x, S) : x \in \{1, \ldots, n-1\}, \{n\} \subseteq S \subseteq \{x+1, \ldots, n\}\big\} \cup \{(n, \emptyset)\},\\ \mathcal{P}' &= \big\{(y, S') : y \in \{n+1, \ldots, 2n\}, \{n+1\} \subseteq S' \subseteq \{n+1, \ldots, y-1\}\big\}\\ &\quad \cup \{(n+1, \emptyset)\}.\end{aligned}$$

3.8.2 The tree T_n^*.

Definition 24. *Let T_1 be a labelled rooted tree with a singleton node with label 1. Inductively, for $2 \leq i \leq n-1$, let T_i be the labelled rooted tree whose root is labelled with i and its children are $T_1, \ldots, T_{i-1}$. We define T_n^* to be the tree with an unlabelled root whose only child is T_n (also see Figure S1). Let r^* denote the root of T_n^*.*

We show that the hitting time of r^* when starting at any node $u \neq r^*$ is exponential with high probability. For a node $\neq r^*$, we call the edge that connects u to its parent u's *exit* edge. For any subtree $T_i \subset T_n^*$, let Z_i be the random variable denoting the number of steps that it takes for a walk starting at the root of T_i to "exit" T_i. The following lemma provides an exponential lower bound on Z_i.

Lemma 25. *There exist positive constants $\alpha, \gamma > 0$ such that, for all $i \geq 2$,*

$$\Pr[Z_i \geq \gamma \cdot 2^{i/(\alpha \log^2 i)}] \geq 1 - \frac{1}{\log i}.$$

Proof. We proceed by induction on i. For convenience, define $g(i) = \alpha \log^2 i$ and $f(i) = \gamma \cdot 2^{i/g(i)}$ for some $\alpha, \gamma > 0$. For any constant $\alpha > 0$, there exists a small enough constant $\gamma > 0$ such that $f(i) \leq 1$; therefore, as $Z_i \geq 1$ with probability 1, the claim holds trivially for any $i \leq i^*$, where i^* is a suitable large constant.

Now consider any $i \geq i^*$ and suppose the claim holds up to $i-1$. Every time the walk is on the root of T_i, it exits T_i with probability $1/i$. Therefore, letting E_t be the event that the first t times the walk is on the root of T_i it does *not* exit T_i, we have $\Pr[E_t] \geq 1 - t/i$. Let $t = i/(2\log i)$, and let D_j, $1 \leq j \leq t$, be the event that, when it is on the root of T_i for the j-th time, the walk moves to the root of one of the subtrees $T_{i-g(i)}, \ldots, T_{i-1}$ *and* takes at least $f(i-g(i))$ steps to exit that subtree. For $1 \leq j \leq t$, we have

$$\begin{aligned}\Pr[D_j \mid E_t] &\geq \frac{g(i)}{i} \cdot \Pr[Z_{i-g(i)} \geq f(i-g(i))] \\ &\geq \frac{g(i)}{i} \cdot \left(1 - \frac{1}{\log(i-g(i))}\right),\end{aligned}$$

by the induction hypothesis on $Z_{i-g(i)}$. Letting χ_j be the indicator function of the event D_j for $1 \leq j \leq t$, the probability that at least two of the events D_j happen, given E_t, is lower bounded by:

$$\begin{aligned}\Pr\left[\sum_{j=1}^{t} \chi_j \geq 2 \;\middle|\; E_t\right] &\geq \Pr\left[\sum_{j=1}^{t/2} \chi_j \geq 1, \sum_{j=t/2+1}^{t} \chi_j \geq 1 \;\middle|\; E_t\right] \\ &= \Pr\left[\sum_{j=1}^{t/2} \chi_j \geq 1 \;\middle|\; E_t\right]^2.\end{aligned}$$

By union bound, we can write

$$\begin{aligned}\Pr\left[\sum_{j=1}^{t/2} \chi_j \geq 1 \;\middle|\; E_t\right] &\geq 1 - \prod_{i=1}^{t/2} (1 - \Pr[D_j|E_t]) \\ &\geq 1 - \left(1 - \frac{g(i)}{i}\left(1 - \frac{1}{\log(i-g(i))}\right)\right)^{t/2} \\ &\geq 1 - \exp\left[-\frac{\alpha \log i}{4}\left(1 - \frac{1}{\log(i-g(i))}\right)\right] \\ &\geq 1 - \frac{1}{i^{\alpha/8}},\end{aligned}$$

where the last step holds for i sufficiently large so that $\log(i-g(i)) \geq 2$. This implies that

$$\Pr\left[\sum_{j=1}^{t} \chi_j \geq 2 \,\middle|\, E_t\right] \geq \left(1 - \frac{1}{i^{\alpha/8}}\right)^2 \geq 1 - \frac{2}{i^{\alpha/8}}.$$

Therefore, we conclude that

$$\begin{aligned}
\Pr[Z_i \geq 2 \cdot f(i - g(i))] &\geq \Pr\left[\sum_{j=1}^{t} \chi_j \geq 2\right] \\
&\geq \Pr\left[\sum_{j=1}^{t} \chi_j \geq 2 \middle| E_t\right] \Pr[E_t] \\
&\geq \left(1 - \frac{2}{i^{\alpha/8}}\right)\left(1 - \frac{t}{i}\right) \\
&\geq 1 - \frac{1}{\log i},
\end{aligned}$$

where the last step holds by choosing α sufficiently large. The claim now follows since $2 \cdot f(i - g(i)) \geq f(i)$. □

Note that any random walk starting at any node $u \neq r^*$ has to exit T_n before hitting r^*. Therefore, an application of Lemma 25 to T_n yields a lower bound to the hitting time of r^* when starting at any node $u \neq r^*$.

Corollary 26. *The hitting time of r^* of a random walk starting at any node $u \neq r^*$ is $2^{\Omega(n/\log^2 n)}$ with high probability.*

3.8.3 Proof of Theorem 20

For $t \geq 0$, let $\mathcal{M}(t)$ be the matching at the beginning of round t and assume $\mathcal{M}(0) \in \mathcal{M}_1$. To analyze the convergence to a perfect matching, we will consider on the event that $\mathcal{M}(t) \notin \mathcal{M}_1$. Note that in order for this event to happen, the bridge edge (a_{n+1}, b_n) of G_n will have to swap out of the matching. Let $E(t)$ be the event that a_n requests b_n during round t. Similarly, let $E'(t)$ be the event that b_{n+1} requests a_{n+1} during round t. Define the random variables

$$\begin{aligned}
\tau_n &= \min\{t : E(t) \text{ happens}\}, \\
\tau'_n &= \min\{t : E'(t) \text{ happens}\}, \\
\tau^*_n &= \min\{\tau_n, \tau'_n\}.
\end{aligned}$$

Then τ_n^* is a lower bound on the number of rounds to reach the perfect matching. Lemma 27 below states that, for some $c > 0$,

$$\Pr\left[\tau_n \le 2^{cn/\log^2 n} \;\middle|\; \tau_n \le \tau_n'\right] = o(1) \quad \text{and} \quad \Pr\left[\tau_n' \le 2^{cn/\log^2 n} \;\middle|\; \tau_n' \le \tau_n\right] = o(1).$$

Then the main theorem follows as

$$\begin{aligned} \Pr\left[\tau_n^* \le 2^{cn/\log^2 n}\right] &= \Pr\left[\tau_n^* \le 2^{cn/\log^2 n} \;\middle|\; \tau_n \le \tau_n'\right] \Pr\left[\tau_n \le \tau_n'\right] \\ &\quad + \Pr\left[\tau_n^* \le 2^{cn/\log^2 n} \;\middle|\; \tau_n' < \tau_n\right] \Pr\left[\tau_n' < \tau_n\right] \\ &= \Pr\left[\tau_n \le 2^{cn/\log^2 n} \;\middle|\; \tau_n \le \tau_n'\right] \Pr\left[\tau_n \le \tau_n'\right] \\ &\quad + \Pr\left[\tau_n' \le 2^{cn/\log^2 n} \;\middle|\; \tau_n' < \tau_n\right] \Pr\left[\tau_n' < \tau_n\right] \\ &= o(1). \end{aligned}$$

Lemma 27.

$$\Pr\left[\tau_n \le 2^{cn/\log^2 n} \;\middle|\; \tau_n \le \tau_n'\right] = o(1) \quad \textit{and} \quad \Pr\left[\tau_n' \le 2^{cn/\log^2 n} \;\middle|\; \tau_n' \le \tau_n\right] = o(1).$$

Proof. We will prove the first bound. The second one follows by symmetry. Conditioning on the event that $\tau_n \le \tau_n'$, we will analyze the matching in the upper half of G_n induced by $\mathcal{M}(t)$. Since $\tau_n \le \tau_n'$, $\mathcal{M}(t) \in \mathcal{M}_1$ as long as $E(t)$ does not happen. By Lemma 23, it is equivalent to study the Markov process $\{(X(t), \mathcal{S}(t)), t \ge 0\}$ over $\mathcal{P} \cup \{(\perp, \emptyset)\}$, where $(X(t), \mathcal{S}(t))$ is defined as the first marginal of $\psi(\mathcal{M}(t))$, and the additional state $(\perp, \emptyset)$ is reached when the event $E(t)$ happens. That is, conditioning on the event $\tau_n \le \tau_n'$, it follows that

$$\tau_n = \min\{t : (X(t), \mathcal{S}(t)) = (\perp, \emptyset)\}. \tag{3.2}$$

If $\tau_n \le \tau_n'$ and $(X(t), \mathcal{S}(t)) \ne (\perp, \emptyset)$, all the neighbors of the unmatched node in the upper half of G_n are matched at the beginning of round t, and hence are *equally likely* to be requested during round t. Therefore, the Markov process $(X(t), \mathcal{S}(t))$ has the following transition probabilities.

$$\Pr\left[(X(t+1), \mathcal{S}(t+1)) = (x', S') \;\middle|\; (X(t), \mathcal{S}(t)) = (x, S) \ne (\perp, \emptyset), \tau_n \le \tau_n'\right] = \frac{1}{x},$$

for any

$$(x', S') \in \begin{cases} \big\{(x'', S \cup x) : x'' < x\big\} \cup \big\{(\min(S), S \backslash \min(S))\big\}, & \text{if } x < n, S \neq \emptyset \\ \\ \big\{(x'', S \cup x) : x'' < x\big\} \cup \big\{(\bot, \emptyset)\big\}, & \text{if } x = n, S = \emptyset \end{cases}$$

The case $(x', S') \in \{(x'', S \cup x) : x'' < x\}$ represents the scenario in which the unmatched node a_x requests a node through a non-horizontal edge: in this case, no progress is made as the unmatched node in the next round will be further away from a_n. If the unmatched node a_x requests the node on its horizontal edge, the next unmatched node will be closer to a_n. In the special case $(x, S) = (n, \emptyset)$, if the unmatched node requests the neighbor on its horizontal edge, then the bridge edge is swapped out of the matching and $\mathcal{M}(t+1) \notin \mathcal{M}_1$.

We will now show that the Markov chain $\{(X(t), \mathcal{S}(t)), t \geq 0\}$ is equivalent to the random walk on T_n^*. For a node v of T_n^*, let x_v be its label and S_v be the set of labels of its ancestors. Define the function ϕ from nodes of T_n^* to states of the chain as follows:

$$\phi(v) = \begin{cases} (\bot, \emptyset), & v = r^* \\ (x_v, S_v), & v \neq r^* \end{cases}$$

It is easy to verify that ϕ is a bijection. Two nodes u and v are adjacent in T_n^* if and only if there is a nonzero transition probability between the states $\phi(u)$ and $\phi(v)$. To see this, suppose there is a nonzero transition probability from (x_u, S_u) to (x_v, S_v) in the Markov chain. Let $u = \phi^{-1}(x_u, S_u)$ and $v = \phi^{-1}(x_v, S_v)$ be the corresponding nodes in T_n^*. There are two cases: (a) if $x_v < x_u$ then $S_v = S_u \cup x_u$, and v is a child of u; (b) if $x_v > x_u$ then $x_v = \min(S_u)$, $S_v = S_u \backslash \min(S_u)$, and v is the parent of u. The other direction is analogous. Therefore, conditioning on $\tau_n \leq \tau_n'$ and $(X(0), \mathcal{S}(0)) \neq (\bot, \emptyset)$, we can conclude that $\min\{t : (X(t), \mathcal{S}(t)) = (\bot, \emptyset)\}$ equals the hitting time of r^* for a random walk on T_n^* starting at the node $\phi^{-1}(X(0), \mathcal{S}(0)) \neq r^*$. The lemma follows by equation (3.2) and Corollary 26. □

3.8.4 Proof of Theorem 21

As in the proof of Theorem 20, we let $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$ be the set of all matchings of G_n of size $2n-1$, where $\mathcal{M}_1$ and $\mathcal{M}_2$ contain all matchings in which the two unmatched nodes are in opposite sides of G_n and in the same side of G_n, respectively. By Theorem 20 we know that starting from any matching in $\mathcal{M}_1$ requires exponentially many steps to reach the perfect matching of G_n with high probability. We will show that these matchings substantially make up for the whole $\mathcal{M}$. Indeed, we prove that

$$|\mathcal{M}_1| = 2^{2n-2}, \qquad \text{and} \qquad |\mathcal{M}_2| = 2^{n+1} - 2.$$

To compute the size of $\mathcal{M}_1$, using Corollary 23 we have that

$$|\mathcal{M}_1| = \left(1 + \sum_{i=1}^{n-1} 2^{n-i-1}\right)^2 = \left(1 + \sum_{j=0}^{n-2} 2^j\right)^2 = \big(1 + (2^{n-1} - 1)\big)^2 = 2^{2n-2}.$$

To compute the size of $\mathcal{M}_2$, let $\mathcal{M}_2'$ contain the matchings of $\mathcal{M}_2$ in which the two unmatched nodes are in the upper half of G_n. Observe that by symmetry $|\mathcal{M}_2| = 2 \cdot |\mathcal{M}_2'|$. To determine the size of $\mathcal{M}_2'$, note first that every matching in $\mathcal{M}_2'$ is such that the nodes in the lower half of G_n are matched through parallel edges, i.e. a_j is matched with b_j for every $n+1 \leq j \leq 2n$. Now consider all matchings in $\mathcal{M}_2'$ where a_k, b_ℓ are the two unmatched nodes, and observe that it must be that $1 \leq k \leq \ell \leq n$ (if not, we would have at least another unmatched node a_t with $t < \ell$). Also, note that for every $1 \leq j \leq k-1$ and every $\ell+1 \leq j \leq n$, it must be that a_j is matched with b_j. Hence, for $k = \ell$, there is a single matching. For $k < \ell$, we show that the remaining nodes can be matched in $2^{\ell-k-1}$ ways. To prove this, first observe that a_{k+1} can be matched to either b_{k+1} or b_k. Then, given the choice for a_{k+1}, a_{k+2} can be matched to either b_{k+2} or the node in $\{b_{k+1}, b_k\}$ which is not matched to a_{k+1}. Similarly, for $i+1 \leq j \leq \ell - 1$, there are two possible choices for a_j given the choice for $\{a_{k+1}, \ldots, a_{j-1}\}$. Finally, given the choices for $\{a_{k+1}, \ldots, a_{\ell-1}\}$ there is only one possible match for a_ℓ, thus obtaining

$2^{\ell-k-1}$ matchings with a_k, b_ℓ unmatched, $1 \leq k < \ell \leq n$. We can conclude that

$$\begin{aligned}
|\mathcal{M}_2'| &= \sum_{k=1}^{n} \left(1 + \sum_{\ell=k+1}^{n} 2^{\ell-k-1} \right) \\
&= n + \sum_{k=1}^{n-1} 2^{-k} \sum_{\ell=k+1}^{n} 2^{\ell-1} \\
&= n + \sum_{k=1}^{n-1} 2^{-k} (2^n - 2^k) \\
&= 1 + \sum_{k=1}^{n-1} 2^{n-k} \\
&= 1 + (2^n - 2) = 2^n - 1.
\end{aligned}$$

□

Chapter 3, in full, is a reprint of the paper "Human matching behavior in social networks: an algorithmic perspective" co-authored with Lorenzo Coviello, Massimo Franceschetti, Mathew D. McCubbins and Ramamohan Paturi, published in PLoS ONE, Volume 7, Number 8, August 2012 [CFM$^+$12]. The dissertation author and Lorenzo Coviello were the primary investigators and authors of this paper.

Chapter 4

Finding Red Balloons

A challenging class of crowdsourcing problems requires an interested party to provide incentives for large groups of people to contribute to the search and retrieval of rare information [Sur04, LRSH11, Han10]. The small world problem, i.e. distributed routing of messages to unknown individuals, is the seminal example of this class and has illustrated the difficulty of the approach for almost 50 years [Mil67, TM69, Sch09, DMW03, WDN02]. In this class of problems, individuals in the social network act as intermediaries to create a channel between the querier and the answer. Observe that the chief difficulty of this approach is to offer incentives to the individuals to propagate the query further in the network as well as to return the answer all the way back to the querier [DMW03]. The goal is therefore to incentivize participation of the users using some form of (possibly financial) reward. In this way, a node who does not know the answer but is offered a sufficiently high reward can act as intermediary and propagate the query by offering the neighbors a share of its reward. This setting models the social network as a marketplace of information where the users strategically act in order to maximize their utility, and raises several questions about the system's performance and the incentive propagation, the main one being: can we retrieve the answer to a difficult query when given a limited budget?

The Defense Advanced Research Projects Agency (DARPA), a research organization of the United States Department of Defense, designed a so called "Network Challenge" that conveyed a positive answer to this question.[1] The challenge consisted of locating ten moored red weather balloons placed at ten undisclosed locations in the continental United States. A single $40,000 cash prize was allocated for the first participant to submit the correct latitude and longitude (within one mile error) of all ten balloons within the contest period. In particular, the competition consisted in recruiting a team to achieve the goal. This task posed varied issues of large-scale, time-critical mobilization. In particular, in order to guarantee the participation and coordination of a large team, an adequate structure of economic incentives had to be built.

The MIT Media Laboratory team won the competition in less than 9 hours, adopting a recruitment scheme based on recursive incentives.[2] Specifically, using the $40,000 they could possibly win, they allocated an amount of $4,000 for finding each balloon. For each balloon, they would distribute the $4,000 up the chain of participants leading to successful balloon spotting, as described in their website: "[In the case we win the competition,] we're giving $2,000 per balloon to the first person to send us the correct coordinates, but that's not all – we're also giving $1000 to the person who invited them. Then we're giving $500 whoever invited the inviter, and $250 to whoever invited them, and so on...". This is equivalent to say that a node u who does not have the desired answer, can offer its friends a $1/2$-*split* contract, stipulating that if the answer is found in the subtree of a child v of u, then u will get back from v a $1/2$ fraction of whatever amount v gets. However, if u is not the querier, the total amount pocketed by u is less, as u has to give a $1/2$ fraction of its reward to its recruiter.

While the success of this strategy has been hailed as an empirical testimony to the power of incentive structures [TCG$^+$11], the theoretical efficiency of the proven scheme has remained an open question, and motivates this work. In particular, we analyze this economic structure in the model for query incentive networks introduced By Kleinberg and Raghavan in [KR05]. This model considers a competitive environment where every node plays strategically. To fit the *split contracts* to this model, we generalize the splits to any fraction $0 < \rho < 1$, in

[1] `https://networkchallenge.darpa.mil/`

[2] `http://balloon.media.mit.edu/`

the sense that any node u can offer a child v a ρ-split contract stipulating the following: if v has the answer, then v would pocket a $(1-\rho)$ of the whole reward while returning a fraction ρ to u; if v does not have the answer, then v can in turn offer some ρ'-split to its (still unrecruited) friends, and so on. Given the strategic setting, nodes will choose the splits to offer to their children so to maximize their expected payoffs; observe that contracts between different nodes can have different splits — and this is indeed the case in the Nash equilibrium as our results show. The details of the original model introduced in [KR05] follow.

4.1 The query incentive network model

The scenario of interest is that of a node, the root, that is willing to invest some amount r^* to retrieve certain information from a large network in which every node plays strategically. The main goal is to characterize the tradeoff between the investment and the rarity of the information. The model, introduced by Kleinberg and Raghavan [KR05], is as follows: the querier node is the root of an infinite d-ary tree, where each node possesses independently the desired information with probability $1/n$, where n represents the rarity of the answer. The root offers each child u a "*fixed-payment*" contract of r^*, stipulating that the root will pay u that amount upon u providing the answer. The query propagates down the tree according to the following scheme: every node u has an integer-valued function f_u encoding its strategies; if u is offered a reward of r by its parent and does not possess the answer, then in turn it offers a reward of $1 \leq f_u(r) \leq r-1$ to its children. When the answer to the query is found, the root selects for payment one among the answer-holders using a fixed non-strategic rule. The payment is then propagated down through the path to that selected node, with each node along the path pocketing its share. If an intermediate node u on this path was offered r by its parent, then its overall payoff is $r - f_u(r) - 1$, where the *unit cost* is associated with the act of returning the answer[3]. The game-theoretical aspect of the model is

[3] As observed in [KR05], if nodes placed no value on this answering effort then the root could simply invest an arbitrarily small reward $\epsilon > 0$, and it would retrieve an answer because each node would have a positive payoff from participating in the game and returning the answer. To

that any node u chooses the function f_u so to maximize its payoff. To break ties, it is assumed that a node who is offered a reward of one (and does not possess the answer) will always forward the query to its children, even if its expected payoff is zero (since the unit reward would be spent when returning the answer up to its parent).

As pointed out in [KR05], there is a subtle deficiency with a deterministic tree: the Nash equilibria of a game played in a deterministic network tacitly assume that the nodes know the entire network. Indeed, in a Nash equilibrium, each node chooses its best strategy by knowing the strategies of every other node. However, this is unrealistic, as we want to model a setting where nodes are only aware of their neighbors. To deal with this technical issue, Kleinberg and Raghavan consider a network that can be thought as a branching process from the root. In particular, the number of children of each node is chosen independently from a binomial distribution $\mathrm{Bin}(d, q)$, where q is a constant probability of a node being present. The expected number of children of a node — i.e., the branching factor — is then $b = qd$. By classical results in the theory of branching processes, if $b < 1$ the process dies out almost surely; therefore there is no amount that the root can offer to obtain an answer with constant probability if the rarity n of the answer is large enough. Instead, for any $b > 1$, there is a constant non-zero probability that the process will generate infinitely many nodes, so that the answer is present within the first $O(\log n)$ levels of the tree with high probability. Nevertheless, Kleinberg and Raghavan show that in the Nash equilibrium the investment needed at the root can be much larger than logarithmic in n. Specifically, while an investment $r^* = O(\log n)$ is sufficient to retrieve the answer with constant probability for $b > 2$, an investment of $r^* = n^{\Theta(1)}$ is needed when $1 < b < 2$. That is, in the latter case the root must invest a reward that is exponentially larger than the expected distance from the closest answer.

Arcaute *et al.* [AKK$^+$07] generalized the work in [KR05] showing that this

avoid this situation, a unit price is placed on the effort of returning the answer, while the cost of participating to the game is zero. This is motivated by the fact that the cost of forwarding requests to a list of friends is typically considered negligible in peer-to-peer and social-network systems [KSS97, YS03, ZA04] (see [KR05] for additional details on the motivations).

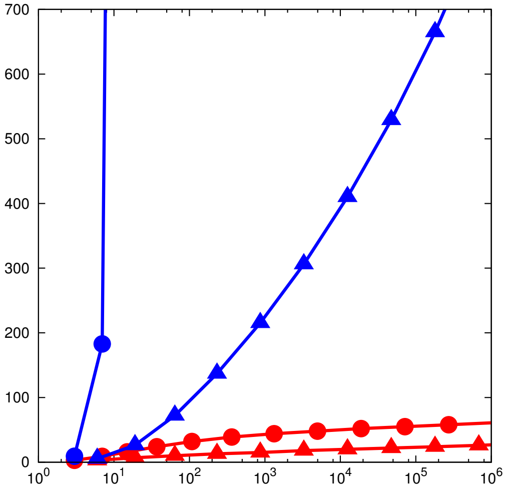

Figure 4.1: Investment as a function of the rarity n for $b = 1.95$ with split-contracts (in red triangles and circles, for $\epsilon = 0.2$ and $\epsilon = 0.05$, respectively) and with fixed-payment contracts (in blue triangles and circles, for $\epsilon = 0.2$ and $\epsilon = 0.05$, respectively).

threshold behavior at $b = 2$ still holds for arbitrary Galton-Watson branching process. They also proved that in a *ray* — a deterministic infinite path ($b = 1$, but with zero extinction probability) — the reward needed is super-exponential in the expected depth of the search tree, that is $r^* = \Omega(n!)$. Finally, they observed that this threshold behavior vanishes if the root desires to find the answer with probability tending to 1: if the desired probability is $1-1/n$, then for any branching process with $b > 1$ and no extinction, the needed reward is $n^{\Theta(1)}$.

4.2 Results

We present a theoretical study of the multi-level marketing strategies that were adopted by the winning team of the DARPA Network Challenge. Given the strong affinity between this challenge and the model of query incentive networks introduced in [KR05, AKK$^+$07], we frame these strategies in this model by considering split contracts as the possible offers between nodes.

Our main result is that split contracts, unlike fixed-payment contracts, are robust to a strategic environment, where every node selfishly determines the offers

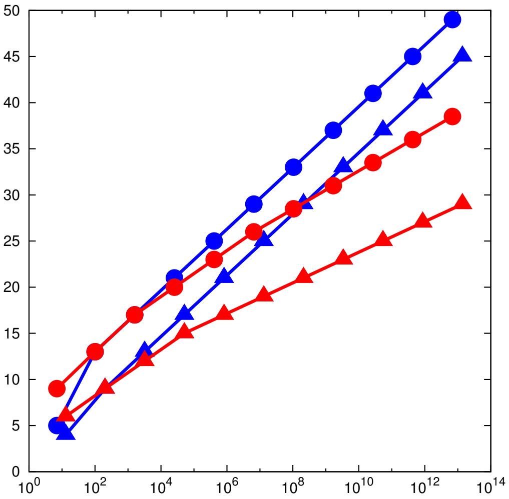

Figure 4.2: Investment as a function of the rarity n for $b = 4$ with split-contracts (in red triangles and circles, for $\epsilon = 0.2$ and $\epsilon = 0.05$, respectively) and with fixed-payment contracts (in blue triangles and circles, for $\epsilon = 0.2$ and $\epsilon = 0.05$, respectively).

to its children based on the offer received from its parent. We show that for any constant $\epsilon > 0$ and Galton-Watson branching process with $b > 1$, the Nash equilibrium with split contracts uses an investment of $r^* = O(\log n)$ to retrieve the answer with probability at least $1 - \zeta - \epsilon$, where ζ is the extinction probability of the process. As the expected distance to the closest answer is $\Theta(\log n)$ and nodes pay a unit cost to return the answer, this is a constant approximation with respect to an ideal centralized non-strategic setting. In other words, the price of anarchy of the game with split contracts is constant (ignoring some pathological equilibria, see Section 4.6 and Appendix 4.9).

Unlike previous work that assumed the parameters of the branching process to be held constant, we are also able to characterize the dependence of the investment with respect to the branching process and the success accuracy. This allows us to show additional improvements of split contracts over fixed-payment contracts: for example, for branching processes with no extinction, an investment of $O(n \log n)$ is enough to retrieve the answer with probability at least $1 - 1/n$, improving upon the $n^{\Theta(1)}$ investment provided in [AKK$^+$07]. In fact, our result is even stronger since it guarantees a success probability of at least $1 - \zeta - 1/n$ in general branching processes. In the case of a ray (where the expected distance from the closest answer is n), we show that the investment needed to find the answer with constant probability is $O(n^2)$, while $\Omega(n!)$ is needed when using fixed-payment contracts [AKK+07].

We also performed empirical simulations to compare the amount of investment needed with fixed-payment and split contracts. Our findings show that split contracts are more efficient than fixed-payment contracts not only for small branching factors $1 < b < 2$ — with an exponential gain, see Figure 4.1 — but also for large branching factors $b > 2$ — with a constant-factor gain, see Figure 4.2. Simulations demonstrate that the improvement is major already for b slightly less than 2. In addition, finer accuracy ϵ leads to a larger improvement.

4.3 Additional related work

Pickard et al. [PPR+11] described and analyzed the winning strategy of the DARPA Network Challenge. However, we distinguish ourselves from [PPR+11] in both aims and methods. The authors of [PPR+11] are mainly concerned with the motivation of the exact $1/2$-split winning strategy that was implemented by the MIT Media Laboratory, for which they show that it is in the participants' interest to recruit the highest number of friends and back the theory with an empirical analysis of the diffusion cascades. Our work considers the more general setting of split contracts in the model of query incentive networks introduced in [KR05] and analyzes the efficiency, in terms of investment, of the Nash equilibria.

In the context of query incentive networks with fixed-payment contracts, Kota and Narahari [KN10] applied the results of general branching processes from [AKK+07] to analyze the reward when the degree distribution follows a power-law and the desired success probability is at least $1 - 1/n$ and show a threshold behavior of the reward with respect to the scaling exponent. Dikshit and Yadati [DN09] considered the issue of the quality of the answers in query incentive networks. In particular, they define a quality conscious model of incentives and derive the same threshold behavior around the branching factor $b = 2$ found in [AKK+07, KR05].

It is worth to mention additional related work that is not in the context of query incentive networks. Emek et al. [EKTZ11] studied strategies of multi-level marketing, in which each individual is rewarded according to direct and indirect referrals, and show that geometric reward schemes are the only guarantee to certain desirable properties. Our setting is substantially different from [EKTZ11], as the reward is based on referral rather that information retrieval. Douceur and Moscibroda [DM07] proposed the lottery tree as a mechanism to incentivize the adoption of a distributed systems and the solicitation of new participants. Influence in social networks is also related to our work. Kempe et al. [KKT03] considered the algorithmic question of selecting an influential set of individuals. Jackson and Yariv [JY07] proposed a game-theoretic framework to model incentives in adoption processes. Hartline et al. [HMS08] studied influence in social networks from a revenue maximization point of view. Singer [Sin11] developed incentive-compatible mechanisms for influence maximization in several models.

4.4 Preliminaries

We model the network as a tree generated via a Galton-Watson branching process with offspring distribution $\{c_k\}_{k=0}^{d}$, that is, c_k is the probability that any node has exactly k children and $\sum_{k=0}^{d} c_k = 1$. We adopt the convention that the root of the tree is at level 0, its children at level 1, and so on. The probability generating function of the offspring distribution is given by

$$\Psi(x) = \sum_{k=1}^{d} c_k x^k, \qquad 0 \leq x \leq 1.$$

The branching factor of the process is defined as

$$b = \Psi'(1) = \sum_{k=0}^{d} k c_k.$$

A fundamental result in the theory of Galton-Watson processes states that the extinction probability ζ of a branching process is the smallest non-negative root of the equation $x = \Psi(x)$. If follows $\zeta = 1$ if and only if $b < 1$, or $b = 1$ with $c_0 > 0$, and $0 \leq \zeta < 1$ otherwise. For classical theory on Galton-Watson branching processes we refer to [AN04].

We assume that each node in the network possesses the answer to the query independently of the other nodes with probability $1/n$, where n represents the *rarity* of the answer. Note that n is the expected number of nodes to query before finding the answer. For $i \geq 0$, let ϕ_i be the probability that no node at level $j \leq i$ has the answer and $\lambda_i = \phi_{i-1} - \phi_i$ be the probability that some node at level i and no node at a lower level possesses the answer. (These probabilities are over the randomness of the branching process and of the process assigning the answers.) Moreover, conditional on the event that the branching process with probability generating function Ψ does not die out, let $h_\Psi(\epsilon, n)$ be the minimum integer i such that $\phi_i < \epsilon$. For branching factor $b > 1$, we have that $h_\Psi(\epsilon, n) = O(\log n)$ for any $\epsilon = n^{-O(1)}$, whereas in the case of $b = 1$ and $c_0 = 0$ (i.e., a ray), $h_\Psi(\epsilon, n) = n \ln \frac{1}{\epsilon}$.

Assume that r^* is the investment available at the root, which desires to retrieve the answer with probability at least $1 - \zeta - \epsilon$, for a given success accuracy $\epsilon > 0$. With the notation introduced so far, we will show that for any constants $b > 1$ and $\epsilon > 0$ an investment of $r^* = O(h_\Psi(\epsilon, n))$ suffices to propagate the query down to level $h_\Psi(\epsilon, n)$ of the tree, and hence to retrieve the answer with probability at least $1 - \zeta - \epsilon$. For ease of analysis, we assume that the root is not willing to explore the tree below level $h_\Psi(\epsilon, n)$, that is, we *truncate* the tree at that height.

4.4.1 Split contracts

We now formalize the notion of *split contracts*. Every node including the root can offer a ρ-split contract to its children, for some $0 < \rho < 1$, stipulating the following. If the root offers a ρ-split to a child u who possesses the answer, then u receives a payment of r^* but is required to return a ρ fraction to the root, earning a total of $r^*(1 - \rho) - 1$, where we introduced a unit cost for returning the answer to the parent, as in [KR05, AKK$^+$07]. If instead u does not possess the answer then it might decide to propagate the query to its children, according to its strategy $f_u(\cdot)$, that is, offering a $f_u(\rho)$-split contract to its children. If one among

u's children possesses the answer, then u receives an $f_u(\rho)$ fraction of the reward but it gives a ρ fraction back to the root and pays the unit cost to return the answer, with an overall earning of $r^*(1-\rho)f_u(\rho)-1$. In general, consider a node u_ℓ which is reached by a query and possesses the answer, and let $u_0, u_1, \ldots, u_\ell$ be the path connecting the root to u_ℓ, where u_0 is the root. Then, if the root offered a ρ_{u_0}-split to its children, and $\rho_{u_i} = f_{u_i}(\rho_{u_{i-1}})$ is the split offered by u_i to its children for all $i < \ell$, then the root u_0 (who need not to pay the unit cost) receives a payoff of

$$r^* \cdot \rho_{u_0} \cdot f_{u_1}(\rho_{u_0}) \cdot f_{u_2}(f_{u_1}(\rho_{u_0})) \cdots f_{u_{\ell-1}}(f_{u_{\ell-2}}(\cdots)) = r^* \cdot \prod_{j=0}^{\ell-1} \rho_{u_j}.$$

Similarly, for $1 \le i \le \ell$, the payoff of node u_i is

$$\left(r^*(1-\rho_{u_{i-1}}) \cdot \prod_{j=i}^{\ell-1} \rho_{u_j} \right) - 1.$$

Without loss of generality, we assume that nodes never propose *useless* split-offers to their children, that is, ρ-split where $\rho > \rho_1 := 1 - 1/r^*$, since their children would not have incentive to play even if they possessed the answer themselves. Also, for simplicity we assume discrete domain and range for the strategy f_u of every node u, that is, $f_u : \mathcal{D}_M \to (\mathcal{D}_M \cup \perp)$, where $f_u(\rho) = \perp$ indicates that u chooses not to propagate the query, and $\mathcal{D}_M = \{\frac{\rho_1}{M}, \frac{2\rho_1}{M}, \ldots, \frac{(M-1)\rho_1}{M}, \rho_1\}$ is a discretization of the interval $(0, \rho_1]$.

4.4.2 Propagation of the payment

We remark that the above payoffs for the path $u_0, \ldots, u_\ell$ will turn into concrete payments only if the root selects u_ℓ among the answer-holders. Indeed, among all answer-holders reached by the query the root, will select only one for payment. In the fixed-payment model of [KR05, AKK$^+$07], this selection is made using a fixed arbitrary procedure that does not affect the strategies of the nodes (e.g., performing a random walk from the root descending down the tree; the first hit answer-holder will be paid along with its ancestors). In their setting, this choice is coherent as the root always spends a fixed investment, no matter how deep in the tree the payment is propagated. In our case this peculiarity is missing as a result of the split contract mechanism. In our model we will assume the root selects for payment one among the answer-holders (reached by the query) at *smallest depth*. This is motivated by different facts. First, if we consider some notion of time related to propagating the query one level down, then our selection mechanism better depicts the strategy adopted in the DARPA Network Challenge, where the payment was given to the first participant reporting the correct location of a balloon. Second, the actual investment of the root is in general smaller if the path to the answer is shorter. Finally, a selection mechanism based on smallest depth alleviates the false-name issue discussed in [PPR+11], but a formal analysis of this claim is beyond the scope of this work. In case of multiple answer-holders at smallest depth, we assume that the root breaks ties in a way that does not affect the strategies of the nodes (e.g., performing a random walk from the root to one of the leaves of the subtree formed by all shortest paths to the answers, and selecting the corresponding answer-holder).

4.4.3 Difference with respect to previous work

We would like to spend a few words highlighting some of the main differences between our analysis and those in [KR05, AKK+07]. One of these differences, the propagation of payments, has been already discussed above; from the technical point of view, the *smallest depth* selection mechanism introduces the hurdle that the strategy of each node does not only depend on the strategies in its subtree (as in the case of [KR05, AKK+07]), but potentially on those of all nodes. We remark that the gap in efficiency of the two models is not related to the different propagation of payment. In fact, if the answer-holder were to be selected according to the smallest depth mechanism in the fixed-payment setting, then the investment needed to retrieve the answer would increase. Roughly speaking, this happens as a node further down in the tree requires higher reward to forward the query, in order to compensate for the smaller probability of having a payment candidate in its subtree.

Another salient difference between the two models concerns the *values* of the contracts: while the nature of the fixed-payment contracts of [KR05, AKK$^+$07] implies that a node being offered a reward of r can only offer an amount $r' < r$ to its children, we do not enjoy this property on the ρ's in the case of split contracts. This unfortunately precludes the inductive arguments adopted in [KR05, AKK$^+$07], making a more involved analysis necessary.

We conclude this section discussing about the gap in efficiency between split contracts, for which an investment proportional to the depth of the search tree suffices for any branching factor $b > 1$, and the results in [KR05, AKK$^+$07], for which the investment becomes exponential in the depth of the search tree when the branching factor drops below 2. In the setting of [KR05], the additional amount of reward δ_j that the root needs in order to explore j levels of the tree (rather than stopping at level $j-1$) can be expressed as

$$\delta_{j+1} = \frac{1-\phi_{j-1}}{\lambda_j}\delta_j + 1.$$

When the branching factor drops below 2, the ratio $\frac{1-\phi_{j-1}}{\lambda_j}$ is greater than 1, and the investment needed at the root to propagate the query down to depth $h_\Psi(\epsilon, n)$ becomes exponential in $\log n$ (hence, $\mathrm{poly}(n)$).

In contrast, the dependency on the ratio $\frac{1-\phi_{j-1}}{\lambda_j}$ is softer in our setting. In the proof of Theorem 39, we show that the ρ-split a node at level ℓ needs to receive in order to propagate the query i levels down its subtree is

$$\rho_i^{\langle \ell \rangle} = 1 - \frac{1}{r^* - i(1 + O(\frac{1-\phi_{i-1}}{\lambda_i}))}.$$

Since we can show that $\frac{1-\phi_{i-1}}{\lambda_i}$ is bounded by a constant for any branching process with $b > 1$, an investment $r^* = O(h_\Psi(\epsilon, n)) = O(\log n)$ suffices for the value $\rho^{\langle 1 \rangle}_{h_\Psi(\epsilon,n)}$ offered by the root to its children to be well-defined (i.e., in $\mathcal{D}_M$), and hence for the answer to be retrieved cheaply.

4.4.4 Roadmap

The rest of the chapter is structured as follows. In Section 4.5, we derive properties that hold for any Nash equilibrium. In Section 4.6, we develop a condition that we call h-consistency under which we can show that a set of strategies

$\mathbf{g}$ for the nodes propagates the query to the desired level and is substantially the unique Nash equilibrium. In Section 4.7, we derive a bound on the investment r^*, depending on quantities related to the branching process, for which h-consistency is guaranteed to hold. Finally, in Section 4.8, we study such quantities of the branching process to conclude that $r^* = O(h_\Psi(\epsilon, n)) = O(\log n)$.

4.5 Properties of Nash equilibria

In this section we present the notion of Nash equilibrium that naturally arises in the context of split-contracts, and we then derive a manageable expression that any Nash equilibrium has to maximize. Let f_v be the function representing the strategy of node v, and $\mathbf{f}$ be the set of strategies of all nodes up to level $h_\Psi(\epsilon, n)$, as we assumed that nodes in lower levels do not play.

Definition 28 (**Nash equilibrium**). *Let r^*, Ψ, ϵ, n be the parameters of the model, and $\mathbf{f}$ be a set of functions for all nodes up to level $h_\Psi(\epsilon, n)$. For any such node v, let ρ^v be the split contract offered to v by its parent under $\mathbf{f}$. Then, $\mathbf{f}$ is a Nash equilibrium if, for each node v, v does not increase its expected payoff by deviating from $f_v(\rho^v)$ when all other nodes play according to $\mathbf{f}$. The expectation is taken over the randomness of the branching process and of the process assigning answers to the nodes.*

We now give a few definitions that will be useful to derive properties of any Nash equilibrium. Given a realization of the branching process, we say that a node v at level $\ell \leq h_\Psi(\epsilon, n)$ is *active* if the branching process reaches v. Moreover, given a set $\mathbf{f}$ of strategies and a realization of the branching process, we say that an active node v is $\mathbf{f}$-*reachable* if $\mathbf{f}$ forwards the query down to v. Given a realization of the branching process and of the process assigning the answer to nodes, we say that an $\mathbf{f}$-reachable node v at level ℓ is an $\mathbf{f}$-*candidate* if v holds the answer and no $\mathbf{f}$-reachable node at a level $\ell' < \ell$ does. Observe that the root selects for payment one among the $\mathbf{f}$-candidates. For each node v at level $\ell \leq h_\Psi(\epsilon, n)$, set $\mathbf{f}$ of strategies, and $j \geq 1$, let $\alpha_v^{\mathbf{f}}(j|\rho)$ be the probability that there is an $\mathbf{f}$-candidate in v's subtree at distance j from v, conditional on v being $\mathbf{f}$-reachable and offering a ρ-split to its children. Similarly, for $j \geq 1$, let $\beta_v^{\mathbf{f}}(j|\rho)$ be the expected payment that v receives from its children given that v offers a ρ-split to its children and there is an $\mathbf{f}$-candidate in v's subtree at distance j from v to whom the root propagates the payment.

The following lemma characterizes an expression that must be maximized by every node up to level $h_\Psi(\epsilon, n)$ in any Nash Equilibrium.

Lemma 29. *Consider any set* $\mathbf{f}$ *of strategies, and let* ρ^v *be the split contract offered to* v *by its parent under* $\mathbf{f}$*. Then,* $\mathbf{f}$ *is a Nash equilibrium if and only if, for every node* v *up to level* $h_\Psi(\epsilon, n)$*,* $f_v(\rho^v)$ *is a value of* ρ *maximizing the function*

$$\chi_v^{\mathbf{f}}(\rho; \rho^v) := \sum_{j \geq 1} \alpha_v^{\mathbf{f}}(j|\rho) \left((1 - \rho^v)\beta_v^{\mathbf{f}}(j|\rho) - 1 \right). \tag{4.1}$$

Proof. Fix the available investment r^*, a set $\mathbf{f}$ of strategies, and a node v at level $\ell \leq h_\Psi(\epsilon, n)$. We need to define the following events. Let A denote the event that the root propagates the payment down through v, that is, the root selects for payment an $\mathbf{f}$-candidate in v's subtree. For each $0 \leq j \leq h_\Psi(\epsilon, n) - \ell$, let B_j denote the event that the $\mathbf{f}$-candidates are at level $\ell + j$. Finally let C denote the event that v is $\mathbf{f}$-reachable and D denote the event that there is an $\mathbf{f}$-candidate in v's subtree. Observe that the co-occurrence of B_0 and D means that v itself is an $\mathbf{f}$-candidate. Given r^* and $\mathbf{f}$, let $Y_{\mathbf{f},r^*}^v$ be the random variable denoting the payment assigned to v.

We have that

$$\begin{aligned} E[Y_{\mathbf{f},r^*}^v] &= \sum_{j \geq 0} E[Y_{\mathbf{f},r^*}|A, B_j, D, C] \Pr(A, B_j, D, C) \\ &= \Pr(A|D) \Pr(C) \sum_{j \geq 0} E[Y_{\mathbf{f},r^*}|A, B_j, D, C] \Pr(B_j, D|C). \end{aligned}$$

The first equality follows from the law of total probability together with the observation that $\Pr(A, \overline{C}) = \Pr(A, \overline{D}) = 0$ and $E[Y_{\mathbf{f},r^*}^v|\overline{A}] = 0$. The second equality follows from the chain rule of probability and the fact that $\Pr(A|B_j, D, C) = \Pr(A|D)$ for all $j \geq 0$. Observe that the term corresponding to $j = 0$ (i.e., v is the $\mathbf{f}$-candidate selected for payment) does not depend on f_v since $E[Y_{\mathbf{f},r^*}|A, B_0, D, C] = (1 - \rho^v)r^* - 1$ and $\Pr(B_0, D|C)$ only depends on the strategies of the nodes that are ancestors of v. Similarly, f_v affects neither $\Pr(C)$, which depends on the strategies of v's ancestors only, nor $\Pr(A|D)$, which is only based on the root's choice of whom to propagate the payment to. Finally, note that if v offers a ρ-split to its children, then, for $j \geq 1$, $E[Y_{\mathbf{f},r^*}|A, B_j, S, C] = (1 - \rho^v)\beta_v^{\mathbf{f}}(j|\rho) - 1$ and $\Pr(B_j, D|C) = \alpha_v^{\mathbf{f}}(j|\rho)$. Therefore, $\mathbf{f}$ is a Nash equilibrium if and only if, for every node v up to level $h_\Psi(\epsilon, n)$, $f_v(\rho^v)$ is a value ρ maximizing $\chi_v^{\mathbf{f}}(\rho; \rho^v)$. □

To break ties in case of multiple maxima for $\chi_v^{\mathbf{f}}(\cdot; \rho^v)$, we make the same assumption as in [KR05, AKK$^+$07] that nodes favor strategies that forward the query further down in the tree. Observe that every node can efficiently compute the strategy that maximizes (4.1) given the strategies of the other nodes. The following two lemmas will lead to a simpler expressions for (4.1). We will start showing that Nash equilibria are "leveled" (proof in the Appendix).

Lemma 30. *Consider any Nash equilibrium* $\mathbf{f}$*. Then for each active node* v *at level* ℓ*,* v *is* $\mathbf{f}$*-reachable if and only if every active node at level* ℓ *is.*

Proof. Let $\mathbf{f}$ be a Nash equilibrium. Fix a node v and let

$$\rho_2 = \max\{\rho \in \mathcal{D}_M : \chi_v^{\mathbf{f}}(\rho_1; \rho) \geq \chi_v^{\mathbf{f}}(\perp; \rho)\}$$

be the maximum split v's father can ask v so that v will in turn prefer to offer a ρ_1-split to their children rather than just participating in the game without propagating the query. We first argue that ρ_2 does not depend on the chosen node v, then show that $f_v(\rho) = \perp$ for every node v and $\rho_2 < \rho \leq \rho_1 = 1 - \frac{1}{r^*}$, and finally use this fact to prove the lemma.

To see that ρ_2 does not depend on v, observe that $\chi_v^{\mathbf{f}}(\perp; \rho) = 0$ as $\alpha_v^{\mathbf{f}}(j|\perp) = 0$ for $j \geq 1$, and that

$$\begin{aligned}\chi_v^{\mathbf{f}}(\rho_1; \rho) &= \alpha_v^{\mathbf{f}}(1|\rho_1)((1 - \rho)\beta_v^{\mathbf{f}}(1|\rho_1) - 1)\\ &= \alpha_v^{\mathbf{f}}(1|\rho_1)((1 - \rho)r^*\rho_1 - 1).\end{aligned}$$

We now show that $f_v(\rho) = \perp$, for every node v and $\rho_2 < \rho \leq \rho_1$. By contradiction, suppose $f_v(\rho) = \rho'$, for some v, $\rho_2 < \rho \leq \rho_1$, $\rho' \in \mathcal{D}_M$. On the one hand, as $\mathbf{f}$ is a Nash equilibrium, Lemma 29 implies that ρ' maximizes $\chi_v^{\mathbf{f}}(\rho'; \rho)$, and thus

$$\chi_v^{\mathbf{f}}(\rho'; \rho) \geq \chi_v^{\mathbf{f}}(\perp; \rho) = 0.$$

On the other hand, we have that

$$\begin{aligned}\chi_v^{\mathbf{f}}(\rho';\rho) &= \sum_{j\geq 1}\alpha_v^{\mathbf{f}}(j|\rho')\left((1-\rho)\beta_v^{\mathbf{f}}(j|\rho')-1\right)\\ &\leq \sum_{j\geq 1}\alpha_v^{\mathbf{f}}(j|\rho')((1-\rho)r^*\rho_1-1),\end{aligned}$$

where the last inequality follows from $\beta_v^{\mathbf{f}}(j|\rho') \leq r^*\rho_1$ for all $j \geq 1$, as ρ' must be at most ρ_1 for v's children to participate to the game. By definition of ρ_2, it must be that $\chi_v^{\mathbf{f}}(\rho_1;\rho) < \chi_v^{\mathbf{f}}(\perp;\rho) = 0$, which implies $((1-\rho)r^*\rho_1 - 1) < 0$ and thus $\chi_v^{\mathbf{f}}(\rho';\rho) < 0$, generating a contradiction.

We are now ready to prove the lemma. By contradiction, suppose the statement of the lemma does not hold. Then there must be two sibling nodes u and v (at some level $\ell < h_\Psi(\epsilon,n)$) and a value $\rho = \rho^u = \rho^v$ such that $f_u(\rho) = \rho' \neq \perp$ and $f_v(\rho) = \perp$, that is, such that u forwards the query when offered a ρ-split by its parent while v does not. By the claim above, $f_u(\rho) = \rho'$ implies that $\rho \leq \rho_2$ and therefore, by definition of ρ_2, v would have incentive to deviate from f_v, offering a ρ_1-split to their children than just participating to the game without propagating the query, contradicting that $\mathbf{f}$ is a Nash equilibrium. □

By means of Lemma 30, we will say that a Nash equilibrium $\mathbf{f}$ is *k-tall* if level k is $\mathbf{f}$-reachable and level $k+1$ is not. This notion is useful in decoupling the probabilities $\alpha_v^{\mathbf{f}}(j|f_v(\rho))$ from the particular equilibrium $\mathbf{f}$ and node v. To see how, assume $\mathbf{f}$ is k-tall, $k \leq h_\Psi(\epsilon,n)$. For any node v at level $\ell \leq k$ and any $j \leq k-\ell$, let $\gamma_j^{\langle\ell\rangle}$ be the probability that there exists an $\mathbf{f}$-candidate in v's subtree at distance j from v (and therefore there is no $\mathbf{f}$-candidate in the first $\ell+j-1$ levels). Then, as $\mathbf{f}$ is k-tall, we have that for any node at level ℓ, $\gamma_j^{\langle\ell\rangle}$ depends only on ℓ and j (and not on $\mathbf{f}$ or the specific node). This observation directly yields the following result relating the probabilities $\alpha_v^{\mathbf{f}}(j|f_v(\rho^v))$ and $\gamma_j^{\langle\ell\rangle}$.

Lemma 31. *Let $\mathbf{f}$ be a k-tall Nash equilibrium, $k \leq h_\Psi(\epsilon,n)$. Then, for every $\ell \leq k$ and node v at level ℓ,*

$$\alpha_v^{\mathbf{f}}(j|f_v(\rho^v)) = \begin{cases} \gamma_j^{\langle\ell\rangle}, & \text{for } 1 \leq j \leq k-\ell \\ 0, & \text{for } j > k-\ell \end{cases}$$

In general, if the query is forwarded j levels down v's subtree when v offers a ρ-split to its children, then we have

$$\alpha_v^{\mathbf{f}}(j|\rho) = \gamma_j^{\langle \ell \rangle}.$$

Lemma 31 implies that, at equilibrium, the payoff of a node v deviating from its strategy depends only on the strategies of the nodes in the subtree rooted at v (needed for the computation of the terms $\beta_v^{\mathbf{f}}(j|\rho)$ in $\chi_v^{\mathbf{f}}(\rho;\rho^v)$).

4.6 The Nash equilibrium

In this section, we derive conditions for the existence of a Nash equilibrium that forwards the query down to level $h_\Psi(\epsilon, n)$, or, equivalently, retrieves the answer with the desired probability $1-\zeta-\epsilon$. For ease of notation, let $h = h_\Psi(\epsilon, n)$. We proceed as follows. First we define the functions $e_i^{\langle \ell \rangle}$ and the thresholds $\rho_i^{\langle \ell \rangle}$, which intuitively represent expected rewards and contracts for a special set of strategies $\mathbf{g}$. However, to define $\mathbf{g}$, we will need all $\rho_i^{\langle \ell \rangle}$ to exist and be decreasing in i, for all $\ell \le h$, property that we will dub h-consistency. Finally, assuming h-consistency, we will show that $\mathbf{g}$ forwards the query to level h and is a Nash equilibrium (in fact with an extra property, we will say $\mathbf{g}$ is a best-interest Nash Equilibrium).

We begin by defining the aforementioned functions and values. We provide an inductive process which defines, for each $1 \le \ell \le h$, a sequence of functions $e_i^{\langle \ell \rangle} : [0,1] \to \mathbb{R}$, $0 \le i \le h-\ell$, and values $\rho_i^{\langle \ell \rangle} \in \mathcal{D}_M$, $1 \le i \le h-\ell+1$. For every $0 \le \ell \le h$, set $e_0^{\langle \ell \rangle}(\rho) = 0$ and $\rho_1^{\langle \ell \rangle} = \rho_1 = 1 - 1/r^*$. Suppose that all $\rho_i^{\langle \ell' \rangle}$ have been defined for $\ell < \ell' \le h$ and $1 \le i \le h - \ell' + 1$. Then, for all $1 \le i \le h-\ell$, the function $e_i^{\langle \ell \rangle}(\rho)$ is defined as

$$e_i^{\langle \ell \rangle}(\rho) = \sum_{j=1}^{i} \gamma_j^{\langle \ell \rangle} \left[(1-\rho) r^* \left(\prod_{t=0}^{j-1} \rho_{i-t}^{\langle \ell+t+1 \rangle} \right) - 1 \right].$$

Having defined $e_i^{\langle \ell \rangle}(\rho)$, we define

$$\rho_{i+1}^{\langle \ell \rangle} = \max\{\rho \in \mathcal{D}_M : e_i^{\langle \ell \rangle}(\rho) \ge e_{i-1}^{\langle \ell \rangle}(\rho)\},$$

if such value exists, and leave $\rho_{i+1}^{\langle \ell \rangle}$ undefined otherwise.

For a node v at level $\ell \leq h$, $e_i^{\langle \ell \rangle}$ has the intuitive meaning of the expected reward that v receives from its children when the query is propagated i levels down v's subtree (assuming the other nodes play accordingly). The value $\rho_{i+1}^{\langle \ell \rangle}$ represents the "cheapest" split to offer a node v at level $\ell \leq h$ so that v prefers to propagate the query i levels down its subtree rather than $i - 1$ (recall that, to break ties, we assumed that nodes prefer to propagate the query further down the tree). To guarantee the propagation of the query to level h, we will need the values $\rho_{h-\ell}^{\langle \ell \rangle}$ to be defined.

Definition 32 (**h-consistency**). *We say that h-*consistency *holds if, for all $1 \leq \ell \leq h$ and $2 \leq i \leq h - \ell + 1$, the value $\rho_i^{\langle \ell \rangle}$ is defined and $\rho_i^{\langle \ell \rangle} < \rho_{i-1}^{\langle \ell \rangle}$ (note that $\rho_1^{\langle \ell \rangle}$ is always defined).*

Intuitively, the ordering of the values $\rho_i^{\langle \ell \rangle}$ in the definition of h-consistency states that if a node v propagates the query i levels down its subtree when offered a ρ-split by its father, then, in order to propagate the query $i + 1$ levels down, it must be that v is offered a split not greater than ρ. This property is at the basis of the following definition of the set of strategies $\mathbf{g}$, which we will then show to be a Nash equilibrium. Note how, under $\mathbf{g}$, nodes at the same level play the same strategy.

Definition 33 (**Strategy $\mathbf{g}$**). *Assume h-consistency. For each $1 \leq \ell \leq h$, consider the function $t^{\langle \ell \rangle}(\rho) : [0,1] \rightarrow \mathcal{D}_M \cup \{\perp\}$ defined by $t^{\langle \ell \rangle}(\rho) = \rho_{i-1}^{\langle \ell+1 \rangle}$ for the unique i such that $\rho_{i+1}^{\langle \ell \rangle} < \rho \leq \rho_i^{\langle \ell \rangle}$ (such i exists under h-consistency), where we assume $\rho_{h-\ell+2}^{\langle \ell \rangle} = 0$ and $\rho_0^{\langle \ell+1 \rangle} = \perp$. The set of strategies $\mathbf{g}$ is defined by setting $g_v(\rho) = t^{\langle \ell \rangle}(\rho)$ to every node v at level ℓ, for each $1 \leq \ell \leq h$, and letting the root play $\rho_h^{\langle 1 \rangle}$.*

It follows that, under $\mathbf{g}$, the root (at level zero) offers a $\rho_h^{\langle 1 \rangle}$-split to its children, who in turn offer $\rho_{h-1}^{\langle 2 \rangle}$-split contracts to their own children, and so on, until the nodes at level h, who do not forward the query (they play $t^{\langle h \rangle}(\rho_1^{\langle h \rangle}) = \perp$). Observe that $\mathbf{g}$ is h-tall, as all nodes up to level h are $\mathbf{g}$-reachable. The following theorem states that $\mathbf{g}$ is a Nash equilibrium.

Theorem 34 (**Nash equilibrium**). *Assume that h-consistency holds. Then the set of strategies* $\mathbf{g}$ *is a Nash equilibrium.*

The key fact in the proof is to show that, for any node v at level $1 \leq \ell \leq h$ (which under $\mathbf{g}$ receives a $\rho^{\langle \ell \rangle}_{h-\ell+1}$-split from its parent and in turn offers a $\rho^{\langle \ell+1 \rangle}_{h-\ell}$-split to its children), $\chi^{\mathbf{g}}_v(\rho^{\langle \ell+1 \rangle}_j; \rho) = e^{\langle \ell \rangle}_j(\rho)$ for all $j \leq h - \ell$, and that $\rho^{\langle \ell+1 \rangle}_{h-\ell}$ is the only maximizer of (4.1) that propagates the query to level h. Then the theorem follows by Lemma 29.

In the proof of the theorem, we make use of the following claim, that is a consequence of h-consistency.

Claim 35. *Assume h-consistency. Then, for every $1 \leq \ell \leq h$, $1 \leq i \leq h - \ell$, and $\rho^{\langle \ell \rangle}_{i+2} < \rho \leq \rho^{\langle \ell \rangle}_{i+1}$, we have that*

$$e^{\langle \ell \rangle}_i(\rho) > e^{\langle \ell \rangle}_{i+1}(\rho) > \cdots > e^{\langle \ell \rangle}_{h-\ell}(\rho),$$

and

$$e^{\langle \ell \rangle}_i(\rho) \geq e^{\langle \ell \rangle}_{i-1}(\rho) \geq \cdots \geq e^{\langle \ell \rangle}_0(\rho),$$

where we assume $\rho^{\langle 1 \rangle}_{h+1} = 0$.

Proof. Consider any $i + 1 \leq j \leq h - \ell$, and observe that, by definition,

$$\rho^{\langle \ell \rangle}_{j+1} = \max\{\rho' \in \mathcal{D}_M : e^{\langle \ell \rangle}_j(\rho') \geq e^{\langle \ell \rangle}_{j-1}(\rho')\},$$

and, by h-consistency (as $j + 1 \geq i + 2$), $\rho^{\langle \ell \rangle}_{j+1} \leq \rho^{\langle \ell \rangle}_{i+2} < \rho$. It follows that $e^{\langle \ell \rangle}_j(\rho) < e^{\langle \ell \rangle}_{j-1}(\rho)$ for all $i + 1 \leq j \leq h - \ell$, which implies that

$$e^{\langle \ell \rangle}_{h-\ell}(\rho) < e^{\langle \ell \rangle}_{h-\ell-1}(\rho) < \cdots < e^{\langle \ell \rangle}_i(\rho),$$

proving the first chain of inequalities in the lemma. Now consider any $2 \leq m \leq i + 1$, and observe that, by definition of $\rho^{\langle \ell \rangle}_m$,

$$e^{\langle \ell \rangle}_{m-1}(\rho^{\langle \ell \rangle}_m) \geq e^{\langle \ell \rangle}_{m-2}(\rho^{\langle \ell \rangle}_m)$$

and, by h-consistency (as $i+1 \geq m$), $\rho \leq \rho^{\langle \ell \rangle}_m$. This implies that $e^{\langle \ell \rangle}_{m-1}(\rho) \geq e^{\langle \ell \rangle}_{m-2}(\rho)$ for all $2 \leq m \leq i + 1$. It follows that

$$e^{\langle \ell \rangle}_i(\rho) \geq e^{\langle \ell \rangle}_{i-1}(\rho) \ldots \geq e^{\langle \ell \rangle}_0(\rho),$$

which proves the second chain of inequalities in the lemma. □

We are now ready to prove Theorem 34.

Proof of Theorem 34. Under h-consistency, for all $\ell \leq h$ and $2 \leq i \leq h-\ell+1$, $\rho_i^{\langle\ell\rangle}$ is defined and $\rho_i^{\langle\ell\rangle} < \rho_{i-1}^{\langle\ell\rangle}$ (recall that $\rho_1^{\langle\ell\rangle}$ is defined for all $\ell \leq h$).

To show that $\mathbf{g}$ is a Nash equilibrium, by Lemma 29, it suffices to prove that, for every node v at level up to h, $g_v(\rho^v)$ is the value that maximizes $\chi_v^{\mathbf{g}}(\cdot;\rho^v)$, where ρ^v is the split offer v receives from its parent. Let $0 \leq i \leq h-1$, and fix a node v at level $\ell = h-i$. Under $\mathbf{g}$, v receives a $\rho_{i+1}^{\langle\ell\rangle}$-split from its parent and in turn offers a $t^{\langle\ell\rangle}(\rho_{i+1}^{\langle\ell\rangle}) = \rho_i^{\langle\ell+1\rangle}$-split to its children. Therefore, it suffices to show that

$$\rho_i^{\langle\ell+1\rangle} = \arg\max_{\rho'}\{\chi_v^{\mathbf{g}}(\rho';\rho_{i+1}^{\langle\ell\rangle})\}.$$

We will in fact prove something stronger, that is, for all $\rho \in \mathcal{D}_M$,

$$g_v(\rho) = t^{\langle\ell\rangle}(\rho) = \arg\max_{\rho'}\{\chi_v^{\mathbf{g}}(\rho';\rho)\}. \tag{4.2}$$

Fix any $\rho \in \mathcal{D}_M$. A few observations allow to prove condition (4.2) for the chosen ρ. First, by h-consistency, there exists unique k such that $\rho_{k+2}^{\langle\ell\rangle} < \rho \leq \rho_{k+1}^{\langle\ell\rangle}$, where we assume $\rho_{h+1}^{\langle\ell\rangle} = 0$. Second, by definition of $\mathbf{g}$ and $\chi_v^{\mathbf{g}}(\cdot;\cdot)$, node v has an incentive to play a given $\rho' \in \mathcal{D}_M$ only if there is no $\hat{\rho} > \rho'$ such that v's children would play exactly the same split contract if either offered a $\hat{\rho}$-split or a ρ'-split (otherwise, the query would propagate the same number of levels down the tree, but with v earning more if offering a $\hat{\rho}$-split to its children). This implies that if ρ' maximizes $\chi_v^{\mathbf{g}}(.,\rho)$, then $\rho' = \rho_j^{\langle\ell+1\rangle}$ for some $0 \leq j \leq i$ (recall that node v is at level $h-i$). Third, by definition of $\mathbf{g}$, $e_j^{\langle\ell\rangle}(\cdot)$ and $\chi_v^{\mathbf{g}}(\cdot;\cdot)$, and by Claim 31, we have that $\chi_v^{\mathbf{g}}(\rho_j^{\langle\ell+1\rangle};\rho) = e_j^{\langle\ell\rangle}(\rho)$ for all $0 \leq j \leq h-\ell$. Finally, by Claim 35, as $\rho_{k+2}^{\langle\ell\rangle} < \rho \leq \rho_{k+1}^{\langle\ell\rangle}$, we have that $e_k^{\langle\ell\rangle}(\rho) > e_j^{\langle\ell\rangle}(\rho)$ for all $k < j < h-\ell$ and $e_k^{\langle\ell\rangle}(\rho) \geq e_j^{\langle\ell\rangle}(\rho)$ for all $0 \leq j < k$. We have that $\rho_k^{\langle\ell+1\rangle}$ is the only maximizers of $\chi_v^{\mathbf{g}}(\cdot;\rho)$ which forwards the query to level h, while any other maximizer forwards the query to some level $\ell' < h$. Therefore, as we assumed that nodes break ties preferring to propagate the query further down the tree, $\rho_k^{\langle\ell+1\rangle} = t^{\langle\ell\rangle}(\rho)$ is the preferred strategy of node v when offered a ρ-split from its parent. Considering all $\rho \in \mathcal{D}_M$, condition (4.2) follows and the theorem is proven. □

Even though $\mathbf{g}$ is not the only Nash equilibrium, the proof of Theorem 34 shows that $\mathbf{g}$ enjoys the additional property that, for each node v and $\rho \in \mathcal{D}_M$,

$$g_v(\rho) = \arg\max_{\rho'}\{\chi_v^{\mathbf{g}}(\rho'; \rho)\}. \tag{4.3}$$

We call any equilibrium enjoying such property a *best-interest* equilibrium, as nodes choose their best option in any scenario. The following theorem shows that $\mathbf{g}$ is substantially the only best-interest equilibrium, meaning that every other best-interest equilibrium $\mathbf{f}$ coincides with $\mathbf{g}$ on all the split-offers that are offered to nodes under $\mathbf{f}$. As a remark, we observe that even the game with fixed-payment contracts in [KR05, AKK$^+$07] admits multiple equilibria, although the authors claim uniqueness (a counter-example is presented in Appendix 4.9). On the positive side, the equilibrium analyzed in [KR05, AKK$^+$07] is the unique best-interest Nash equilibrium of their game.

Theorem 36 (**Uniqueness**). *Assume h-consistency. Let $\mathbf{f}$ be any ℓ-tall best-interest Nash equilibrium, for some $1 \leq \ell \leq h$, and, for each node v up to level ℓ, let ρ^v be the split contract offered to v by its parent under $\mathbf{f}$. Then, for every node v up to level ℓ, $f_v(\rho) = g_v(\rho)$, for all $\rho^v \leq \rho \leq \rho_1$.*

Proof. Under h-consistency, for all $\ell \leq h$, $\rho_i^{\langle \ell \rangle}$ is defined and $\rho_i^{\langle \ell \rangle} < \rho_{i-1}^{\langle \ell \rangle}$ for all $2 \leq i \leq h - \ell + 1$. Ler $\mathbf{f}$ be a best-interest Nash equilibrium that is ℓ-tall for some $\ell \leq h$. As $\mathbf{f}$ is best-interest, for every node v up to level ℓ,

$$f_v(\rho) = \arg\max_{\rho'}\{\chi_v^{\mathbf{f}}(\rho'; \rho)\}, \quad \forall \rho \in \mathcal{D}_M.$$

We want to prove that, for every node v up to level ℓ,

$$f_v(\rho) = g_v(\rho), \quad \forall \rho^v \leq \rho \leq \rho_1, \tag{4.4}$$

where ρ^v is the split offered to v by its parent under $\mathbf{f}$, $\rho_1 = 1 - 1/r^*$ and $\mathbf{g}$ is the best-interest Nash equilibrium from Definition 33.

We proceed by induction on the levels of the tree, starting from level ℓ and going backwards. In particular we prove by induction that (4.4) holds for every node at level ℓ, for every level $\ell' \leq \ell$. Consider any node v at level ℓ. As $\mathbf{f}$ is ℓ-tall (i.e., level ℓ is $\mathbf{f}$-reachable, while level $\ell+1$ is not), node v plays $\perp$. Therefore, v's parent (at level $\ell-1$) has incentive to offer v a ρ_1-split (the maximum split such that v has incentive to forward the answer to its parent). It follows that $\rho^v = \rho_1$ and $f_v(\rho_1) = \perp = g_v(\rho_1)$, and (4.4) holds for level ℓ.

Fix $0 \le i < \ell$, and suppose (4.4) holds for every node at level $\ell - i$. Let $\ell' = \ell - i - 1$, and consider any node v at level ℓ'. In the proof of Theorem 34, we showed that, for every $\rho \in \mathcal{D}_M$ and $1 \le j \le i$,

$$\chi_v^{\mathbf{g}}(\rho_j^{\langle \ell'+1\rangle}; \rho) = e_j^{\langle \ell' \rangle}(\rho).$$

By the inductive hypothesis on level $\ell'+1 = \ell - i$ and the fact that both $\mathbf{f}$ and $\mathbf{g}$ are best-interest, we have that, for every $\rho \in \mathcal{D}_M$ and $1 \le j \le i$,

$$\chi_v^{\mathbf{f}}(\rho_j^{\langle \ell'+1\rangle}; \rho) = \chi_v^{\mathbf{g}}(\rho_j^{\langle \ell'+1\rangle}; \rho).$$

The last two observations imply that, for every $\rho \in \mathcal{D}_M$ and $1 \le j \le i$,

$$\chi_v^{\mathbf{f}}(\rho_j^{\langle \ell'+1\rangle}; \rho) = e_j^{\langle \ell' \rangle}(\rho). \tag{4.5}$$

Lemma 35, togethet with (4.5), implies that

(i) for every $j < i$ and $\rho_{j+2}^{\langle \ell' \rangle} < \rho' \le \rho_{j+1}^{\langle \ell' \rangle}$, node v has incentive to play $\rho_j^{\langle \ell'+1\rangle}$ among all $\rho_i^{\langle \ell'+1\rangle} \le \rho \le \rho_1$, and

(ii) for $\rho' = \rho_{i+1}^{\langle \ell' \rangle}$, node v has incentive to play $\rho_i^{\langle \ell'+1\rangle}$ among all $\rho_i^{\langle \ell'+1\rangle} \le \rho \le \rho_1$.

We need the following technical result in order to proceed with the proof.

Claim 37. *Let v be a node at level $\ell' = \ell - i - 1$. Suppose that v receives a ρ'-split from its parent, with $\rho_{j+2}^{\langle \ell' \rangle} < \rho' \le \rho_{j+1}^{\langle \ell' \rangle}$ for some $j \le i$, and that v forwards the query exactly to level $\hat{\ell} \le \ell$. Moreover, assume that (4.4) holds for every node below v. Then, $\hat{\ell} = \ell' + j + 1$ and $f_v(\rho') = \rho_j^{\langle \ell'+1\rangle}$.*

Proof. Let $m = \hat{\ell} - \ell' - 1$. First we show that $f_v(\rho') \le \rho_m^{\langle \ell'+1\rangle}$, and then we argue that equality must hold. To show that $f_v(\rho') \le \rho_m^{\langle \ell'+1\rangle}$, suppose by contradiction that $f_v(\rho') > \rho_m^{\langle \ell'+1\rangle}$, that is, there exists $k < m$ such that $\rho_{k+1}^{\langle \ell'+1\rangle} < f_v(\rho') \le \rho_k^{\langle \ell'+1\rangle}$. Then, the query would only be forwarded to level $\ell' + 1 + j < \ell' + 1 + m = \hat{\ell}$, as we assumed that (4.4) holds for all nodes below v. This generates a contradiction, and, therefore, it must be $f_v(\rho') \leq \rho_m^{\langle \ell'+1 \rangle}$.

We now show that $f_v(\rho') = \rho_m^{\langle \ell'+1 \rangle}$. As $f_v(\rho') \leq \rho_m^{\langle \ell'+1 \rangle}$, we have that $\beta_v^{\mathbf{f}}(k|f_v(\rho')) \leq \beta_v^{\mathbf{f}}(k|\rho_m^{\langle \ell'+1 \rangle})$ for all $1 \leq k \leq m$, with equality if and only if $f_v(\rho') = \rho_m^{\langle \ell'+1 \rangle}$. This yields $\chi_v^{\mathbf{f}}(f_v(\rho'); \rho') < \chi_v^{\mathbf{f}}(\rho_m^{\langle \ell'+1 \rangle}; \rho')$, for $f_v(\rho') < \rho_m^{\langle \ell'+1 \rangle}$, which implies $f_v(\rho') = \rho_m^{\langle \ell'+1 \rangle}$. By (i), it must be $m = j$, which gives $\hat{\ell} = \ell' + m + 1 = \ell' + j + 1$. □

We now proceed with the proof. As $\mathbf{f}$ is ℓ-tall, $f_v(\rho^v)$ must forward the query exactly to level ℓ. We first show that $\rho^v = \rho_{i+1}^{\langle \ell' \rangle}$ and $f_v(\rho_{i+1}^{\langle \ell' \rangle}) = \rho_i^{\langle \ell'+1 \rangle}$, and then we show that (4.4) holds for v. Note that the claim above implies that if v receives a ρ'-split from its parent with $\rho' > \rho_{i+1}^{\langle \ell' \rangle}$, then $f_v(\rho')$ does not forward the query to level exactly ℓ. Therefore, it suffices to show that $f_v(\rho_{i+1}^{\langle \ell' \rangle}) = \rho_i^{\langle \ell'+1 \rangle}$. Indeed, this would imply that $\rho^v = \rho_{i+1}^{\langle \ell' \rangle}$, as no better (larger) split forwards the query to level exactly ℓ. By contradiction, suppose v plays $f_v(\rho_{i+1}^{\langle \ell' \rangle}) = \hat{\rho} \neq \rho_i^{\langle \ell'+1 \rangle}$. As we are assuming v is offered a $\rho_{i+1}^{\langle \ell' \rangle}$-split, and (ii) implies that v prefers to play $\rho_i^{\langle \ell'+1 \rangle}$ among all $\rho > \rho_i^{\langle \ell'+1 \rangle}$, it must be $\hat{\rho} < \rho_i^{\langle \ell'+1 \rangle}$. Moreover, it must be the case that $\hat{\rho}$ forwards the query below level ℓ, otherwise v would prefer to play $\rho_i^{\langle \ell'+1 \rangle}$. However, if it was the case, v would prefer to play $\hat{\rho}$ over $\rho_i^{\langle \ell'+1 \rangle}$ when offered any ρ'-split with $\rho' < \rho_{i+1}^{\langle \ell' \rangle}$. This contradicts the assumption that $\mathbf{f}$ is ℓ-tall, for which there exists $\rho' = \rho^v$ such that $f_v(\rho')$ forwards the query exactly to level ℓ.

We showed that $\rho_v = \rho_{i+1}^{\langle \ell' \rangle}$ and $f_v(\rho_v) = g_v(\rho_v)$. To complete the inductive step, we need to prove that $f_v(\rho') = g_v(\rho')$ for all $\rho^v \leq \rho' \leq \rho_1$. Fix any $\rho^v \leq \rho' \leq \rho_1$. We already proved that $f_v(\rho')$ does not forward the query exactly to level ℓ. Moreover, $f_v(\rho')$ cannot forward the the query below level $\ell' > \ell$, as otherwise v would prefer this strategy even when offered a ρ_v-split. Thus, $f_v(\rho')$ must forward the query to some level $\hat{\ell} < \ell$, and Claim 37 concludes the proof. □

Theorem 36 implies that every best-interest equilibrium $\mathbf{f}$ in which the root offers a $\rho_h^{\langle 1 \rangle}$-split to its children has to be h-tall, as $\mathbf{f}$ agrees with $\mathbf{g}$ on all split-offers made in $\mathbf{g}$. As h-tall equilibria retrieve the answer with the desired probability, the root has incentive to play $\rho_h^{\langle 1 \rangle}$ as its strategy and would have incentive to deviate to $\rho_h^{\langle 1 \rangle}$ if playing a different strategy. This observation implies the following Corollary.

Corollary 38. *Under the assumption of h-consistency, every best-interest equilibrium retrieve the answer with the desired probability.*

Furthermore, Theorem 36 shows that all best-interest equilibria are *identical* to $\mathbf{g}$ for all the values that matter, i.e. all the values of $\rho^v \leq \rho \leq \rho_1$. (The values $\rho < \rho^v$ are uninteresting as, when v is offered ρ^v, the query will already reach all nodes in v's subtree, hence v's parent has no incentive to offer v a value $\rho < \rho^v$.) This suggests that nodes can easily reach one such equilibrium. In particular, as shown in the proof of Theorem 36, nodes at level h will participate only when offered a ρ-split for $\rho \leq \rho_1$; nodes at level $h-1$ can infer this and, by playing according to (4.3), will offer a ρ_1-split to their children when offered a ρ'-split for $\rho' \leq \rho^{\langle 2 \rangle}_{h-1}$, and will only participate (without propagating the query) if offered a ρ'-split for $\rho^{\langle 2 \rangle}_{h-1} < \rho' \leq \rho_1$; the argument extends up to the root.

4.7 Guaranteeing h-consistency

Until now, we assumed h-consistency both in the definition of $\mathbf{g}$ and in the proof that $\mathbf{g}$ is a Nash equilibrium. It therefore remains to derive conditions that ensure h-consistency. In the following theorem, we provide a lower bound on the reward r^* above which h-consistency is guaranteed. The bound reads in terms of the probabilities $\gamma^{\langle \ell \rangle}_i$ through the quantities $\Gamma^{\langle \ell \rangle}_i = \frac{1}{\gamma^{\langle \ell \rangle}_i} \sum_{j=1}^{i-1} \gamma^{\langle \ell \rangle}_j$, which, for all $1 \leq \ell \leq h$ and $1 \leq i < h - \ell$, intuitively represent the ratio between the probability that a node at level ℓ has a candidate at depth $j < i$ in its subtree versus the probability that it has one at depth i.

Theorem 39. *Suppose the discretization parameter M is large enough, say $M = \Theta(r^{*2})$, and that*

$$r^* \geq 4 \cdot h \cdot \max\left\{1, \max_{\substack{1 \leq \ell \leq h \\ 1 \leq i < h - \ell}} \Gamma^{\langle \ell \rangle}_i \right\}. \tag{4.6}$$

Then h-consistency holds. In particular, for all $1 \leq \ell \leq h$ and $1 \leq i \leq h - \ell$, $\rho^{\langle \ell \rangle}_i$ is defined and satisfies

$$1 - \frac{1}{r^* - i} < \rho^{\langle \ell \rangle}_i \leq 1 - \frac{1}{r^* - i + 1}. \tag{4.7}$$

Proof. Suppose condition (4.6) holds. We show by induction that, if the discretization parameter M is large enough, for all $1 \leq \ell \leq h$ and $1 \leq i \leq h-\ell+1$, $\rho_i^{\langle \ell \rangle}$ is defined and satisfies (4.7), that is, h-consistency holds.

By definition we have $\rho_1^{\langle \ell \rangle} = \rho_1 = 1 - 1/r^*$, for all $1 \leq \ell \leq h$. Therefore (4.7) holds for all $1 \leq \ell \leq h$ and $i = 1$. Fix $\ell \leq h$ and suppose the claim holds for all $\ell \leq \ell' \leq h$ and $1 \leq i \leq h - \ell'$. We recall that $\rho_{i+1}^{\langle \ell-1 \rangle}$ is defined as

$$\rho_{i+1}^{\langle \ell-1 \rangle} = \max\{\rho \in \mathcal{D}_M : e_i^{\langle \ell-1 \rangle}(\rho) \geq e_{i-1}^{\langle \ell-1 \rangle}(\rho)\},$$

where

$$e_i^{\langle \ell-1 \rangle}(\rho) = \sum_{j=1}^{i} \gamma_j^{\langle \ell-1 \rangle} \left[(1-\rho) r^* \left(\prod_{t=0}^{j-1} \rho_{i-t}^{\langle (\ell-1)+t+1 \rangle} \right) - 1 \right].$$

By definition of $\rho_{i+1}^{\langle \ell-1 \rangle}$, it must be that

$$1 - \frac{1}{r^* \Delta_i} - \frac{1}{M} \leq \rho_{i+1}^{\langle \ell-1 \rangle} \leq 1 - \frac{1}{r^* \Delta_i},$$

where

$$\Delta_i = \prod_{j=0}^{i-1} \rho_{i-j}^{\langle \ell+j \rangle} - \sum_{j=1}^{i-1} \frac{\gamma_j^{\langle \ell \rangle}}{\gamma_i^{\langle \ell \rangle}} \left[\prod_{t=0}^{j-1} \rho_{i-t-1}^{\langle \ell+t \rangle} - \prod_{t=0}^{j-1} \rho_{i-t}^{\langle \ell+t \rangle} \right],$$

and M is the discretization parameter of the domain $\mathcal{D}_M$. To see this, compute the difference $e_i^{\langle \ell-1 \rangle}(\rho_{i+1}^{\langle \ell-1 \rangle}) - e_{i-1}^{\langle \ell-1 \rangle}(\rho_{i+1}^{\langle \ell-1 \rangle})$, and argue that $1 \leq (1-\rho_{i+1}^{\langle \ell-1 \rangle}) r^* \Delta_i \leq 1 + r^* \Delta_i / M$.

We find lower and upper bounds to the term between brackets in the expression for Δ_i. First, by the inductive hypothesis, $\rho_{i-t-1}^{\langle \ell+t \rangle} > \rho_{i-t}^{\langle \ell+t \rangle}$ for all $0 \leq t \leq h-\ell$ and $0 \leq t \leq j-1$ (with $j < i$). Therefore, we have

$$\prod_{t=0}^{j-1} \rho_{i-t-1}^{\langle \ell+t \rangle} - \prod_{t=0}^{j-1} \rho_{i-t}^{\langle \ell+t \rangle} > 0.$$

Also by induction, we have

$$\begin{aligned}
\prod_{t=0}^{j-1} \rho_{i-t-1}^{\langle \ell+t \rangle} - \prod_{t=0}^{j-1} \rho_{i-t}^{\langle \ell+t \rangle} &< \prod_{t=0}^{j-1} \frac{r^* - i + t + 1}{r^* - i + t + 2} - \prod_{t=0}^{j-1} \frac{r^* - i + t - 1}{r^* - i + t} \\
&= \frac{r^* - i + 1}{r^* - i + j + 1} - \frac{r^* - i - 1}{r^* - i + j - 1} \\
&= \frac{2j}{(r - i + j + 1)(r - i + j - 1)} \\
&< \frac{2i}{(r^*)^2},
\end{aligned}$$

as $j < i$ in the expression of Δ_i. Therefore, again by induction, we have

$$\frac{r^* - i - 1}{r^* - 1} - \frac{2i}{(r^*)^2}\Gamma_i^{\langle \ell \rangle} < \Delta_i < \frac{r^* - i}{r^*}.$$

The upper bound on $\rho_{i+1}^{\langle \ell-1 \rangle}$ follows immediately. For the lower bound, it suffices to show that $r^* \cdot \Delta_i > (r^* - i - 1)(1 + r^*/M)$. Also, note that this would imply that $\Delta_i > 0$, and so that $\rho_{i+1}^{\langle \ell-1 \rangle}$ is defined. Rearranging the terms, it suffices to show that

$$\frac{2i}{r^*(r^* - i - 1)}\Gamma_i^{\langle \ell \rangle} < \frac{1}{r^* - 1} - \frac{r^*}{M}. \tag{4.8}$$

By (4.6), we have that

$$i \leq \frac{r^*}{4} \min\{1, 1/\Gamma_i^{\langle \ell \rangle}\} - 1.$$

Then, (4.8) holds if

$$\frac{1/2}{1 - \frac{1}{4}\min\{1, 1/\Gamma_i^{\langle \ell \rangle}\}} < 1 - \frac{(r^*)^2}{M},$$

which is satisfied for M large enough. □

Recall that $\rho_{h-\ell+1}^{\langle \ell \rangle}$ is the split offered to the nodes at level ℓ in the Nash equilibrium $\mathbf{g}$. Theorem 39 along with Theorem 36 implies that, in every best-interest equilibrium, the nodes at level ℓ receive a split offer close to $1 - \frac{1}{r^* - (h-\ell)}$.

It can also be proven that, for a fixed i, $\rho_i^{\langle \ell \rangle}$ is decreasing in ℓ for $0 \leq \ell \leq h - i$. The intuition for this property is that a node further down in the tree is willing to give a smaller fraction of its reward back to its parent, in order to compensate the smaller probability of having a candidate in its subtree. However, we do not need this property to ensure h-consistency.

Theorem 39 along with Corollary 38 directly yields the following pivotal result, which relates the quantities $\Gamma_j^{\langle \ell \rangle}$ to the investment that is sufficient at the root to retrieve the answer with the desired probability.

Corollary 40. *Suppose condition* (4.6) *holds. Then, in any best-interest Nash equilibrium, the query reaches all nodes at level* $h = h_\Psi(\epsilon, n)$ *of the tree. That is, an answer is retrieved with probability at least* $1 - \zeta - \epsilon$.

4.8 Efficiency

In the previous section, we derived a lower bound on the investment r^* as a function of the values $\Gamma_i^{\langle \ell \rangle}$, for $1 \leq \ell \leq h$ and $1 \leq i \leq h - \ell$. In this section, we show our main result by relating these values to the branching process and the desired success probability. The following lemma bounds these quantities in terms of the probabilities λ_i and ϕ_i of the branching process. Recall that, for each $i \geq 0$ we defined ϕ_i as the probability that no node at level $j \leq i$ possesses the answer, and $\lambda_i = \phi_{i-1} - \phi_i$ as the probability that a node at level i possesses the answer and no node at level $j < i$ does.

Lemma 41. *For every $1 \leq \ell \leq h$ and $1 \leq i \leq h - \ell$, it holds that*

$$\Gamma_i^{\langle \ell \rangle} \leq \frac{1}{\phi_{\ell+i-1}} \frac{1 - \phi_{i-1}}{\lambda_i}.$$

The key in proving Lemma 41 is to express $\gamma_i^{\langle \ell \rangle}$ in terms of the probabilities ϕ_j and λ_j defined above, and then to bound $\Gamma_i^{\langle \ell \rangle}$ exploiting the memory-less property of the branching process and of the process assigning the answer to the nodes.

Proof. Recall that, for each $i \geq 0$ we defined ϕ_i as the probability that no node at level $j \leq i$ possesses the answer, and $\lambda_i = \phi_{i-1} - \phi_i$ as the probability that a node at level i possesses the answer while no node at level $j < i$ does. Also, for every $0 \leq \ell \leq h$ and $0 \leq i \leq \ell$, we defined

$$\Gamma_i^{\langle \ell \rangle} = \frac{\sum_{j=1}^{i-1} \gamma_j^{\langle \ell \rangle}}{\gamma_i^{\langle \ell \rangle}},$$

where $\gamma_i^{\langle \ell \rangle}$ is the probability that, fixed any node v at level ℓ, there is a $\mathbf{g}$-candidate u in v's subtree at distance i from v, given that v is active. We recall that a node u at level ℓ' is a $\mathbf{g}$-candidate if, under strategy $\mathbf{g}$, u is an active answer-holder and there is no active answer-holder in the first $\ell' - 1$ levels. Let L_j be the event that there is an answer holder at level j of the tree, and F_j be the event that no event L_k happens for all $k \leq j$. Observe that $\Pr(L_j, F_{j-1}) = \lambda_j$ and $\Pr(F_j) = \phi_j$. Fix a node v at level $\ell < h$. Let L_j^v be the event that there is an answer holder in

v's subtree at distance j from v, and F_j^v be the event that no L_k^v happens for all $k \leq j$. Also, let A_v be the event that v is active. We have

$$\begin{aligned}
\gamma_j^{\langle \ell \rangle} &= \Pr(L_j^v, F_{\ell+j-1} | A^v) \\
&= \Pr(L_j^v | A^v, F_{\ell+j-1}) \Pr(F_{\ell+j-1} | A^v) \\
&= \Pr(L_j | F_{j-1}) \Pr(F_{\ell+j-1} | A^v) \\
&= \frac{\Pr(L_j, F_{j-1})}{\Pr(F_{j-1})} \Pr(F_{\ell+j-1} | A^v),
\end{aligned}$$

where the third equality follows by the fact that the branching process is memoryless. By Bayes' rule,

$$\gamma_j^{\langle \ell \rangle} = \frac{\Pr(L_j, F_{j-1})}{\Pr(F_{j-1})} \frac{\Pr(A^v | F_{\ell+j-1}) \Pr(F_{\ell+j-1})}{\Pr(A^v)}.$$

Observe that the probability that v is active only depends on the existence of answer-holders on the path from the root to v or in the subtree rooted at v. Therefore, letting P^v be the event that there is no answer-holder in the path from the root to v, we can write

$$\begin{aligned}
\Pr(A^v | F_{\ell+j-1}) &= \Pr(A^v | P^v, F_{j-1}^v) \\
&= \frac{\Pr(F_{j-1}^v | A^v, P^v) \Pr(A^v | P^v)}{\Pr(F_{j-1}^v | P^v)} \\
&= \frac{\Pr(F_{j-1}) \Pr(A^v | P^v)}{\Pr(F_{j-1}^v | P^v)},
\end{aligned}$$

where the second equality follows by Bayes' rule, and the third equality by the memory-less property of the branching factor. It follows that, for all $\ell \leq h$ and $0 \leq j \leq h - \ell$,

$$\gamma_j^{\langle \ell \rangle} = \frac{\Pr(L_j, F_{j-1}) \Pr(F_{\ell+j-1}) \Pr(A^v | P^v)}{\Pr(F_{j-1}^v | P^v) \Pr(A^v)}.$$

Plugging the last expression into the definition of $\Gamma_i^{\langle \ell \rangle}$, we get

$$\begin{aligned}
\Gamma_i^{\langle \ell \rangle} &= \frac{1}{\Pr(L_i, F_{i-1}) \Pr(F_{\ell+i-1})} \sum_{j=1}^{i-1} \Pr(L_j, F_{j-1}) \Pr(F_{\ell+j-1}) \frac{\Pr(F_{i-1}^v | P^v)}{\Pr(F_{j-1}^v | P^v)} \\
&= \frac{1}{\lambda_i \phi_{\ell+i-1}} \sum_{j=1}^{i-1} \lambda_j \phi_{\ell+j-1} \frac{\Pr(F_{i-1}^v | P^v)}{\Pr(F_{j-1}^v | P^v)}.
\end{aligned}$$

As $\Pr(F^v_{i-1}|P^v) \leq \Pr(F^v_{j-1}|P^v)$ for $j \leq i$, and $\phi_{\ell+j-1} \leq 1$, we have that

$$\begin{aligned}\Gamma_i^{\langle \ell \rangle} &\leq \frac{1}{\lambda_i \phi_{\ell+i-1}} \sum_{j=1}^{i-1} \lambda_j \\ &< \frac{1}{\phi_{\ell+i-1}} \frac{1-\phi_{i-1}}{\lambda_i}.\end{aligned}$$

□

The following technical lemma provides an upper bound to $\frac{1-\phi_{i-1}}{\lambda_i}$. In particular, for any fixed branching process with $b > 1$, this ratio is bounded by a constant, as long as ϕ_i is bounded away from the extinction probability ζ. The lemma characterizes the bound with respect to the branching process and the gap $\phi_i - \zeta$, and its proof builds on the mathematical properties of the probability generating function of the branching process. Recall that the desired success probability is $1 - \zeta - \epsilon$.

Lemma 42. *Consider any Galton-Watson branching process with branching factor $b > 1$. Then, for every i such that $\zeta + \epsilon \leq \phi_i \leq 1$, it holds that*

$$\frac{1-\phi_i}{\lambda_{i+1}} \leq \max\left\{\frac{1}{b-1}, \frac{1}{\epsilon} \cdot \frac{1}{1-\Psi'(\zeta)}\right\}.$$

Proof. For all $i \geq 0$, let $\hat{\phi}_i = \phi_i / p$ be the probability that for all levels up to i no node has the answer given that the root (at level zero) does not. Observe that no node up to level $i+1$ has the answer given that the root does not if and only if the root's children and their subtrees up to depth i do not have the answer. Therefore, we have that $\hat{\phi}_{i+1} = \Psi(p \cdot \hat{\phi}_i)$, where $\Psi(x)$, $0 \leq x \leq 1$ is the probability generating function of the branching process. It follows that

$$\begin{aligned}\lambda_{i+1} &= \phi_i - \phi_{i+1} \\ &= \phi_i - p \cdot \hat{\phi}_{i+1} \\ &= \phi_i - p \cdot \sum_{k=0}^{d} c_k \hat{\phi}_i^k p^k \\ &= \phi_i - p \sum_{k=0}^{d} c_k \phi_i^k.\end{aligned}$$

Therefore we conclude

$$\lambda_{i+1} > \phi_i - \sum_{k=0}^{d} c_k \phi_i^k = \phi_i - \Psi(\phi_i). \tag{4.9}$$

For $0 < \epsilon \leq 1 - \zeta$ and $0 \leq z < 1 - \zeta$, let

$$a(\epsilon) = \max\left\{\frac{1}{b-1}, \frac{1}{\epsilon}\frac{1}{1-\Psi'(\zeta)}\right\},$$

and

$$t(z,\epsilon) = a(\epsilon) \cdot (1 - z - \Psi(1-z)) - z.$$

We need to show that, for any $0 < \epsilon \leq 1 - \zeta$,

$$\frac{1-\phi_i}{\lambda_{i+1}} \leq a(\epsilon).$$

Observe that, by inequality (4.9),

$$\frac{1-\phi_i}{\lambda_{i+1}} \leq \frac{1-\phi_i}{\phi_i - \Psi(\phi_i)}$$

and therefore it suffices to prove that, for every $\epsilon > 0$,

$$t(1-\phi_i, \epsilon) = a(\epsilon)\left(\phi_i - \Psi(\phi_i)\right) - (1-\phi_i) \geq 0.$$

First, observe that, for every $\epsilon > 0$, we have $t(0,\epsilon) = 0$, since $\Psi(1) = 1$ (see [AN04]). Also note that

$$\left.\frac{\partial}{\partial z} t(z,\epsilon)\right|_{z=0} = a(\epsilon) \cdot (\Psi'(1) - 1) - 1 = a(\epsilon) \cdot (b-1) - 1,$$

which is non-negative since $a(\epsilon) \geq 1/(b-1)$. Also, observe that $\frac{\partial^2}{\partial z^2} t(z,\epsilon) < 0$ and $\frac{\partial}{\partial \epsilon} t(z,\epsilon) > 0$, for all z and ϵ in their respective domains. Therefore, since the function $t(z,\epsilon)$ is continuous, it suffices to check that $\lim_{\epsilon \to 0} t(1-\zeta-\epsilon, \epsilon) \geq 0$. As $(1-b)^{-1} \leq \epsilon^{-1}(1-\Psi'(\zeta))^{-1}$ for ϵ small enough, we have that

$$\lim_{\epsilon \to 0} t(1-\zeta-\epsilon,\epsilon) > \lim_{\epsilon \to 0}\left[\frac{1}{1-\Psi'(\zeta)}\frac{1}{\epsilon}\left(\zeta + \epsilon - \Psi(\zeta+\epsilon)\right)\right] - 1.$$

Since $\zeta = \Psi(\zeta)$, by l'Hôpital's rule, $\lim_{\epsilon \to 0} t(1-\zeta-\epsilon,\epsilon) > 0$. □

Our main result directly follows by combining Corollary 40, Lemma 41 and Lemma 42, along with the observation that $\phi_{\ell+i-1} > \epsilon$ (as $\phi_{\ell+i-1} \geq \phi_{h_\Psi(\epsilon,n)-1} > \epsilon$). For the case of a ray, the bound can be obtained observing that $\phi_i = (1 - {}^1\!/_n)^i$ and $\lambda_{i+1} = \frac{\phi_i}{n}$, which implies $\Gamma_i^{\langle \ell \rangle} \leq \Gamma_h^{\langle 1 \rangle} \leq \epsilon^{-2} n$.

Theorem 43 (**Efficiency**)**.** *Consider any Galton-Watson branching process with $b > 1$. Then, at equilibrium, the root retrieves the answer with probability at least $\sigma = 1 - \zeta - \epsilon$ provided an investment of*

$$r^* = \frac{4}{\epsilon} \cdot \max\left\{\frac{1}{b-1}, \frac{1}{\epsilon} \cdot \frac{1}{1 - \Psi'(\zeta)}\right\} \cdot h_\Psi(\epsilon, n).$$

In the case of a ray, with $b = 1$ and $c_0 = \zeta = 0$, an investment of $r^* = 4 \cdot \frac{n}{\epsilon^2} \cdot h_\Psi(\epsilon, n) = 4 \cdot \frac{n^2}{\epsilon^2} \ln \frac{1}{\epsilon}$ suffices.

Observe that an investment of $h_\Psi(\epsilon, n)$ is necessary even in a centralized (non-strategic) setting, where the root decides the strategies of all nodes while only guaranteeing a non-negative payoff to them (each node pays a unit cost when returning the answer). In line with intuition, the investment grows as b tends to 1 (in the limit, when the tree becomes a ray, the investment is polynomial in n), and when the accuracy ϵ approaches 0. The term $\frac{1}{1-\Psi'(\zeta)}$ can be crudely bounded by $\frac{1}{c_0}$. However, when c_0 tends to zero, so does the extinction probability ζ, which implies $\frac{1}{1-\Psi'(\zeta)} \approx \frac{1}{1-c_1}$, also suggesting a more expensive investment when the tree tends to a ray (i.e., when c_1 approaches 1).

4.9 Discussion: non-uniqueness of the Nash equilibrium

In this section, we discuss the existence of multiple Nash equilibria both in the game with fixed-payment contract of [KR05, AKK$^+$07] and in the game with split contracts presented in this work.

First we recall the setting of [KR05, AKK$^+$07]. Each node has an *integer-valued* function f_v; if v is offered a reward of $r \geq 1$ by its parent, and v does not possess the answer to the query, then v offers in turn a reward of $f_v(r) < r$ to its children. Also, by definition, $f_v(1) = 0$. Kleinberg and Raghavan [KR05] show that a set of strategies $\mathbf{f}$ is a Nash equilibrium if and only if, for every node v, $f_v(r^v)$ is the value x maximizing the function

$$h_v(x; r^v) = (r^v - x - 1)p_v(\mathbf{f}, x).$$

Here r^v is the reward offered to v by its parent under $\mathbf{f}$, and $p_v(\mathbf{f}, x)$ is the probability that the subtree below v yields the answer, given that v does not possess the answer and offers reward x to its children. This characterization of the Nash equilibria for the game with fixed-payment contract is analogous to our result of Lemma 29 for split contracts, where the optimization is with respect to the function $\chi_v^{\mathbf{f}}(\cdot, \rho^v)$.

Using the functions $h_v(x; r^v)$, it is possible to construct a set of strategies $\mathbf{g}^{\text{fixed}}$ which optimizes $h_v(x; r^v)$ for every node v and is therefore a Nash equilibrium of the game with fixed-payment contracts. Theorem 2.2 in [KR05] claims that $\mathbf{g}^{\text{fixed}}$ is the unique equilibrium, in the sense that any other Nash equilibrium $\mathbf{f}$ in which $f_v(2) = 1$ is such that for all nodes v and rewards r that are reachable at v with respect to f, $f_v(r) = g_v^{\text{fixed}}(r)$. Note that this claim would imply that all equilibria have the same efficiency, in that the query is forwarded to the same levels in every equilibrium.

Unfortunately, this claim can be showed to hold true only when restricted to *best-interest* equilibria (as in our setting, see Theorem 36), that is, when considering only equilibria where $f_v(r')$ is the value x maximizing $h_v(x; r')$, for *every* r'. Note that in a *best-interest* equilibrium, nodes choose their strategies to optimize their payoff for any possible offer they *may* receive. This suggests that equilibria that are *not* best-interest are somewhat pathological, as contain nodes who do not consider their payoff globally. It is possible to show that both games admit (non-best-interest) equilibria that can be very inefficient in the sense that the query is only forwarded to a constant number of levels in the tree no matter how large the available investment r^* is. We present one of these equilibria for the case of fixed-payment contracts (the case with split contracts is similar). Consider the set of strategies $\mathbf{f}$ in which all nodes at level 1 play $f_1(r)$, all nodes at level 2 play $f_2(r)$, and all nodes below play $f_3(r)$ (recall that the root is at level zero). For a parameter $r' \geq 4$, the functions are defined as follows.

$$f_1(r) = \begin{cases} 0, & \text{if } r = 1 \\ 1, & \text{if } r = 2 \\ 2, & \text{if } r \geq 3 \text{ and } \frac{(r-r'-1)(\lambda_1+\lambda_2+\lambda_3)}{(r-2-1)(\lambda_1+\lambda_2)} < 1 \\ r', & \text{if } r \geq 3 \text{ and } \frac{(r-r'-1)(\lambda_1+\lambda_2+\lambda_3)}{(r-2-1)(\lambda_1+\lambda_2)} \geq 1 \end{cases}$$

$$f_2(r) = \begin{cases} 0, & \text{if } r = 1 \\ 1, & \text{if } 2 \leq r < r' \\ 2, & \text{if } r \geq r' \end{cases}$$

$$f_3(r) = \begin{cases} 0, & \text{if } r = 1 \\ 1, & \text{if } r \geq 2 \end{cases}$$

It can be verified that $\mathbf{f}$ is a Nash equilibrium, which thus forwards the query to level at most 3, regardless of the reward r^* offered by the root to the nodes at level 1. The bottleneck in the equilibrium is created by the nodes at level 3 or more, who cannot forward the query more than a single level as they never offer their children more than 1; in light of this, the nodes at level 2 are not going to offer their children more than 2 (and they do so when receiving at least r'), and in turn the nodes at level 1 do not offer more than r'. This causes the query not to be forwarded efficiently. This phenomenon cannot happen in a best-interest equilibrium as, roughly speaking, the nodes at level 3 (or more) would consider the scenario in which they get offered an amount larger than 2 and realize that it is more convenient to offer their children an amount larger than 1 (assuming the nodes below reason similarly), therefore forwarding the query deeper down the tree.

4.10 Simulations

In this section, we experimentally compare the amount of investment needed with fixed-payment and split contracts. Our results show that split contracts are more efficient than fixed-payment contracts not only for small branching factors ($1 < b < 2$), but also for large branching factors. Figures 4.1 and 4.2 show the investments, as a function of the rarity n, in the case of a binomial offspring

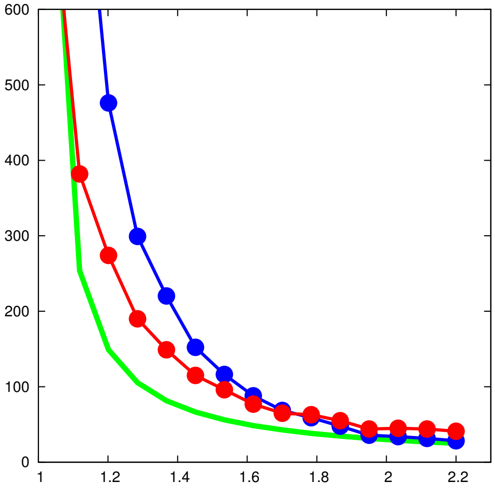

Figure 4.3: As a function of $b > 1$: investment with split-contracts (in red circles) and **logarithm** of the investment with fixed-payment contracts (in blue circles). In green, the function $\frac{1}{b-1}$ scaled by a constant factor.

distribution $\text{Bin}(5, q)$ with $q = 0.39$ (i.e. $b = 1.95$) and $q = 0.8$ (i.e., $b = 4$), with success accuracy $\epsilon \in \{0.2, 0.05\}$. Figure 4.1 shows that the improvement is major (exponential) already for b slightly less than 2, while Figure 4.2 shows a constant-factor gain for large b. Similar improvements were observed in general for any $b > 1$ and $\epsilon > 0$, with finer accuracy ϵ leading to more improvement. The dependence of the investment w.r.t. the accuracy ϵ (see Theorem 43) is more pronounced for small b (i.e. in Figure 4.1), but it is still noticeable even for larger b when employing smaller values of ϵ.

Also, we consider the investments as a function of b tending to 1. Figure 4.3 shows the case of a binomial branching process $\text{Bin}(5, q)$ with q tending to $1/5$, accuracy $\epsilon = 0.01$, and rarity $n = 10^{-6}$. As the investment with fixed-payment is exponentially larger than that with split contracts for small b, we plotted its logarithm. The figure confirms the $\frac{1}{b-1}$ factor from Theorem 43 in the case of split contracts (note that $\frac{1}{\epsilon(1-\Psi'(\zeta))}$ instead is bounded by a constant, as $c_0 = \Omega(1)$), and also suggests that the investment with fixed-payment contracts is exponential in ${}^{1}/_{(b-1)}$.

Chapter 4, in part, is a reprint of the paper "Finding red balloons with split contracts: robustness to individuals' selfishness" co-authored with Manuel Cebrián, Lorenzo Coviello, and Panagiotis Voulgaris published in the proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012 [CCVV12]. The dissertation author was the primary investigator and author of this paper.

Part II

Models for Other Phenomena

Chapter 5

Models for Aggregation

Aggregation of different entities manifests itself in several dynamic systems. Global population is one example: people are aggregated in a few dense urban areas rather than being distributed uniformly over the entire planet. Similar aggregation phenomena are observed in smaller scale as well: concentration of stores inside malls, abundance of restaurants around the center of a city, high density of students living near a university, and so on. Many more examples are encountered in the areas of economics, sociology, biology, and other fields.

Given the large body of evidence of aggregation phenomena, we wish to provide a theoretical model that explains it. The formalization we use in this chapter models a population of t individuals inhabiting a world – represented as an undirected graph[1] with n nodes– and measures aggregation by the number of edges induced by the nodes occupied by the individuals. Since all the aforementioned examples are dynamic systems, evolving by means of choices taken by a large number of competitive entities, game theory provides an appropriate framework for analysis. Note that optimizing aggregation in this form can be seen as an instance of the densest t-subgraph problem which is known to be NP-hard [FPK01] and likely to be hard to approximate [GL09, FPK01, BCC$^+$10]. Given the complexity of the underlying problem it is natural to ask whether competitive entities are able to achieve high levels of aggregation. Specifically, we can ask: what strategies drive selfish behavior to form aggregated networks? In this work we consider a

[1] We assume that each node of the graph accommodates at most one individual.

natural *class* of possible behaviors that players can follow, and we analyze the whole spectrum of games defined by this class. We identify behaviors that define games yielding high aggregation as well as subclasses of them that inherently incur low aggregation.

For specific players' behaviors, we measure the quality of aggregation by studying the Nash equilibria[2] of the corresponding game; i.e., placements of the population for which no individual has an incentive to move from its current position. Our main focus is the study of the *price of anarchy* in our games, which refers to the ratio of an optimum centralized solution to the *worst* Nash equilibrium [KP09, Pap01]. In addition, we study the *price of stability* (also known as "optimistic price of anarchy"), which is defined as the ratio of the optimum to the *best* Nash equilibrium [ADK$^+$08, ADTW03]. The price of stability is useful in applications where a central authority proposes a collective solution so that every player has no incentive to unilaterally deviate from it. On the other hand, the price of anarchy captures worst-case situations where no central coordination exists. A low price of anarchy implies good outcomes of the game even when players act exclusively in their own interest.

5.1 Results

In this work, we initiate a game-theoretic study of aggregation phenomena[3] which can be considered as a competitive version of the k-induced subgraph problem. Our findings highlight the significance of *heterogeneity* in achieving high levels of aggregation. We show that *heterogeneous* populations (i.e., not all individuals follow the same strategy) composed of individuals following very simple, yet diverse, strategies, outperform *homogeneous* populations (i.e., all follow the same strategy) regardless of how sophisticated the strategy followed by the latter is. In

[2] In this paper we are only concerned with pure Nash equilibria; i.e., equilibria based on deterministic strategies. It is an interesting question how the price of anarchy and stability would change when considering mixed strategies.

[3] We emphasize that in this work we study *aggregation phenomena of populations* that occupy a network. Aggregation in our context has nothing to do with *information aggregation* which refers to economics mechanisms designed explicitly for the purpose of collecting and aggregating information.

Table 5.1: Summary of all upper and lower bounds for price of anarchy and price of stability.

	Population	Price of Anarchy		Stab.	Reference
		LB	UB		
Homog.	Followers	$\Theta(t)$ (even for $t = \Omega(n)$)		1	Obs. 44
	Leaders	∞		∞	Obs. 46
	Arbitrary	$\Theta(t)$ (even for $t = \Omega(n)$)			Thm. 47
Heterog.	Mixed	$\min\{\Theta(t), \Theta(n/t)\}$		$1 + \epsilon$	Thm. 48,49,51,52
Informed homogeneous		$\Omega(n/t)$	$O(\min\{t, n/t\})$	$1 + \epsilon$	Thm. 57,56,58,52

particular, we show that a population composed of only two types of individuals, the *leaders* – who have a tendency to "invest" by moving to high degree nodes (in hope that other players will follow) – and the *followers* – who are more prudent and look for immediate rewards – achieves price of anarchy which is asymptotically lower than that achieved by *any* homogeneous population.

Our results suggest that the power of diversity manifests itself more significantly in large populations (i.e. when the number of players t is comparable with the number n of nodes), in which case the gap in the price of anarchy between heterogeneous and homogeneous populations can be as large as $\Omega(n)$. Interestingly, we show that homogeneous strategies cannot outperform the simple heterogeneous strategy even if they are provided with additional information about the parameters of the game.

For all the games we study the best-response dynamics and prove fast convergence. We also consider the price of stability of our games. In particular, for the population obtained as a mixture of followers and leaders the price of stability can be made arbitrarily close to 1 by tuning the mixing parameter (while preserving a low price of anarchy). We tighten this result by showing that no population (even heterogeneous) can achieve optimal price of stability and low price of anarchy at the same time. Our results are summarized in Table 5.1.

Discussion

In order to model heterogeneous populations one can take two possible views. In one, all players have the *same true payoff* (number of neighbors in the present context) but adopt *different strategies* towards optimizing their payoff. In this case, the global welfare is the sum of the players' payoffs which corresponds to the number of induced edges. In the second view, there are two types of players with two *different payoff functions*, but the social welfare is *not* the sum of the players' payoffs. The most natural view for this work is the first one: All players have the same ultimate goal, i.e. maximize their neighbors, and act strategically towards this goal.

5.2 Related work

The aggregation game we study in this work can be interpreted as a network formation game, where the subgraph induced by the individuals at equilibrium represents the created network. Several network creation games of different flavors have been considered in the literature and most of them are related to network design [Vet02, FLM$^+$03, ADK$^+$08, ADTW03, CP05] and social networks [BG00, Jac05]. One of the common settings [BG00, FLM$^+$03, CP05, AEED$^+$06] assumes that each player is associated to a particular node (during the entire course of the game) and can buy edges to any other node (i.e., the underlying graph is complete). The goal of each player is to minimize the distances to all other nodes paying as little as possible. Most of the work for this game aims to bound the price of anarchy [FLM$^+$03, CP05, AEED$^+$06]. Another line of work for network formation can be interpreted as a competitive version of the Steiner tree problem [ADK$^+$08, ADTW03], and focus on bounding the price of stability of the game, since the price of anarchy can be $\Omega(n)$ in graphs of n nodes.

All the literature we mentioned associates costs with edges of the networks, so that good solutions try to avoid dense graphs. Instead, in the games we consider, edges are beneficial. A different line of work in social networks that takes into account this aspect is exchange theory. A vast body of empirical evidence in this field shows that high-degree nodes represent more powerful positions in networks [Eme62, Wil99]. In particular, the bargaining problem has received considerable attention and a study of it in general networks is provided in [KT08b].

There has also been a lot of attention recently in circumventing high price of anarchy of certain games. A line of work considers the noisy best-response dynamics, which reach high-quality states with high probability but only after exponentially many steps [Blu03, MS12]. In [SW09, BBM09], high price of anarchy is circumvented by centrally coordinating some of the players. In particular, [BBM09] considers a model in which a globally optimal behavior (which brings to the optimum) is proposed and a fraction of the players follows this advise for a while but ultimately acts in a way that maximizes their utility. Finally, [BBM10] considers a model where each player uses an experts learning algorithm to choose between an optimal (but untrusted) behavior and the best response dynamics. Observe that in our work we consider games with different classes of players, but (a) players in each class are not centrally controlled, and (b) none of the classes follows an optimal behavior (each class separately fails indeed to achieve low price of anarchy).

Related to our work is also a seminal study of *segregation* by economist Thomas Schelling [Sch71, Sch78]. The general formulation of the model proposed by Schelling assumes a population residing in the cells of a grid. Each cell has eight adjacent cells (including diagonal contact). Also, each individual of the population is either of type A or B (the type represents some characteristic such as race, ethnicity, etc.) and wants to have at least r adjacent individuals of its own type, where r is a satisfaction threshold common to all individuals. The system evolves in steps, and at each step an unsatisfied individual is selected and moved to a cell offering more neighbors of its own kind. Interestingly, experiments simulating this model display a high level of segregation of the two kind of individuals even with a mild threshold r (e.g., $r = 3$). Observe that the incentives of the individuals are in fact aggregation rules (as opposed to segregation rules), therefore Schelling's model can be interpreted as a model of aggregation as well. Throughout the paper we will point out some relations of our games to the scenario proposed by Schelling.

Roadmap

We discuss preliminaries in Section 5.3. Section 5.4 is dedicated to homogeneous populations and provides the theorem establishing their inherent high price of anarchy. In Section 5.5, we consider the population obtained by a mixture of followers and leaders and show that it yields both low price of anarchy and price of stability. In Section 5.6 we consider possible extensions. We conclude with future directions in Section 5.7.

5.3 Preliminaries

Consider an undirected graph $G = (V, E)$ and t players, where t possibly depends on $|V|$. For a placement $H \subseteq V$ (with $|H| = t$) of the players onto the graph, the *global welfare* is defined as the number of edges induced by H in G. We will say that a placement is *optimum* if it induces the maximum possible number of edges. For convenience we will often use the term optimum to also indicate the value of an optimum placement (that is, the number of induced edges). Note that it is NP-hard to find an optimum placement for general graphs because it is equivalent to solving the densest t-subgraph problem [FPK01]. Also, this problem is likely to be hard to approximate since the best known centralized algorithm gives a $O(n^{1/4+\epsilon})$ approximation [BCC$^+$10]. To circumvent this computational barrier, we are mainly interested in the case $t = \Theta(n)$ when the t-densest subgraph admits a constant factor approximation, but we show our results for general t as well.

Given a placement H of players onto a graph G, we let $\Gamma_H(u)$ denote the degree of the node u in H (that is, the number of adjacent individuals that a hypothetical player located in u would have under placement H). Similarly, we let $\overline{\Gamma}_H(u)$ be the degree of u in the graph obtained by G after removing the edges in H (that is, the number of *empty* adjacent positions that a hypothetical player located in u would have under H). Clearly for all placements $H \subseteq V$, $\deg_G(u) = \Gamma_H(u) + \overline{\Gamma}_H(u)$.

It is easy to see (Section 5.4.1) that if every player plays the most natural strategy, that of greedily moving to a location with the highest number of neighbors, then the equilibria can be very poor compared to an optimum placement. Therefore, we will consider richer classes of games, where players might make decisions that take into account possible future benefits. In all our games, the way every player i decides where to move can be described in the following manner: player i "ranks" every (available) location in the graph through a *ranking function* $f_i(\cdot)$ that gives a score to each location (i.e. node) u of the graph with respect to the current configuration, and moves to the location with highest score. The functions $f_i(u)$ we consider are "local" to the location u, in the sense that $f_i(u)$ depends only on the current configuration of the neighborhood of u, i.e. on $\Gamma_H(u)$ and $\overline{\Gamma}_H(u)$.

Most of our proofs are obtained analzying configurations reached by a *best-response dynamics*. Best-response dynamics studies the game in an evolving fashion. Specifically, the system evolves in steps: at each step a player is chosen and given the opportunity to move to a new better location with respect to its ranking function. The way players are chosen depends on some (possibly randomized) scheduling. We note that Schelling's original work on segregation [Sch71, Sch78] also uses a best-response dynamics to model evolution.

5.3.1 Notation

We are interested in undirected and connected[4] graphs. Given an undirected graph $G = (V, E)$ and any $S_1, S_2 \subseteq V$ we denote by E_{S_1,S_2} the set of edges with one endpoint in S_1 and the other in S_2. When clear from the context, the same notation will be used for the cardinality of the edges from S_1 to S_2. Abusing notation, we will use E_S instead of $E_{S,S}$. Since we only consider undirected graphs, $E_{S_1,S_2} = E_{S_2,S_1}$. Likewise we use $\tilde{d}_{S_1,S_2}$ to denote the average degree of nodes in S_1 when considering edges only in E_{S_1,S_2} (notice that $\tilde{d}_{S_1,S_2} \neq \tilde{d}_{S_2,S_1}$). It is not hard to see that $\tilde{d}_{S,S} = \frac{2E_S}{|S|}$ while if $S \cap T = \emptyset$, $\tilde{d}_{S,T} = \frac{E_{S,T}}{|S|}$. Also if $T_1, ..., T_k$ form a partition of T then $E_{S,T} = \sum_{i=1}^{k} E_{S,T_i}$ and $\tilde{d}_{S,T} = \sum_{i=1}^{k} \tilde{d}_{S,T_i}$. Finally, we will use t and n to denote the *size of the population* and the *size of the graph* under

[4] The connectivity requirement can be dropped for most of our results. However, there are cases for which disconnected graphs give rise to some rare pathologies (e.g., when the graph is an independent set) that need special care.

consideration respectively.

5.4 Homogeneous Populations

In this section we analyze populations where all individuals have the same ranking function, i.e. $f_i = f_j$ for all players i, j. We call such populations *homogeneous*. We start by studying two very natural strategies and prove that both fail in achieving a low price of anarchy. We conclude the section by showing that a high price of anarchy is inherent in all homogeneous populations regardless of the ranking function they use.

5.4.1 A Population of Followers

We begin by looking at the most natural ranking function for the individuals which makes a player move to another (non-occupied) location if it offers more adjacent players than its current location. Formally, given a placement H of the players onto the graph, the ranking function of each player is defined by:

$$f(u) := \Gamma_H(u),$$

and the player is incentivized to move to another location v if $v \notin H$ and $\Gamma_{H'}(v) > \Gamma_H(u)$, where $H' = (H \setminus \{u\}) \cup \{v\}$. Note that the evolution of this game captures Schelling's model provided that the satisfaction threshold r is large (see related work).

The price of anarchy and stability of this simple game are established in the following result which also demonstrates a fast convergence of the best-response dynamics. As such, an equilibrium is easy to find.

Observation 44. *Consider the aggregation game with a population of followers. Then, for any connected graph, the price of stability is exactly* 1*, the price of anarchy is* $O(t)$*, and there are connected graphs of size* n *with price of anarchy as high as* $\Omega(t)$ *even for* $t = \Theta(n)$*. Finally, best-response dynamics converges in polynomial time.*

Proof. For the polynomial convergence of the best-response dynamics, define the simple (potential) function $R(H) = E_H$. Every time a player moves from u_1 to u_2 then the new configuration is $H' = H \setminus \{u_1\} \cup \{u_2\}$ with $R(H') - R(H) = \Gamma_{H\setminus\{u_1\}}(u_2) - \Gamma_{H\setminus\{u_1\}}(u_1) \geq 1$. Also $R(H) = O(t^2)$ for all placements H of size t and therefore the aggregation game with a population of followers reaches an equilibrium after at most $O(t^2) = O(n^2)$ iterations. For the price of stability, let $G_t = (V_t, E_t)$ be a t-densest subgraph of G. Then $H = V_t$ is an equilibrium (since G_t is a densest subgraph of G) and hence $\frac{E_t}{E_H} = 1$. For the upper bound on the price of anarchy note that any subgraph of size t has at most $O(t^2)$ edges and any equilibrium has at least $\Omega(t)$ edges (since the graph is connected). Finally, for the lower bound consider the graph shown on the left in Figure 5.2 with $N = 0$ and $k = t$. The configuration shown is an equilibrium with $E_H = O(t)$ whereas the optimum subgraph of size t has $\Omega(t^2)$ edges. Therefore the price of anarchy can be as high as $\Omega(t) = \Omega(n)$ for $t = \Theta(n)$. □

An optimal price of stability is appealing, however we show that in general, these optimal equilibria do not even satisfy some simple requirements. As explained in the related work section, the model provided by Schelling embeds a notion of satisfaction for the individuals[5]. Along the same lines, for any graph, we can classify Nash equilibria with respect to the minimum satisfaction among the individuals. Specifically, we say that a Nash equilibrium is r-stable if every individual has at least r adjacent players. We define the price of r-stability as the ratio of the optimum to the best r-stable equilibrium. As discussed previously, the price of stability can be interpreted as a reasonable solution proposed from a central authority to the players. In light of this, Observation 44 suggests that there are proposals of optimum value such that no individual has incentive to move. On the contrary, the following observation implies that if we were to look for proposals that guarantee even just a small amount of satisfaction for every individual (and assuming that satisfied individuals do not move), then the overall social welfare

[5] An individual is satisfied if it has at least r neighbors, for some threshold r common to all individuals. A satisfied individual has no incentive to move even if there exist available positions with more neighbors.

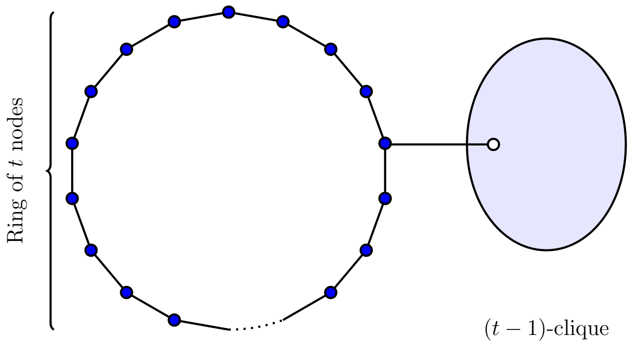

Figure 5.1: 2-stable placement with high price of anarchy. This is used in Observation 45.

can be much worse than the optimum welfare[6].

Observation 45. *There are connected graphs of size n with price of 2-stability of $\Omega(n)$. In general, for connected graphs, the price of r-stability is $\Theta(t/r)$ for $r \geq 2$ and is exactly 1 for $r \in \{0, 1\}$.*

Proof. For $r = 0$ or $r = 1$ the statement is obvious since for connected graphs, all players in any optimum placement H are 1-stable (they have at least one adjacent player). For $r = 2$ consider the graph shown in Figure 5.1. Clearly the optimal size-t subgraph has $O(t^2)$ edges. However, the only 2-stable placement is the one where all t players are placed on the ring of size t (notice that the clique can accommodate at most $t-1$ players). Hence, the best 2-stable equilibrium has $O(t)$ edges which implies that the price of 2-stability is $\Omega(t) = \Omega(n)$. For $r > 2$ the proof is identical (we need only replace the ring with an r-regular graph). □

Finally, we note that deciding if there exists an r-stable equilibrium is NP-hard for any $r \geq 3$ [APP$^+$12].

5.4.2 A Population of Leaders

The previous game failed in providing a low price of anarchy due to the fact that the individuals were short-sighted and did not look for long-term rewards. In

[6] A common measure for the social welfare of an equilibrium is the *egalitarian* objective function which is defined as the maximum player's utility. The argument above instead quantifies the quality of equilibria by their minimum player's utility.

particular, the followers' function failed to spot strategic positions in the graph. In this section we analyze a population of "leader" individuals that tries to overcome this issue. Specifically, individuals will move to high degree nodes even if they do not offer many adjacent individuals at the time of the move. In other words, individuals are investing in empty positions with the hope of gaining many adjacent players as the system evolves. Given their relation to common measures such as betweenness, high degree nodes play an important role in the study of power in social networks [Eme62, Wil99].

In order to account for high-degree nodes we define the following ranking function:

$$\ell(u) := \Gamma_H(u) + \overline{\Gamma}_H(u).$$

A player moves to a node v from v if $\ell(v) > \ell(u)$. Unfortunately, this population performs even worse than a population of "followers".

Observation 46. *The best-response dynamics of the game converges in polynomial time. However, there exist connected graphs for which all Nash equilibria have zero social welfare.*

Proof. For the convergence of best-response dynamics, consider the simple (potential) function $R(H) = \sum_{u \in H} \ell(u)$. Notice that every time a player moves $R(H)$ increases by at least 1. However $R(H)$ is bounded by $t(n-1)$ therefore the population achieves an equilibrium after at most $t(n-1)$ iterations. For the second part of the proof, consider the graph on the right of Figure 5.2 with $k = 5$. All its nodes have degree either 2 or 4. Hence, all individuals will eventually move to the nodes of degree 4. This placement induces no edges between individuals and hence the price of stability is infinite. □

5.4.3 Lower Bounds for Homogeneous Populations

In the previous sections we analyzed two simple kinds of populations: followers and leaders. For different reasons, both populations failed in ensuring a non-trivial price of anarchy. At this point, one could be tempted to think that more sophisticated ranking functions might result in lower prices of anarchy. In

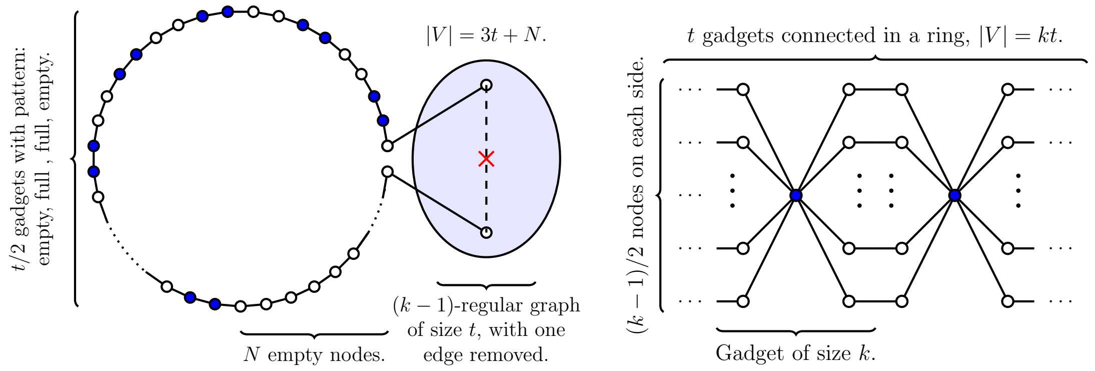
t/2 gadgets with pattern:
empty, full , full, empty.
$|V| = 3t + N$.
N empty nodes.
$(k-1)$-regular graph
of size t, with one
edge removed.
t gadgets connected in a ring, $|V| = kt$.
$(k-1)/2$ nodes on each side.
Gadget of size k.

Figure 5.2: Graphs for lower bounds. They are used in the proof of Theorem 47.

this section, we show that this is not the case. On the contrary, the seemingly naive strategy of followers gives (in asymptotic terms) the lowest possible price of anarchy among all homogeneous strategies. More specifically, in Theorem 47, we show that *any* homogeneous population cannot yield low price of anarchy. Interestingly, the graphs used in Observations 44 and 46 entirely capture the hardness of achieving a low price of anarchy and play a central role in the proof of the aforementioned lower bound.

Theorem 47. *Consider any homogeneous population. Then, there exists an infinite increasing sequence of* $\{t_i\}_{i=1}^{\infty}$ *such that for all* $n_i \geq 3 \cdot t_i$ *there exists a connected graph on* n_i *nodes on which the homogeneous population of size* t_i *has price of anarchy at least* t_i*.*

Proof. In the case of homogeneous populations, the ranking function of the individuals can be represented as a table $s(i,j)$, where $s(i,j)$ denotes the value of the function at a node with i adjacent individuals and j adjacent empty positions. First, we claim that it must be that $s(1,1) > s(0,d)$ for every $d \geq 2$. Suppose not: then the placement shown on the right of Figure 5.2 with $k = d+1$ (a similar graph can be constructed for odd d as well) is an equilibrium (note that every node in the graph has either degree d or 2) with zero social welfare and therefore infinite price of anarchy. So, we may assume that $s(1,1) > s(0,d)$ for every $d \geq 2$. But then, for any even t (and $n \geq 3t$), the placement on the left of Figure 5.2 with $k = t$ is an equilibrium that induces $t/2$ edges, while the optimum is obtained by placing the t individuals in the $(t-1)$-regular graph of size t which yields $(t-1)t/2 - 1$ induced edges. Therefore the price of anarchy in this case is at least t which concludes the proof. □

5.5 Heterogeneous Populations

In Section 5.4, we showed that neither a population of followers nor a population of leaders can achieve a low price of anarchy. Even worse, Theorem 47 suggests that we cannot hope for low price of anarchy when considering homogeneous populations. However, it leaves open the door for heterogeneous populations. It is natural to ask how many different "classes" of individuals are required in order to reduce the price of anarchy or even how complex the strategies of each class should be. In this section we settle both questions with a favorable answer that suggests an extreme separation between homogeneous and heterogeneous populations: while the naive populations of only leaders or only followers have high price of anarchy ($\Omega(t)$, even for $t = \Theta(n)$) when considered separately, we show that a simple hetergeneous population composed of a mixture of the two achieves a low price of anarchy, in particular a *constant* price of anarchy for $t = \Theta(n)$.

We also study the price of stability achieved by the heterogeneous population. Namely, in Theorem 51, we prove that the price of stability can be made arbitrarily close to 1 by tuning the mixing parameter (while maintaining a low price of anarchy). We conclude the section by proving that this is essentially the best price of stability one can achieve without increasing the price of anarchy. More specifically, we provide an impossibility theorem showing that no population can achieve an optimal price of stability and a low price of anarchy simultaneously (see Theorem 52).

5.5.1 Achieving Low Price of Anarchy

Consider a λ-*heterogeneous population* (for some $0 < \lambda < 1$), with λt leaders (players with ranking function ℓ from Section 5.4.2) and $(1-\lambda)t$ followers (players with ranking function $f(u)$ from Section 5.4.1).

The following theorem shows that the best-response dynamics of such a game converges in polynomial time and provides upper bounds for the price of anarchy. Interestingly, the price of anarchy of λ-heterogeneous populations with *constant* λ, is upper bounded by $O(\sqrt{n})$ and can be as low as constant when $t = \Theta(n)$.

Theorem 48. *Fix any $0 < \lambda < 1$ and any connected graph G of n nodes. Then the λ-heterogeneous population achieves a constant price of anarchy for $t = \Theta(n)$.*

In general, the price of anarchy is $O\left(\min\left\{\frac{1}{1-\lambda}t, \frac{1}{\lambda(1-\lambda)}\frac{n}{t}\right\}\right)$. In addition, best-response dynamics converges in polynomial time.

Proof. The proof of polynomial time convergence is a simple combination of Observations 44 and 46. First notice that the number of leaders' moves cannot exceed $O(\lambda tn)$ (every time a leader moves the potential function $\sum_{u:leader} \deg_G(u)$ increases by at least 1 and this sum cannot exceed the value λtn). Now conditioned on leaders not moving, followers' moves are also polynomially bounded (Observation 44). The two bounds together guarantee polynomial time convergence for the whole population.

Let now H be the set of nodes occupied by the population in any equilibrium. We will use $F, L \subseteq H$ to denote the set of nodes occupied (upon convergence) by followers and leaders respectively. We have $|L| = \lambda t, |F| = (1-\lambda)t$. Also let B the subset of nodes of any t-densest subgraph of the graph. We want to bound the price of anarchy, i.e. E_B/E_H. For the upper bound of $t/(1-\lambda)$, simply observe that $E_B < t^2/2$ while $E_H \geq (1-\lambda)t/2$ since the followers will have at least one neighbor because the graph is connected.

For the other bound, we define ℓ_0 (resp. f_0) to be the minimum value of the ranking function ℓ (resp. f) over the positions of the leaders (resp. followers) in H. That is, $\ell_0 = \min_{u \in L} \ell(u)$ and $f_0 = \min_{u \in F} f(u)$. We observe that every node in $B \setminus H$ can not have more than $f_0 + 1$ neighbors in H otherwise H would not be an equilibrium. Also there are at most t nodes in $B \setminus H$ and therefore $E_{B\setminus H,H} \leq (f_0+1)\cdot t$. On the other hand, $E_H \geq f_0 \cdot (1-\lambda)t/2$. By combining these two inequalities with the fact that $E_B \leq E_H + E_{B\setminus H,H} + E_{B\setminus H}$, we conclude that the price of anarchy is

$$\begin{aligned}\frac{E_B}{E_H} &\leq \frac{E_H}{E_H} + \frac{E_{B\setminus H,H}}{E_H} + \frac{E_{B\setminus H}}{E_H} \\ &\leq 1 + \frac{4}{1-\lambda} + \frac{E_{B\setminus H}}{E_H}.\end{aligned}$$

It remains to bound term $E_{B\setminus H}/E_H$. We start showing a lower bound for E_H. Observe that

$$\ell_0 \cdot \lambda t \leq \sum_{u\in L} \ell(u) \leq 2E_{L,H} + E_{L,V\setminus H}. \tag{5.1}$$

In addition, in any equilibrium H, the average degree $E_{V\setminus H,H}/|V\setminus H|$ from $V\setminus H$ to H is at most the average degree $E_{F,H}/|F|$ of F to H. Also we have $E_{V\setminus H,H} \geq E_{V\setminus H,L}$ from which $\frac{E_{F,H}}{|F|} \geq \frac{E_{V\setminus H,L}}{|V\setminus H|}$. This implies that $\frac{E_{F,H}}{(1-\lambda)t} \geq \frac{E_{V\setminus H,L}}{n}$ which entails $E_{V\setminus H,L} \leq \frac{n}{(1-\lambda)t}E_{F,H}$. Combining the last inequality with (5.1) we get

$$\begin{aligned}2E_{L,H} + \frac{n}{(1-\lambda)t}E_{F,H} &\geq \ell_0\lambda t \quad \Rightarrow \\ 2\frac{n}{(1-\lambda)t}(E_{L,H} + E_{F,H}) &\geq \ell_0\lambda t\end{aligned}$$

Also we know that $2E_H \geq E_{L,H} + E_{F,H}$, therefore the above implies that $E_H \geq \ell_0\lambda(1-\lambda)t^2/(4n)$. We now bound $E_{B\setminus H}$. Note that the nodes in $B\setminus H$ cannot have degree more than ℓ_0, otherwise H would not be an equilibrium, therefore $E_{B\setminus H} \leq \ell_0 \cdot t$. Combining the bounds on E_H and $E_{B\setminus H}$ we obtain $\frac{E_{B\setminus H}}{E_H} \leq 2\frac{n}{\lambda(1-\lambda)t}$ from which the theorem follows. □

The following Theorem shows that the upper bound for the price of anarchy of the aforementioned heterogeneous strategy is asymptotically tight.

Theorem 49. *For the λ-heterogeneous population, for any λ and t, there exists a connected graph of size n such that the price of anarchy is $\Omega\left(\min\left\{\frac{1}{1-\lambda}t, \frac{1}{\lambda(1-\lambda)}\frac{n}{t}\right\}\right)$.*

Proof. We present a construction of a disconnected graph for simplicity. It is not hard to extend it to a connected graph with similar price of anarchy. Consider a graph G that contains a t-clique (let V_c be the set of the corresponding nodes) and a bipartite graph (V_1, V_2) with $|V_1| = \lambda t$ and $|V_2| = n_0 = n - \lambda t - t$. Each node of V_1 has $t-1$ edges with nodes from V_2 so that the degree of the nodes in V_2 is equally distributed and is at most $\lceil \frac{\lambda t(t-1)}{n_0} \rceil \leq 1 + \frac{\lambda t(t-1)}{n_0}$. The densest t-subgraph is the t-clique which gives $t(t-1)/2$ edges. Now notice that if all the leaders are placed on the nodes of V_1 and all the followers on the highest degree nodes of V_2 we get an equillibrium (call this placement H). Notice that each follower will have at most $1 + \frac{\lambda t(t-1)}{n_0}$ edges, so the total number of edges is $(1-\lambda)t(1 + \frac{\lambda t(t-1)}{n_0})$. Now if $1 < \frac{\lambda t(t-1)}{n_0}$, then $E_{V_c}/E_H > \frac{n_0}{2\lambda t(1-\lambda)}$, while if $1 > \frac{\lambda t(t-1)}{n_0}$, $E_{V_c}/E_H > \frac{t-1}{2(1-\lambda)}$. □

5.5.2 Price of Stability and Relation to Price of Anarchy

We now investigate the price of stability for the heterogeneous population. We need the following technical lemma to establish our main result.

Lemma 50. *Let $G = (V, E)$ be a graph and $G_t = (V_t, E_t)$ a densest t-size subgraph of G. Then $\forall k$ with $2 \leq k \leq t$ there exists a subgraph $G_k = (V_k, E_k)$ of size k such that $E_{V_k} \geq \frac{k(k-1)}{t(t-1)} E_{V_t}$.*

Proof. In fact we will prove that there exists a subgraph G_k of G_t which has at least a $\frac{k(k-1)}{t(t-1)}$ fraction of G_t's edges. Let $S_k(G_t)$ be the set of all possible subgraphs of G_t that have size *exactly* k (clearly $|S_k(G_t)| = \binom{t}{k}$) and $E^*_{V_k}$ be the number of edges of the optimum size-k subgraph of V_t. Each edge e of E_{V_t} belongs to exactly $\binom{t-2}{k-2}$ size-k subgraphs of G_t. Therefore, we can write $\sum_{H \in S_k(G_t)} E_H = \binom{t-2}{k-2} E_{V_t}$ which in turn implies $\binom{t}{k} E^*_{V_k} \geq \binom{t-2}{k-2} E_{V_t}$. The theorem follows by rearranging and simplifying the terms. □

The following theorem shows that an almost optimal price of stability can be achieved while preserving a low price of anarchy.

Theorem 51. *For every constant $\epsilon > 0$, there exists a constant $\lambda = \lambda(\epsilon) > 0$ such that the mixed population with parameter λ achieves price of stability of at most $1 + \epsilon$.*

Proof. Set $\lambda = \lambda(\epsilon) = \frac{1}{2}\left(1 - \frac{1}{\sqrt{1+\epsilon}}\right) \in (0, 1)$. and let $\alpha = \alpha(\epsilon) = 1 - \lambda = \frac{1}{2}\left(1 + \frac{1}{\sqrt{1+\epsilon}}\right)$. Let H' be a αt-densest subgraph of G and D the subset of λt highest degree nodes in G. Consider a placement where the λt leaders are placed in D and the followers are placed in $H' \setminus D$. If $|H' \setminus D| < (1-\lambda)t = \alpha t$ then the remaining followers are placed in arbitrary positions in the graph. Let H_0 be this initial placement. First notice that $H' \subseteq H_0$ since $\alpha t = (1-\lambda)t = |H'| \geq |H' \setminus D|$. Allow the individuals move according to the best-response dynamics until they reach an equilibrium H. Note that, throughout the game, leaders won't move since they have initially been placed in the highest degree nodes. This means that only followers move and hence the total number of edges among individual can only increase ($E_H \geq E_{H_0} \geq E_{H'}$). Let now B be a t-densest subgraph of G. We then have

$$\begin{aligned}\frac{E_H}{E_B} &\geq \frac{E_{H'}}{E_B} \overset{\text{Lemma 50}}{\geq} \frac{\alpha t(\alpha t-1)}{t(t-1)} \\ &= \frac{\alpha^2(t-1/\alpha)}{t-1} = \alpha^2\left(1-\frac{1/\alpha-1}{t-1}\right) \\ &\overset{t\geq 2}{\geq} \alpha^2\left(2-1/\alpha\right) \geq \frac{1}{1+\epsilon}\end{aligned}$$

where in the last inequality we used the fact that by definition $\alpha \geq \frac{1}{\sqrt{1+\epsilon}}$ and $2\alpha - 1 = \frac{1}{\sqrt{1+\epsilon}}$.

□

Theorem 51 guarantees an arbitrarily good price of stability for the heterogeneous population. The following question arises: Can we achieve an optimum or quasi-optimum price of stability (that is, approaching 1) without increasing the price of anarchy? The following theorem provides a negative answer stating that any population (homogeneous or heterogeneous) that achieves a quasi-optimum price of stability *must* have a high price of anarchy. We emphasize that the following theorem holds for *any* kind of heterogeneous population: specifically, it holds true even if every individual has a different strategy.

Theorem 52. *Consider any (even heterogeneous) population of t individuals. Suppose that for any connected graph the price of stability is at most $1 + 1/n$. Then, there exist connected graphs for which the price of anarchy is $\Omega(n)$.*

Proof. We consider the two graphs in Fig. 5.3. We will show that quasi-optimum stability on the left graph implies $\Omega(n)$ price of anarchy on the right graph. Clearly

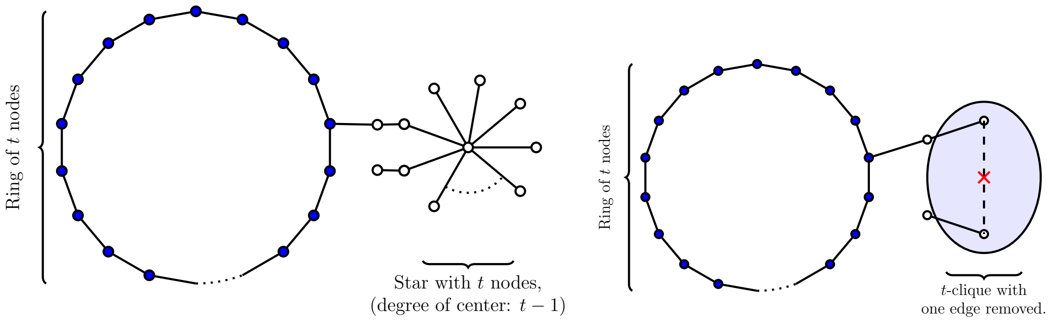

Figure 5.3: Graphs for Theorem 52 showing the impossibility of good price of anarchy given optimal price of stability.

the optimal arrangement H for the left graph is obtained by placing all the players on the ring which yields $E_H = t$. Also, *any* other arrangement gives at most $t-1$ edges. Therefore if the price of stability is less than $1 + 1/n < t/(t-1)$, it should be the case that the players on the ring do not have incentive to move to the center of the star, or the bridge. Now consider the graph on the right where the star is replaced by a t-clique with one node removed (the t-densest subgraph has $t(t-1)/2 - 1$ edges). Notice that the nodes of the clique have the same value (degree) as the central node for the star[7]. So for any population with stability less than $1 + 1/n$, the ring is an equilibrium and therefore the price of anarchy can be as high as $(t-1)/2 - 1/t = \Omega(t) = \Omega(n)$. □

5.6 Extensions

Under our model, we proved that any homogeneous strategy is bound to have high price of anarchy (i.e. $\Omega(t)$ even for $t = \Theta(n)$), while a mixture of leaders and followers achieves a low price of anarchy, in particular a constant price of anarchy for $t = \Theta(n)$. In this section we provide a less strict notion of leaders and show how it affects the price of anarchy (when mixed with followers). Moreover, we connect this new concept to a new, more powerful kind of homogeneous

[7] Notice the existence of this extra node of degree 1 that is present in both graphs of Figure 5.3. Since we want the clique to connect with the ring, we need the bridge node but then have to remove one edge from the clique in order to maintain degree $t-1$ for the nodes in the clique. However, removing this edge leaves the node shown on the bottom of the clique in the right graph with degree $t-2$. The extra node added (which is adjacent to only this bottom node in the clique) is there to ensure degree $t-1$ for the bottom node too.

populations.

5.6.1 Generalized β-leaders

In this section we consider a generalized definition for a population of leaders. Leaders as defined in section 5.4.2 make decisions based solely on the total number of adjacent nodes regardless of how many of them are occupied. In other words leaders actions are somewhat indifferent towards aggregation[8]. We can obtain a more natural behavior considering a ranking function of the kind $\ell_\beta(u) := \Gamma_H(u) + \beta\overline{\Gamma}_H(u)$, where $0 \leq \beta \leq 1$. We call β-leaders individuals with this ranking function. The parameter β is the relative weight of an adjacent empty position to an adjacent individual. As such, it quantifies how much players are willing to invest in empty positions. Notice that for $\beta = 0$, $\ell_\beta(u)$ falls back to the ranking function of a *follower*, while for $\beta = 1$ we obtain a (pure) *leader*.

The convergence of the best-response dynamics in this game is not immediate. Define

$$\begin{aligned} R_\beta(H) &= \frac{1}{2}\sum_{u\in H}\Gamma(u) + \frac{\beta}{1+\beta}\sum_{u\in H}\overline{\Gamma}(u) \\ &= E_H + \frac{\beta}{1+\beta}E_{H,V\setminus H}, \end{aligned}$$

where H is some placement of the individuals onto the graph. By definition of R_β and ℓ_β,

$$\sum_{u\in H}\ell_\beta(u) \geq R_\beta(H) \geq \frac{1}{2}\sum_{u\in H}\ell_\beta(u) \tag{5.2}$$

The following lemma establishes that R_β is a potential function, that is, all equilibria of the game are local optima of R_β and viceversa.

Lemma 53. *The function $R_\beta(H)$ strictly increases at each step of the best-response dynamics. Moreover, best-response dynamics converges in polynomial time.*

[8] To be more precise, leaders do seek aggregation even though they do so in a more indirect way (and not on a per step basis) by creating the conditions for better aggregation in the future. In fact this strategy turns out to be quite succesful when leaders are mixed with followers.

Proof. Let H_1 be the subgraph induced by any placement of the individuals, $H_2 = H_1 \cup \{u_2\} \setminus \{u_1\}$ be the induced subgraph after one step of the best-response dynamics (where an individual moved from u_1 to u_2) and $H' = H_1 \cap H_2$. Then we have: $\Delta R = R_\beta(H_2) - R_\beta(H_1) = E_{H_2} - E_{H_1} + \frac{\beta}{1+\beta}(E_{H_2,V\setminus H_2} - E_{H_1,V\setminus H_1})$. We will show that this quantity is strictly positive. First we observe that $E_{H_1,V\setminus H_1} = E_{H',V\setminus H'} - E_{u_1,H'} + E_{u_1,V\setminus H'}$, which implies $E_{H_2,V\setminus H_2} - E_{H_1,V\setminus H_1} = -E_{u_2,H'} + E_{u_2,V\setminus H'} + E_{u_1,H'} - E_{u_1,V\setminus H'}$. Also we have $E_{H_2} - E_{H_1} = E_{u_2,H'} - E_{u_1,H'}$. Therefore,

$$\begin{aligned}
\Delta R &= E_{u_2,H'} - E_{u_1,H'} \\
&\quad + \frac{1}{1+\beta}(-E_{u_2,H'} + E_{u_2,V\setminus H'} \\
&\quad + E_{u_1,H'} - E_{u_1,V\setminus H'}) \\
&= \frac{1}{1+\beta}[\big(E_{u_2,H'} + \beta E_{u_2,V\setminus H'}\big) \\
&\quad - \big(E_{u_1,H'} + \beta E_{u_1,V\setminus H'}\big)] \\
&= \frac{1}{1+\beta}\left[\ell_\beta(u_2) - \ell_\beta(u_1)\right] > 0.
\end{aligned}$$

□

Note that Theorem 47 implies that, for any constant $0 \le \beta \le 1$, a population composed excusively of β-leaders has high price of anarchy. Analogously to the λ-heterogeneous population, we can consider a λ-mixture of β-leaders and followers. For this game we were not able to show convergence of the best-response dynamics. We leave as an open question if this game is a potential game and if it admits convergence for every $0 \le \lambda, \beta \le 1$. As for the quality of the equilibria the following theorem holds. The proof is similar in spirit to the proof of Theorem 48 and is omitted.

Theorem 54. *Consider any $0 < \lambda < 1$, $0 \le \beta \le 1$, and any connected graph G of n nodes. Then for any equilibrium H, and t-densest subgraph B,*

$$\frac{E_B}{E_H} = \begin{cases} O(\frac{t}{1-\lambda}), & 0 \le \beta < \frac{1}{t}, \\ O(\min\{\frac{t}{1-\lambda}, \frac{1}{\beta\lambda(1-\lambda)}\}), & \frac{1}{t} \le \beta \le \frac{1}{2(1-\lambda)}\frac{t}{n}, \\ O(\min\{\frac{t}{1-\lambda}, \frac{1}{\lambda(1-\lambda)}\frac{n}{t}\}), & \frac{1}{2(1-\lambda)}\frac{t}{n} < \beta \le 1, \end{cases}$$

5.6.2 The effects of information

Theorem 47 implies that, for any fixed $0 \leq \beta \leq 1$, there exists a graph where a population of β-leaders behaves poorly. This raises a dual question: is it possible that for any graph there exists a β for which the β-leader population achieves high levels of aggregation? Note that this yields a new (somewhat less realistic[9]) model under which individuals are given more information about the game and they are allowed to adapt their strategies based on it. We show that if we let β depend on two additional quantities, the size n of the graph and the size t of the population, then there are (homogeneous) β-leader populations that achieve lower price of anarchy[10]. However, we observe that the ranking functions need fine tuning and still cannot do better than the simple heterogeneous population we described. In particular, we can show (Theorem 56) that an "informed" homogeneous population of $\beta(n,t)$-leaders with $\beta(n,t) = \frac{\lambda n}{t}$, for $0 < \lambda < 1/4$, achieves price of anarchy $O(\min\left\{\frac{t}{1-4\lambda}, \frac{1}{\lambda(1-4\lambda)}\frac{n}{t}\right\})$. Moreover, this is (asymptotically) the best possible for any homogeneous population that is informed with t and n (Theorem 57). Finally, in Theorem 58, we study the price of stability for this informed population.

Lemma 55. *Consider the aggregation game with the above $\beta(n,t)$-leaders population with parameter λ. For any connected graph $G = (V, E)$ of n nodes and any equilibrium H, it holds that $E_H \geq (1 - 4\lambda)R_{n,t}(H)$.*

Proof. Let ℓ_0 be the minimum value of the ranking function over the individuals in H (i.e. $\ell_0 = \min_{u \in H} \ell_\beta(u)$), and let u_0 be a node in H that has $\ell_\beta(u_0) = \ell_0$. We note that for connected graphs $\ell_0 \geq 1$. Since we are at equilibrium, the individual placed in u_0 does not have any incentive to move, so the rankings of all the nodes in $V \setminus H$ after the player is removed must be at most ℓ_0. This shows that every node in $V \setminus H$ has at most ℓ_0 adjacent individuals in $H \setminus \{u_0\}$, and at most $\ell_0 + 1$ neighbors in H. We conclude that $E_{H,V\setminus H} \leq (\ell_0 + 1)(n - t) \leq 2\ell_0(n - t)$. Also,

[9] In this section we consider ranking functions that can depend on information such as the graph size n and the size t of the population. The ranking functions we considered so far were not allowed to use this information. This is a valid constraint especially in dynamic situations where these quantities might change over time.

[10] When the ranking function of an homogeneous population depends on only one of n and t, it is possible to extend the proof of Theorem 47 to show that the price of anarchy can be as high as $\Omega(n)$.

inequality (5.2) implies that $R_\beta(H) \geq \ell_0 t/2$. Combining the above inequalities we get $E_{H,V\setminus H} \leq 4(n-t)R_\beta(H)/t$. Now write

$$\begin{aligned} E_H &= R_\beta(H) - \lambda \frac{t}{n+\lambda t} E_{H,V\setminus H} \\ &\geq R_\beta(H) - \lambda \frac{4(n-t)}{n+\lambda t} R_\beta(H) \\ &\geq (1-4\lambda) R_\beta(H), \end{aligned}$$

which concludes the proof.

□

The following theorem provides the desired bound on the price of anarchy.

Theorem 56. *Consider the aggregation game with the informed population above with parameter λ. For any connected graph $G = (V, E)$ of n nodes, the price of anarchy is at most*

$$\min\left\{\frac{1}{1-4\lambda}t, 1 + \frac{2}{\lambda(1-4\lambda)}\frac{n}{t}\right\}.$$

Proof. Let H be the set of nodes occupied by the population in any equilibrium and ℓ_0 be the minimum value of ranking function ℓ_β achieved over all individuals in H. Recall that for a connected graph $\ell_0 \geq 1$, and inequality (5.2) implies that $R_\beta(H) \geq \ell_0 t/2$. First we show that E_H cannot be too small. By Lemma 55 we get

$$E_H \geq (1-4\lambda)R_\beta(H) \geq (1-4\lambda)\ell_0 t/2. \tag{5.3}$$

The first part of the bound on the price of anarchy now is immediate since every optimal solution has at most $t^2/2$ edges. For the second part of the bound, let B be a densest subgraph of size t of G. All the nodes in $B \setminus H$ have degree at most $\ell_0 n/(\lambda t)$, or else H is not an equilibrium. As a result $E_{B\setminus H,V} \leq |B \setminus H| \ell_0 n/(\lambda t) \leq \ell_0 n/\lambda$. Therefore the price of anarchy is given by:

$$\begin{aligned} \frac{E_B}{E_H} &\leq \frac{E_H + E_{B\setminus H,V}}{E_H} \leq 1 + \frac{\ell_0 n}{\lambda E_H} \\ &\leq 1 + \frac{2n}{\lambda(1-4\lambda)t}. \end{aligned}$$

□

Theorem 56 shows that the informed population achieves a price of anarchy of $O(n/t)$ for any $t = \Omega(\sqrt{n})$. The following theorem tightens this result by showing that this is the best possible for homogeneous informed populations.

Theorem 57. *For any homogeneous informed population, there exist connected graphs of n nodes for which the price of anarchy is $\Omega(n/t)$ for any t such that $\sqrt{n} \leq t \leq n/4$ and n mod $t = 0$.*

Proof. Consider a homogeneous informed population of t individuals with some informed ranking function. Without loss of generality we can assume that for each n and t this function is represented as a table $s_{n,t}(i,j)$, where the value $s_{n,t}(i,j)$ is the value of the ranking function in a location with i adjacent individuals and j adjacent empty positions. We will show that for any assignment of the values $s_{n,t}(0,2)$, $s_{n,t}(1,1)$ and $s_{n,t}(0,(n/t)-1)$, there are graphs of n nodes for which the price of anarchy is $\Omega(n/t)$.

We proceed by cases. First suppose that $s_{n,t}(0,2) \geq s_{n,t}(1,1)$. Then, placing the individuals on a ring of n nodes such that they are at distance at least 2 each other yields an equilibrium of zero social welfare (i.e., the price of anarchy is infinite).

Therefore we can assume that $s_{n,t}(1,1) > s_{n,t}(0,2)$. Suppose $s_{n,t}(0,(n/t)-1) \geq s_{n,t}(1,1) > s_{n,t}(0,2)$. Then, consider the graph on the right of Fig. 5.2 with $k = n/t$. Note that every node in the graph has either degree $(n/t)-1$ or 2. Thus, placing the individuals onto the nodes of degree $(n/t)-1$ is an equilibrium with infinite price of anarchy.

The remaining case is $s_{n,t}(1,1) > \max(s_{n,t}(0,2), s_{n,t}(0,(n/t)-1))$. Consider the graph on the left of Fig. 5.2 with[11] $N = n - 3t$ and $k = n/t$. Now if we place the t individuals on the ring in groups of two at distance two each other (as shown in the figure) we obtain an equilibrium. The value of this placement is $t/2$, while the optimum is obtained by placing the individuals in the $(n/t)-1$ regular graph which yields a value of $(n-t)/2-1$. Therefore the price of anarchy in this case is $(n-t-2)/t = \Omega(n/t)$. □

[11] For $t \geq \sqrt{n}$ it is always possible to construct a $(k-1)$-regular graph with size t and degree $n/t-1$.

We close this section by analyzing the price of stability for the informed population of parameter λ. For any $0 < \lambda < 1/4$, the price of stability is a constant. Moreover, by tuning the parameter λ, we can make the price arbitrarily close to one.

Theorem 58. *For every constant $\epsilon > 0$, there exists a constant $\lambda = \lambda(\epsilon) > 0$ such that, for any connected graph $G = (V, E)$, the homogeneous informed population with parameter λ achieves price of stability of at most $1 + \epsilon$.*

Proof. Set $\lambda = \frac{\epsilon}{4(1+\epsilon)} < \frac{1}{4}$ and consider any connected graph G of n nodes. Let B be a densest t-size subgraph of G. Place the individuals on the nodes in B and let them move according to the best-response dynamics until they achieve an equilibrium. Let $H \subseteq V$ be the set of nodes occupied by the population in this equilibrium. Lemma 53 shows that the ranking function R_β strictly increases after each step of the best response dynamics, therefore $R_\beta(H) \geq R_\beta(B)$. Now we have:

$$\begin{aligned} E_H &\geq (1-4\lambda)R_\beta(H) \geq (1-4\lambda)R_\beta(B) \\ &\geq (1-4\lambda)E_B. \end{aligned}$$

As such, the price of stability is $E_B/E_H \leq 1/(1-4\lambda) = 1+\epsilon$. □

The proof of Theorem 58 shows that $\lambda = \Theta(\epsilon)$. This observation and Theorem 56 imply that, for any $\epsilon = \epsilon_n\omega(1/n)$, the informed population achieves price of stability at most $1 + \epsilon_n$ and price of anarchy $o(n)$. This result is tightly complemented by Theorem 52 in section 5.5.2.

5.7 Conclusions

We have proved that in aggregation games, populations with diverse strategies achieve more efficient equilibria than homogeneous populations. Somewhat similarly, a number of recent results [SW09, BBM09, BBM10] circumvent high price of anarchy by considering mixtures of strategies, in the sense that some players might follow a "globally optimal" behavior. These results together with ours open new avenues for future research: how does the number of different strategies affects the price of anarchy? Is there a connection between diverse strategies and the quality of equilibria for more general classes of games? Do mixed strategies improve significantly the price of anarchy?

Our work also suggests that game theory can be a useful tool in analyzing a wider class of dynamic systems relevant to aggregation. Extending our analysis to segregation might lead to a better theoretical understanding of the seminal work of Schelling [Sch71, Sch78]. Finally, there are a number of possible extensions to aggregation games. For example one can consider aggregation of individuals of varying popularity or aggregation games over weighted graphs.

Chapter 5, in full, is a reprint of the paper "The Effects of Diversity in Aggregation Games" co-authored with Petros Mol and Panagiotis Voulgaris published in the proceedings of the 2nd Symposium on Innovations in Computer Science, ICS 2011 [MVV11]. The dissertation author was the primary investigator and author of this paper.

Chapter 6

The Secretary Problem

The secretary problem [Dyn63, Fer89] is a perfect example of online decision-making under uncertainty. The setting is humble: candidates for a secretary position arrive online in a random order and the goal is to choose the best candidate, with the constraint that no past decision can be reverted. The optimal algorithm is to skip the first $1/e$ fraction of the candidates and to choose the next arriving candidate who is the best seen so far; this algorithm yields a success probability of $1/e$. The secretary problem has a rich history dating back at least a century, and is a frequent object of study even to this day. See the survey article by Ferguson [Fer89] for an excellent historical perspective of the secretary problem.

Implicit in the classical setting is the assumption that there is a total order on the candidates, but this assumption rarely holds in real life since candidates often have incomparable attributes. This leads to the natural *poset secretary problem*: if the elements of the permutation (candidates) are only partially ordered, how to maximize the probability of returning a maximal element in the poset? Note that the incomparable elements present the main challenge: many simple modifications of the total order algorithm to handle the incomparable elements can be shown to have vanishing success probabilities.

Secretary problems have recently been shown to lie at the core of online auction and mechanism design problems [BIKK08]. For instance, Hajiaghayi, Kleinberg, and Parkes [HKP04] showed how to convert the classic secretary problem into a group strategy-proof mechanism for the online single item auction. The algorithm we present can be adapted in a similar fashion to a setting where the seller has a multidimensional utility function that does not lead to a total ordering on the bidders. The bidders arrive with potentially incomparable bids and the goal is to sell the item at a no-regret price, i.e., to select a bidder who is not dominated by any of the others.

The poset secretary problem was first studied by Preater [Pre99], who proposed an algorithm with a success probability of 1/8. Recently, Georgiou et al. [GKMN08] improved this bound to 1/4; they also showed that this bound is tight for Preater's algorithm. These algorithms suffer from two major drawbacks. First, the success probability does not match the classical bound when the poset is a total order. Second, these bounds do not improve with the number of maximal elements in the poset, which is undesirable since the problem should only become easier as the number of solutions grows.

6.1 Results

In this chapter we study the secretary problem in the partial order setting. We assume that we know k, the number of maximal elements in the poset. Our algorithms take on the standard form with one subtle difference. As before, we examine all of the elements up to a threshold and then consider the first undominated element. We select this element only if the poset at the time has at most k maximal elements. The latter condition may make us pass on a maximal element early in the sequence, but we will never pass on the last maximal element. We show that for a judicious choice of the threshold (that depends on k) our algorithm succeeds with probability roughly $k^{-\frac{k}{k-1}}\left(\left(1+\log k^{1/(k-1)}\right)^k - 1\right)$; see Theorem 61 for a precise statement. This quantity recovers the $1/e$ bound in the limit as $k \to 1$, but quickly surpasses it, reaching 0.47 at $k = 2$ and 0.52 at $k = 3$. We show an almost matching upper bound of $k^{-1/(k-1)} + o(1)$ (Theorem 67), showing that no algorithm succeeds with probability better than 0.5 for $k = 2$ and better than 0.57 for $k = 3$. Figure 6.1 shows these bounds. Closing the gap between the two remains an interesting open problem.

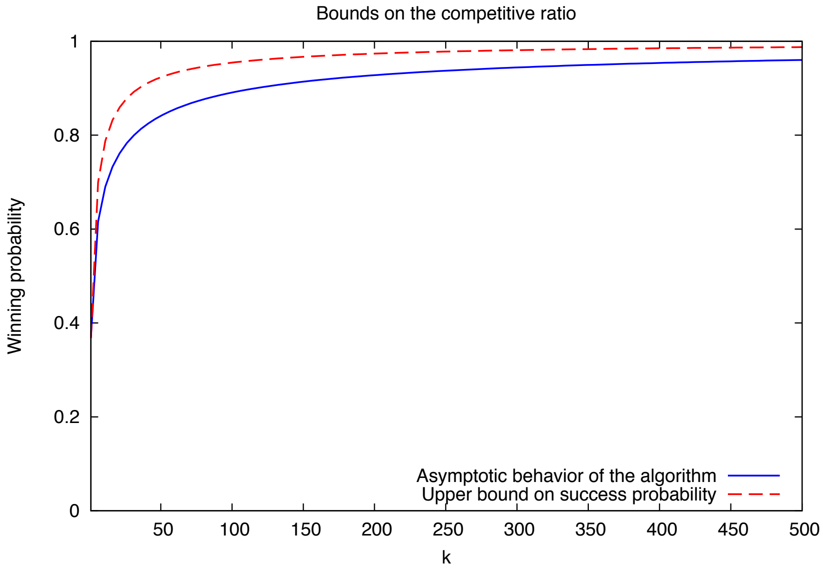

Figure 6.1: A visualization of the upper and lower bounds for the poset secretary problem as a function of the number of maximal elements in the poset.

On the technical side, we proceed as follows. To analyze the algorithm, we concentrate on the probability of the algorithm reaching the last maximal element. We introduce the concept of a *blocking set* and show that if the blocking set occurs early in a random permutation of elements, then the second condition on accepting an element (requiring that the number of maximal elements in the induced poset be at most k) prevents the algorithm from returning a suboptimal element. We then proceed to construct such permutations by starting from the end and increasing the suffix so as to keep the blocking sets early in the sequence.

For an upper bound on the success probability of any algorithm, we use a linear programming approach introduced in the work of Buchbinder, Jain, and Singh [BJS10]. We use this approach to consider the very specific poset of k disjoint total orders, each on ${}^{n}/_{k}$ elements, and show that no algorithm has a good competitive ratio on this poset. To that end, we construct a linear program whose value upper bounds the probability of success of any algorithm on this poset. We then present a feasible solution to the corresponding dual, thus establishing a bound on the success probability of any algorithm for this problem. It is worth to note that this bound holds even if the algorithm knows that the poset under consideration consists of k disjoint total orders.

6.2 Related work

Independently and concurrently with our work, Freij and Wästlund [FW10] recently proposed an algorithm for the partially ordered secretary problem and claimed a bound of $1/e$ on its success probability. Their algorithm works as follows. Assign a score uniformly at random in $[0, 1]$ as elements arrive, skip the first $1/e$ fraction of elements, and pick the first element which is the greedy maximum of the poset seen so far including this element. Here, the greedy maximum of a poset with weighted elements is defined inductively: it is the lowest weight element if it is the maximal or it is the greedy maximum of the sub-poset induced by the elements that are bigger than the lowest weight element. Their algorithm has the same downside as that of [GKMN08, Pre99], namely, the competitive ratio does not increase with the number of maximal elements and remains bounded by $1/e$.

The poset secretary problem has been previously considered for specific cases of posets. For example, Morayne [Mor98] and Kubicki et al. [KLM02] present an optimal stopping time for the case of the complete binary tree. Gnedin [Gne92] explored other specific poset structures. Generally, Bruss [Bru00] defined a way (known as the Odds algorithm) to compute optimal stopping rules for any last-success problem, which applies to the classical version of the secretary problem as well as to the case of any known poset with a single maximal element. For poset-oblivious algorithms, very recently, Kozik [Koz10] proposed a dynamic threshold algorithm that selects a maximal element of any poset with probability at least $1/4 + \epsilon$ (for some small $\epsilon > 0$), therefore beating the $1/4$ bound of Georgiou et al. [GKMN08].

Other variants of secretary problems have been previously applied to the online auction setting. For example, designing mechanisms to maximize some function of the top k elements [AMW01, BDG$^+$09, Kle05] or some function on the accepted set of elements, such as online submodular function maximization [GRST10, MHB10], finding the heaviest weight independent set of a matroid [BIK07, Sot11], etc. See [BIKK08] for a survey of some of these results.

6.3 Preliminaries

Let U be a universe of n elements. A *poset* $\mathcal{P} \subseteq U^2$ is a binary relation that is reflexive, anti-symmetric, and transitive. We use $a \prec_{\mathcal{P}} b$ to denote $(a, b) \in \mathcal{P}$ and use $a \|_{\mathcal{P}} b$ to denote $(a, b) \notin \mathcal{P} \wedge (b, a) \notin \mathcal{P}$, i.e., a and b are incomparable. A *linear extension* of $\mathcal{P}$ is a permutation π on U such that $a \prec_{\mathcal{P}} b \implies \pi^{-1}(a) < \pi^{-1}(b)$.

An element a is *maximal* (aka a *secretary*) if there is no element b such that $a \prec_{\mathcal{P}} b$. Let $\max \mathcal{P}$ be the set of all secretaries of $\mathcal{P}$ and let $k = |\max \mathcal{P}|$, the number of secretaries.

We denote by $S_i = \bigcup_{j<i}\{\pi(j)\}$ the set of elements preceding i in the permutation.

Given $S \subseteq U$, let $\mathcal{P}|_S = \mathcal{P} \cap S^2$, the poset obtained from $\mathcal{P}$ by using the elements only in S.

Definition 59 (Pareto frontier)**.** *Given a poset $\mathcal{P}$ and a subset $S \subseteq U$, the* Pareto frontier *$\mathcal{F}_{\mathcal{P}}(S)$ is defined to be* $\max \mathcal{P}|_S$.

6.4 Algorithm and Analysis

Let $\mathcal{P}$ be the given poset. We assume that the algorithm is given $k = |\max \mathcal{P}|$, the number of secretaries. Our algorithm proceeds in a way similar to the algorithm in the total order setting. It examines all of the elements before a threshold τ. An element a arriving after the threshold is returned if two conditions are met. First, the element must be *undominated*, i.e., $a \in \mathcal{F}_{\mathcal{P}}(S)$, where S is the set of all of elements seen thus far; since any dominated element cannot be maximal, this is without loss of generality. Second, the total size of $\mathcal{F}_{\mathcal{P}}(S)$ is at most k. While the second condition may lead the algorithm to pass on a maximal element, it will never pass on the last maximal element in the permutation.

We will denote by π the order in which the elements arrive. To describe

Algorithm 6.1: Secretary (π, k).

```
$S = \{\pi(1), \ldots, \pi(\tau_k)\}$
for $i = \tau_k + 1 \ldots n$ do
    $a = \pi(i)$
    $S \leftarrow S \cup \{a\}$
    if $a \in \mathcal{F}_{\mathcal{P}}(S) \wedge |\mathcal{F}_{\mathcal{P}}(S)| \leq k$ then return $a$
```

the algorithm, let τ_k be the *stopping threshold*:

$$\tau_k = \begin{cases} n/e, & k = 1, \\ n/k^{\frac{1}{k-1}}, & k > 1. \end{cases}$$

Note that $\lim_{k \to 1^+} \tau_k = n/e$.

6.4.1 Warmup: Analysis for a single secretary

Theorem 60. *For any poset $\mathcal{P}$ with $k = 1$, Algorithm 6.1 succeeds with probability at least $1/e$.*

Proof. Let $\mathcal{P}_L$ be an arbitrary linear extension of $\mathcal{P}$; by definition, if $a \prec_{\mathcal{P}_L} b$, then either $a \prec_{\mathcal{P}} b$ or $a \|_{\mathcal{P}} b$. Now, we compare the performance of Algorithm 6.1 on $\mathcal{P}$ and $\mathcal{P}_L$. Consider any permutation π such that the algorithm outputs the secretary when run on $\mathcal{P}_L$. We claim that the algorithm outputs the secretary when run on $\mathcal{P}$ as well. This will complete the proof since Algorithm 6.1 on a linear order (i.e., $\mathcal{P}_L$) is the optimal algorithm for the classical secretary problem, and therefore succeeds with probability at least $1/e$.

To prove the claim, we only need to show that the algorithm does not output any element before encountering the secretary. Let i^* be the position in π where the secretary occurs and consider any position $i \in (\tau_1, i^*)$. It must be the case that $\pi(j) \succ_{\mathcal{P}_L} \pi(i)$ for some $j < i$, since otherwise the algorithm would have output $\pi(i)$ when run on $\mathcal{P}_L$. Therefore, either $\pi(j) \succ_{\mathcal{P}} \pi(i)$ or $\pi(j) \|_{\mathcal{P}} \pi(i)$. In both cases, the element $\pi(i)$ is not output when the algorithm runs on $\mathcal{P}$: indeed, in the former case, $\pi(i) \notin \mathcal{F}_{\mathcal{P}}(S_i)$ and in the latter $|\mathcal{F}_{\mathcal{P}}(S_i)| \geq 2 > k$, where $S_i = \bigcup_{j \leq i} \{\pi(j)\}$. □

6.4.2 Analysis for general posets

In this section we show that the algorithm succeeds with increasing probability as k increases.

Theorem 61. *For any poset $\mathcal{P}$ with k maximal elements, Algorithm 6.1 succeeds with probability at least*

$$\frac{\binom{\tau_k}{k}}{\binom{n}{k}} \cdot \left(\left(1 + \log \frac{n-k}{\tau_k}\right)^k - 1 \right).$$

Before proceeding further, let us briefly interpret the above bound. Let $k = \epsilon n$, and consider a slightly different threshold $\tau_k' = (1-\epsilon)\tau_k = (1-\epsilon)nk^{-\frac{1}{k-1}}$. Note that holding k fixed and letting $n \to \infty$, we have $\epsilon \to 0$.

Then, we can bound $\binom{\tau_k'}{k}/\binom{n}{k}$ as:

$$\begin{aligned}\frac{\prod_{i=0}^{k}(1-\epsilon)nk^{-\frac{1}{k-1}}}{\prod_{i=0}^{k}(n-i)} &> \left(\frac{(1-\epsilon)k^{-\frac{1}{k-1}} - \epsilon}{(1-\epsilon)}\right)^k \\ &\geq k^{-\frac{k}{k-1}} - O(\epsilon).\end{aligned}$$

And,

$$\begin{aligned}\left(1 + \log \frac{n-k}{\tau_k'}\right)^k &= \left(1 + \log \frac{(1-\epsilon)nk^{\frac{1}{k-1}}}{(1-\epsilon)n}\right)^k \\ &= \left(1 + \frac{1}{k-1}\log k\right)^k \\ &\geq k\left(1 - \frac{2\log^2 k}{k}\right) \\ &\geq k(1 - o_k(1)).\end{aligned}$$

Combining the two, we obtain that the probability of winning is at least

$$\left(k^{-\frac{k}{k-1}} - O(\epsilon)\right) k(1 - o_k(1)) = (1 - o_k(1))k^{-\frac{1}{k-1}} - O(\epsilon).$$

As we will show in Section 6.5, the term $k^{-\frac{1}{k-1}}$ is tight and thus the maximum difference between the lower bound and the upper bound approaches 0 for large k (Figure 6.1).

Proof of Theorem 61

To prove the theorem, we will describe the set of permutations on which the algorithm is guaranteed to succeed. In particular, we focus on the probability that the algorithm does not return an element before reaching the last secretary in the permutation. Observe that the algorithm will never pass on the last secretary: it will surely be in $\mathcal{F}_{\mathcal{P}}(S)$, and at that point $\mathcal{F}_{\mathcal{P}}(S) = k$. Obviously, the algorithm will fail if all of the secretaries come before the threshold. A harder to analyze failure mode is that of returning a *faux*-secretary: an element that looks like a maximal element before reaching the actual secretary that dominates it. A way to avoid it is to insist that either $\mathcal{F}_{\mathcal{P}}$ is of size at least $k+1$ before the last secretary is reached or that the maximal element comes before any of the potential faux-secretaries (the latter is exactly the analysis in the $k = 1$ case).

We begin by describing the permutations on which Algorithm 6.1 will succeed. We first give few definitions that we use in the proof.

Let $\tau = \tau_k$. Fix any $k+1$ special positions in the permutation $1 \leq \ell_0 \leq \cdots \leq \ell_k \leq n$, such that for some $0 \leq i^* < k$ we have $\ell_{i^*} = \tau$, and the other positions are all distinct from each other. We define the set $P = P_{\ell_0,\ldots,\ell_k}$ of all permutations such that the positions ℓ_i with $i \neq i^*$ are occupied by the secretaries in any order.

For a suffix t_i of elements from position ℓ_i to n, we define $P(t_i)$ be the set of all permutations in P that have t_i as a suffix. For a set of suffixes T, we let $P(T) = \bigcup_{t \in T} P(t)$.

We will now inductively define a set T_{i^*} of suffixes such that Algorithm 6.1 returns a maximal element in all permutations in $P(T_{i^*})$. To begin, let T_k be the set of all suffixes from ℓ_k to n; thus we have that $P(T_k)$ contains all the permutations in P. Inductively, define T_i, for $i = k-1, \ldots, i^*$, in the following way. Let the $\mathcal{F}_{\mathcal{P}}\left(U \setminus t_{i+1}\right)$ be Pareto frontier of the elements that are *not* in t_{i+1}, and define $\mathcal{G}(t_{i+1})$ to be the set of non-secretary elements of $\mathcal{F}_{\mathcal{P}}\left(U \setminus t_{i+1}\right)$.

Now let $B(t_{i+1})$ be any subset of $\mathcal{G}(t_{i+1})$ of $\min\{k-i, |\mathcal{G}(t_{i+1})|\}$ elements; we call the set $B(t_{i+1})$ a *blocking set*. Note that in a permutation π where all of the elements in $B(t_{i+1})$ come before ℓ_i, Algorithm 6.1 cannot terminate with any element between ℓ_i and ℓ_{i+1}. In this case, we say that $B(t_{i+1})$ is a *good* blocking set in π. In order to find a lower bound on the number of winning permutation for a fixed position of i^*, we can bound the number of permutations where $B(t_{j+1})$ is a good blocking set for every $i^* \leq j < k$. To this end, let $A(t_{i+1})$ be the set of all suffixes from ℓ_i to n that agree with t_{i+1} and that do not contain elements from $B(t_{i+1})$. Let $T_i = \bigcup_{t_{i+1} \in T_{i+1}} A(t_{i+1})$.

Lemma 62. *Algorithm 6.1 returns a maximal element on all permutations in $P(T_{i^*})$.*

Proof. Suppose not and consider any permutation in $P(T_{i^*})$ where the algorithm fails. Suppose the returned element is in the interval between ℓ_i and ℓ_{i+1} for some $i^* \leq i \leq k-1$. Then, this permutation has to be in $P(T_i)$ (since $P(T_i) \subseteq P(T_{i+1})$). But by definition of T_i, if $\mathcal{G}$ is the Pareto frontier of the elements before ℓ_{i+1}, either all of $\mathcal{G}$ or a subset of at least $k-i$ elements of $\mathcal{G}$ comes before ℓ_i. Either way, in the positions $t_i, \ldots, t_{i+1}$, the Pareto frontier is composed of at least i secretaries and the good blocking set B_{i+1}. Thus the **if** statement in step 5 of Algorithm 6.1 avoids that an element is returned in the interval between ℓ_i and ℓ_{i+1}. So we have a contradiction.

□

Suppose there were j secretaries that came after the threshold. In this case, we can look at the fraction of the permutations whose suffix agrees with T_{k-j}. Let

$$\gamma(j) = \frac{|P(T_{k-j})|}{|P|}.$$

Note that γ is implicitly a function of $\ell_k, \ldots, \ell_{i^*}$. We begin by bounding $\gamma(j)$ from below. Let

$$\gamma'(j) = \prod_{i=k-j}^{k-1} \left(\prod_{w=0}^{k-i-1} \left(\frac{\ell_i - i - w}{\ell_{i+1} - (i+1) - w} \right) \right).$$

Lemma 63. $\gamma(j) \geq \gamma'(j)$.

Proof. By definition, $P(T_i) \subseteq P(T_{i+1})$ and hence

$$\begin{aligned} |P(T_i)| &= |P(T_{i+1})| \cdot \big(\text{fraction of permutation in } P(T_{i+1}) \text{ with good } B_{t_{i+1}}\big) \\ &\geq |P(T_{i+1})| \cdot \big(\text{fraction of permutation in } P(T_{i+1}) \text{ with } k \text{ elements of} \\ &\qquad\qquad \mathcal{G}(t_{i+1}) \text{ before } k-i-1\big) \\ &\geq |P(T_{i+1})| \prod_{w=0}^{k-i-1} \left(\frac{\ell_i - i - w}{\ell_{i+1} - (i+1) - w} \right). \end{aligned}$$

Therefore, we can conclude that

$$|P(T_{i^*})| \geq |P| \prod_{i=i^*}^{k-1} \left(\prod_{w=0}^{k-i-1} \left(\frac{\ell_i - i - w}{\ell_{i+1} - (i+1) - w} \right) \right).$$

And,

$$\begin{aligned} \gamma(k - i^*) &= \frac{|P(T_{i^*})|}{|P|} \\ &\geq \prod_{i=i^*}^{k-1} \left(\prod_{w=0}^{k-i-1} \left(\frac{\ell_i - i - w}{\ell_{i+1} - (i+1) - w} \right) \right) \\ &= \gamma'(k - i^*). \end{aligned}$$

□

We first show that $\gamma'(r)$ can be rewritten in a more convenient way.

Lemma 64.

$$\gamma'(r) = \frac{(\ell_{k-r} - (k-r))!}{(\ell_{k-r} - k)!} \prod_{s=0}^{r-1} \frac{1}{\ell_{k-s} - k}.$$

Proof. We prove the statement by induction. By inspection, the equality holds for

$r = 1$. Now suppose it holds up to a certain $1 \leq r < k$. Then, we have

$$\begin{aligned}
\gamma'(r+1) = & \prod_{i=k-r}^{k-1} \left(\prod_{w=0}^{k-i-1} \left(\frac{\ell_i - i - w}{\ell_{i+1} - (i+1) - w} \right) \right) \\
= \quad & \gamma'(r) \cdot \prod_{w=0}^{r} \left(\frac{\ell_{k-(r+1)} - (k-(r+1)) - w}{\ell_{k-r} - (k-r) - w} \right) \\
= \quad & \frac{(\ell_{k-r} - (k-r))!}{(\ell_{k-r} - k)!} \left(\prod_{s=0}^{r-1} \frac{1}{\ell_{k-s} - k} \right) \\
& \cdot \frac{(\ell_{k-r} - (k-r) - (r+1))!}{(\ell_{k-r} - (k-r))!} \\
& \cdot \frac{(\ell_{k-(r+1)} - (k-(r+1)))!}{(\ell_{k-r+1} - (k-(r+1)) - (r+1))!} \\
= \quad & \frac{(\ell_{k-(r+1)} - (k-(r+1)))!}{(\ell_{k-r+1} - k)!} \cdot \frac{1}{\ell_{k-r} - k} \left(\prod_{s=0}^{r-1} \frac{1}{\ell_{k-s} - k} \right),
\end{aligned}$$

which concludes the proof. □

Next, we obtain an analytical bound that will be useful later.

Lemma 65.

$$\sum_{\substack{\ell_k, \ldots, \ell_{k-j+1}: \\ n \geq \ell_k > \cdots > \ell_{k-j} = \tau}} \prod_{s=0}^{j-1} \frac{s+1}{\ell_{k-s} - k} \geq \log^j \frac{n-k}{\ell_{k-j} - (k-j)}.$$

Proof. For convenience, define ϕ as follows,

$$\phi(r) = \begin{cases} 1, & r = k+1 \\ \displaystyle\sum_{\ell_r : \ell_r = \ell_{r-1}+1}^{n-(k-r)} \frac{k-r+1}{\ell_r - k} \phi(r+1), & r \leq k. \end{cases}$$

It is easy to see that

$$\sum_{\substack{\ell_k, \ldots, \ell_{k-j+1}: \\ n \geq \ell_k > \cdots > \ell_{k-j} = \tau}} \prod_{s=0}^{j-1} \frac{s+1}{\ell_{k-s} - k} = \phi(k - (j-1)).$$

We now provide a lower bound on $\phi(r)$.

$$\phi(r) \geq \log^{k-r+1} \frac{n-k}{\ell_{r-1} - (r-1)}.$$

We proceed by backward induction on r. For $r = k+1$, the claim holds trivially. Suppose the claim holds down to a certain $r+1$. Then,

$$\begin{aligned}
\phi(r) &= \sum_{\ell_r=\ell_{r-1}+1}^{n-(k-r)} \frac{k-r+1}{\ell_r - k}\phi(r+1) \\
&= \sum_{\ell_r=\ell_{r-1}+1}^{n-(k-r)} \frac{k-r+1}{\ell_r - k}\log^{k-r}\frac{n-k}{\ell_r - r} \\
&\geq \sum_{\ell_r=\ell_{r-1}+1}^{n-(k-r)} \frac{k-r+1}{\ell_r - r}\log^{k-r}\frac{n-k}{\ell_r - r} \\
&\geq \int_{\ell_{r-1}+1}^{n-(k-r)} \frac{k-r+1}{x-r}\log^{k-r}\frac{n-k}{x-r} \\
&= -\log^{k-r+1}\frac{n-k}{x-r}\bigg|_{x=\ell_{r-1}+1}^{n-(k-r)} \\
&= \log^{k-r+1}\frac{n-k}{\ell_{r-1}-(r-1)}.
\end{aligned}$$

□

Now we are ready to put all of the pieces together. To count the total number of permutations on which the algorithm succeeds, we begin by conditioning on the number of secretaries that come after the specified threshold, τ. Let E_j be the event such that there are exactly $j \geq 1$ fixed secretaries after the threshold $\ell_{k-j} = \tau$ and let WIN be the event of the algorithm returning a maximal element.

Lemma 66.

$$\Pr[\textsc{win}|E_j] \geq \frac{(\tau-(k-j))!}{(\tau-k)!}\frac{(n-\tau-j)!}{(n-\tau)!}\log^j\left(\frac{n-k}{\tau-(k-j)}\right).$$

Proof. We enumerate over all permutations that have j maximal elements after the threshold. Since $\gamma(j)$ depends only on the position and not on the order of these elements, we have:

$$\Pr[\textsc{win}|E_j] = j! \sum_{\substack{\ell_k,\ldots,\ell_{k-j+1}: \\ n\geq\ell_k>\cdots>\ell_{k-j}=\tau}} \left(\frac{1}{n-\tau}\cdots\frac{1}{n-\tau-(j-1)}\right)\gamma(j).$$

Since $\ell_{k-j} = \tau$, applying Lemma 63, Lemma 64, and Lemma 65 completes the proof.

$$\begin{aligned}
\Pr[\text{WIN}|E_j] &\geq \frac{(\tau-(k-j))!}{(\tau-k)!}\frac{(n-\tau-j)!}{(n-\tau)!} j! \sum_{\substack{\ell_k,\ldots,\ell_{k-j+1}: \\ n\geq\ell_k>\cdots>\ell_{k-j}=\tau}} \prod_{s=0}^{j-1} \frac{1}{\ell_{k-s}-k} \\
&= \frac{(\tau-(k-j))!}{(\tau-k)!}\frac{(n-\tau-j)!}{(n-\tau)!} \sum_{\substack{\ell_k,\cdots,\ell_{k-j+1}: \\ n\geq\ell_k>\cdots>\ell_{k-j}=\tau}} \prod_{s=0}^{j-1} \frac{s+1}{\ell_{k-s}-k} \\
&\geq \frac{(\tau-(k-j))!}{(\tau-k)!}\frac{(n-\tau-j)!}{(n-\tau)!} \log^j\left(\frac{n-k}{\tau-(k-j)}\right).
\end{aligned}$$

□

Proof of Theorem 61. Finally, we can remove the conditioning in Lemma 66 to prove an overall bound on the success probability of the algorithm.

$$\begin{aligned}
\Pr[E_j] = &\binom{k}{j}\left(\frac{\tau}{n}\cdots\frac{\tau-(k-j-1)}{n-(k-j-1)}\right) \\
&\cdot\left(\frac{n-\tau}{n-(k-j)}\cdots\frac{n-\tau-(j-1)}{n-(k-1)}\right) \\
= &\binom{k}{j}\frac{\tau!}{(\tau-(k-j))!}\cdot\frac{(n-\tau)!}{(n-\tau-j)!}\cdot\frac{(n-k)!}{n!}.
\end{aligned}$$

Now, using Lemma 66, we have that

$$\begin{aligned}
\Pr[\text{WIN}] &= \sum_{j=1}^{k}\Pr[\text{WIN}|E_j]\Pr[E_j] \\
&= \frac{\tau!}{(\tau-k)!}\cdot\frac{(n-k)!}{n!}\cdot\sum_{j=1}^{k}\binom{k}{j}\log^j\frac{n-k}{\tau-(k-j)} \\
&\geq \frac{\tau!}{(\tau-k)!}\cdot\frac{(n-k)!}{n!}\cdot\sum_{j=1}^{k}\binom{k}{j}\log^j\frac{n-k}{\tau} \\
&= \frac{\tau!}{(\tau-k)!}\cdot\frac{(n-k)!}{n!}\cdot\left(\left(1+\log\frac{n-k}{\tau}\right)^k-1\right).
\end{aligned}$$

□

6.5 Upper bounds on success

In this section we prove an upper bound on the success probability of any algorithm for the poset secretary problem. For $k = 1$, it is well-known that no algorithm can succeed with probability more than $1/e$. Here we explore how the bound grows with k. Our main result is the following:

Theorem 67. *Let $2 \leq k = o(\sqrt{n})$. For any poset $\mathcal{P}$ with k maximal elements, every algorithm has success probability at most $k^{-\frac{1}{k-1}} + o(1)$.*

To prove this result we will analyze the performance of any algorithm on a specific poset $\mathcal{P}_k$. Let $\mathcal{L}$ be a total order on n/k elements; we will call such a poset a *line*. We define $\mathcal{P}_k$ to be the poset consisting of k disjoint lines: $\mathcal{P}_k = \{\mathcal{L}_1, \ldots, \mathcal{L}_k\}$.

Our strategy is to write down a linear program whose value is an upper bound on the success probability of any algorithm. We will then analyze the dual formulation and derive a feasible solution for it, which will serve as the bound in Theorem 67.

We begin by restricting the class of algorithms and the class of permutations we consider. A *τ-threshold* algorithm is one that never returns any of the first τ elements. Recall that $S_i = \cup_{j<i}\{\pi(j)\}$ denotes the set of elements preceding i in the permutation and $\mathcal{F}(S_i)$ denotes the set of maximal elements of S_i. We insist that the algorithms we consider are *sane*, i.e., they never knowingly return a dominated element; formally, if the element at position i is returned by the algorithm, then $\pi(i) \in \mathcal{F}(S_{i+1})$.

We also restrict the permutations under consideration. A permutation π is called *τ-covering* if the following two conditions hold:

1. $\mathcal{F}(S_\tau) \cap \mathcal{F}_\mathcal{P} = \emptyset$, i.e., π has no maximal elements in the first τ positions; and
2. for any $1 \leq j \leq k$, $\mathcal{F}(S_\tau) \cap \mathcal{L}_j \neq \emptyset$, i.e., at least one descendant of each maximal element occurs among the first τ elements.

These restrictions on the algorithm and the permutations do not change the success probability substantially.

$$
\begin{array}{llcll}
\max\limits_{q_1,\dots,q_n} \dfrac{k}{n}\sum\limits_{i=1}^{n} q_i & & \qquad & \min\limits_{x_1,\dots,x_n} \sum\limits_{i=1}^{n} x_i & \\
q_i + \sum\limits_{j=1}^{i-1} \dfrac{k}{j} q_j \le 1, & 1 \le i \le n & & x_i + \dfrac{k}{i} \sum\limits_{j=i+1}^{n} x_j \ge \dfrac{k}{n}, & 1 \le i \le n \\
q_i \ge 0, & 1 \le i \le n & & x_i \ge 0, & 1 \le i \le n
\end{array}
$$

Figure 6.2: Linear program (left) and its dual (right). These are used in the proof of the upper bound.

Lemma 68. *Consider any algorithm $\mathcal{A}$ that succeeds with probability ρ. Then $\mathcal{A}$ succeeds with probability at least $\rho - o(1)$ on all $(2k \log n)$-covering permutations. Furthermore, when run on $(2k \log n)$-covering permutations, $\mathcal{A}$ is a sane and $(2k \log n)$-threshold algorithm without loss of generality.*

Proof. To prove the first claim, observe that $(2k \log n)$-covering permutations constitute an

$$O\left(\left(1 - \frac{1}{k}\right)^{2k \log n} + \left(1 - \frac{2k \log n}{n}\right)^{k}\right) = o(1)$$

fraction of all of the permutations, when $k = o(\sqrt{n})$. Moreover, on these permutations, any algorithm returning one of the first $2k \log n$ elements is guaranteed to fail, therefore we can assume that the algorithm is a $(2k \log n)$-threshold algorithm without loss of generality. □

For the remainder of the proof we therefore assume that the algorithms under consideration are sane and $(2k \log n)$-threshold. We proceed by writing down a linear program that encodes the success probability of any algorithm on $\mathcal{P}_k$.

Lemma 69. *Consider any optimal solution of the linear program in Fig. 6.2 and let v be its value. Then, any sane $(2k \log n)$-threshold algorithm $\mathcal{A}$, has success probability at most v on the poset $\mathcal{P}_k$.*

Proof. Let $p_i = \Pr[\mathcal{A} \text{ returns } \pi(i)]$ denote the probability that $\mathcal{A}$ returns the ith element of the permutation[1]. Similarly, let $q_i = \Pr[\mathcal{A} \text{ returns } \pi(i) | \pi(i) \in \mathcal{F}(S_{i+1})]$.

[1] The probability is over both the permutations and the coins of the algorithm.

Note that since $\mathcal{A}$ is sane, $\mathcal{A}$ never returns $\pi(i)$ if $\pi(i) \notin \mathcal{F}(S_{i+1})$. Therefore we can write $p_i = q_i \Pr[\pi(i) \in \mathcal{F}(S_{i+1})]$. Moreover, $\mathcal{A}$ returns an element $\pi(i)$ only if it discards all of the elements in positions $j < i$. Thus we can write:

$$q_i \leq 1 - \sum_{j<i} p_j = 1 - \sum_{j<i} f_j q_i, \tag{6.1}$$

where $f_j = \Pr[\pi(j) \in \mathcal{F}(S_{j+1})]$.

We can express the probability that $\mathcal{A}$ returns a maximal element as

$$\begin{aligned}\Pr[\mathcal{A} \text{ wins}] &= \sum_{j=1}^{k}\sum_{i=1}^{n} \Pr[\mathcal{A} \text{ returns } \pi(i) | \pi(i) \in \mathcal{F}(\mathcal{L}_j)] \Pr[\pi(i) \in \mathcal{F}(\mathcal{L}_j)] \\ &= \frac{k}{n}\sum_{i=1}^{n} \Pr[\mathcal{A} \text{ returns } \pi(i) | \pi(i) \in \mathcal{F}(S_{i+1})] \\ &= \frac{k}{n}\sum_{i=1}^{n} q_i,\end{aligned}$$

where the second step follows because the algorithm cannot determine whether a maximal element of the poset induced from the first i elements in π is a maximal element of the whole poset or not, and hence the contributions are equal. More formally, we observe that, for every $1 \leq j \leq k$, $\Pr[\mathcal{A} \text{ returns } \pi(i) | \pi(i) \in \mathcal{F}(\mathcal{P}_k) \cap \mathcal{L}_j] = \Pr[\mathcal{A} \text{ returns } \pi(i) |\ \pi(i) \in \mathcal{F}(S_{i+1}) \cap \mathcal{L}_j]$. This follows because for any two permutations π, π' identical up to $i-1$, and with $\pi(i) \in \mathcal{F}_\pi(\mathcal{P}_k) \cap \mathcal{L}_j$ and $\pi'(i) \in \mathcal{F}_{\pi'}(S_{i+1}) \cap \mathcal{L}_j$, we have that the poset induced by the first i elements is exactly the same. Hence, the algorithm's behavior is unchanged (here the subscript on $\mathcal{F}$ denotes the permutation of the elements under consideration).

Finally we show that we correctly captured the constraints on q. Assume that $i > 2k \log n$, and denote by S_τ the set of elements appearing before the threshold. Since for all j, $S_\tau \cap \mathcal{L}_j \neq \emptyset$, the size of the Pareto set at i is exactly k. Therefore $f_i = {}^k\!/_i$. For $i > 2k \log n$, inequality (6.1) implies that $q_i + \sum_{j<i} \frac{k}{j} q_j \leq 1$.The same inequality trivially holds when $i \leq 2k \log n$ since $q_i = 0$ for these elements. □

Next, we focus on the feasible solution to the dual program.

Lemma 70. *There exists a feasible solution to the dual program in Figure 6.2 that has value* $k^{-\frac{1}{k-1}} + o(1)$.

Proof. We define the following feasible solution to the dual program in Figure 6.2: inductively, $x_n = \frac{k}{n}$ and $x_i = \max\{0, \frac{k}{n} - \frac{k}{i}\sum_{j=i+1}^{n} x_j\}$. Note that the value of the objective function for this solution is $\sum_{i=1}^{n} x_i = \sum_{i=T+1}^{n} x_i$, where T is the maximum index such that $x_T = 0$. Consider the sequence a_i inductively defined by $a_n = \frac{k}{n}$ and $a_i = a_{i+1} + \left(\frac{k}{n} - \frac{k}{i}a_{i+1}\right)$. We observe that for any $j \geq T+1$, it holds that $a_j = \sum_{i=j}^{n} x_i$ and that $\frac{k}{n} - \frac{k}{j}a_{T+1} \leq 0$. Specifically, either $a_T < a_{T+1}$ and $a_{T+1} > \cdots > a_n$, or $a_{T-1} < a_T = a_{T+1}$ and $a_{T+1} > \cdots > a_n$.

Note that the problem of computing $\sum_{i=1}^{n} x_i$ now reduces to the problem of finding the last local maximum of the sequence $\{a_i\}_{i=1}^{n}$.

We proceed as follows. First, we introduce the function $s(z)$ defined over the real domain $[1, n]$ by

$$s(z) = \frac{k}{n}\sum_{\ell=0}^{n-z}\prod_{t=1}^{\ell}\left(1 - \frac{k}{z+t-1}\right).$$

Note that $s(i) = a_i$ for every $i \in \{1, \ldots, n\}$. We study $s(z)$ in $[g(n), n]$ with $g(n) = \omega(1)$ and show that it has only one stationary point (a maximum) in this interval at $z^* = (1 \pm o(1))n/k^{1/(k-1)}$. Finally, since $s(z)$ is continuous, we can conclude that $T + 1 = \max\{\lfloor z^* \rfloor, \lceil z^* \rceil\}$.

We have

$$\begin{aligned}
s(z) &= \frac{k}{n}\sum_{\ell=0}^{n-z}\prod_{t=1}^{\ell}\left(1 - \frac{k}{z+t-1}\right) \\
&= \frac{k}{n}\sum_{\ell=0}^{n-z}\exp\left[\sum_{t=1}^{\ell}\log\left(1 - \frac{k}{z+t-1}\right)\right] \\
&\geq \frac{k}{n}\sum_{\ell=0}^{n-z}\exp\left[\int_{t=1}^{\ell}\log\left(1 - \frac{k}{z+t-1}\right)dt\right] \\
&= \frac{k}{n}\sum_{\ell=0}^{n-z}\exp\left[(z+t-1)\log\left(1 - \frac{k}{z+t-1}\right)\right. \\
&\qquad\qquad \left. -k\log(z-1-k+t)\Big|_{t=1}^{\ell}\right] \\
&= \frac{k}{n}\sum_{\ell=0}^{n-z}\frac{\left(1 - \frac{k}{z+\ell-1}\right)^{z+\ell-1}(z-k)^k}{\left(1 - \frac{k}{z}\right)^z(z+\ell-1-k)^k}.
\end{aligned}$$

One can similarly show that

$$s(z) \le \frac{k}{n} \sum_{\ell=0}^{n-z} \frac{\left(1-\frac{k}{z+\ell}\right)^{z+\ell}(z-k)^k}{\left(1-\frac{k}{z}\right)^z (z+\ell-k)^k}.$$

For $z \ge k \log n$, we have that $\left(1-\frac{k}{z+l-1}\right)^{z+l-1} = (1-o(1))e^{-k}$ and also $\left(1-\frac{k}{z}\right)^z = (1-o(1))e^{-k}$, where the $o(1)$ term hides factors going to 0 as $n \to \infty$. So we can write $s(z)$ as

$$\begin{aligned} s(z) \ge (1-o(1))\frac{k}{n}(z-k)^k \sum_{\ell=0}^{n-z} \frac{1}{(z+\ell-1-k)^k} \\ \ge \ (1-o(1))\frac{k}{n}(z-k)^k \int_{\ell=0}^{n-z} \frac{1}{(z+\ell-1-k)^k} d\ell \\ = \ (1-o(1))\frac{k}{n}\frac{1}{k-1}(z-k)^k \\ \left(\frac{1}{(z-k-1)^{k-1}} - \frac{1}{(n-k-1)^{k-1}}\right) \\ \ge \ (1-o(1))\frac{k}{n}\frac{1}{k-1}\left((z-k) - \frac{(z-k)^k}{(n-k-1)^{k-1}}\right). \end{aligned}$$

One can similarly show that

$$s(z) \le (1+o(1))\frac{k}{n}\frac{1}{k-1}\left((z-k) - \frac{(z-k)^k}{(n-k-1)^{k-1}}\right).$$

Taking the derivative and setting it to zero gives a maximum at z^* at $\frac{n}{k^{1/(k-1)}}(1 \pm o(1))$. Define $i^* = \max\{\lfloor z^* \rfloor, \lceil z^* \rceil\}$. By definition of a_i and the maximality of i^*, it must be that $\frac{k}{n} - \frac{k}{i^*}a_{i^*+1} \ge 0$, which implies

$$a_{i^*+1} \le \frac{i^*}{n} \le k^{-1/(k-1)}(1+o(1)).$$

Finally,

$$\sum_{i=1}^{n} x_i = a_{i^*} \le a_{i^*+1} + \frac{k}{n} \le k^{-1/(k-1)} + o(1).$$

This gives a feasible solution to the dual and a bound on its value. □

6.6 Tightness of the algorithm

In Section 6.5 we showed that no algorithm can succeed with probability more than $k^{-\frac{1}{k-1}} - o(1)$ on the poset $\mathcal{P}_k$ consisting of k disjoint total orders. In this section we use a different analysis to show that Algorithm 6.1 achieves this bound on a large family of posets including $\mathcal{P}_k$.

The main idea is to define an event of the algorithm passing on a maximal element after the threshold τ_k and then returning a non-maximal element; call this event PASS. For any $\tau_k < i \leq n$, we define A_i as the event that the first secretary after the threshold occurs at position i and D_i as the event that the algorithm discards the element $\pi(i)$; let $D_{i_1:i_2} = D_{i_1} \wedge \cdots \wedge D_{i_2}$. Now, if WIN denotes the event that the algorithm returns a secretary, then $\textsc{pass} = \bigcup_{i>\tau_k} \textsc{pass}_i$, where $\textsc{pass}_i = D_{\tau_k+1:i} \wedge A_i \wedge \overline{\textsc{win}}$.

We begin by showing that the disjoint events WIN and PASS together account for the vast majority of the outcomes of the algorithms. We then show how to upper bound the probability of PASS on particular posets, leading to a bound on the success probability of the algorithm.

Lemma 71. *It holds that*

$$\Pr[\textsc{win}] + \Pr[\textsc{pass}] \geq \frac{k}{n} \sum_{i=\tau_k+1}^{n} \frac{\binom{\tau_k}{k}}{\binom{i-1}{k}}.$$

Proof. Suppose the first secretary after the threshold is at i. Then the algorithm can win by returning $\pi(i)$ or discarding $\pi(i)$ and then winning later on. In the first case, it must be that $\mathcal{F}(S_{i+1}) \leq k$ while in the other, $\mathcal{F}(S_{i+1}) \geq k+1$. Let F_i denote the event that $|\mathcal{F}(S_{i+1})| \geq k+1$. Then, we have

$$\Pr[\textsc{win}] = \sum_{i>\tau_k} \left(\Pr[D_{\tau_k+1:i-1} \wedge A_i \wedge \overline{F_i}] + \Pr[D_{\tau_k:i-1} \wedge A_i \wedge F_i \wedge \textsc{win}] \right).$$

The second term can be rewritten as

$$\begin{aligned} \Pr[D_{r_k:i-1} \wedge A_i \wedge F_i \wedge \textsc{win}] &= (1 - \Pr[\overline{\textsc{win}} | D_{r_k:i-1} \wedge A_i \wedge F_i]) \\ &\quad \cdot \Pr[D_{r_k:i-1} \wedge A_i \wedge F_i] \\ &= \Pr[D_{r_k:i-1} \wedge A_i \wedge F_i] - \Pr[\textsc{pass}_i]. \end{aligned}$$

Since the events $\textsc{pass}_i$ are disjoint, we have that

$$\Pr[\textsc{win}] + \Pr[\textsc{pass}] = \sum_{i=\tau_k+1}^{n} \Pr[D_{\tau_k:i-1} \wedge A_i].$$

We proceed to bound $\Pr[D_{\tau_k:i-1} \wedge A_i]$. Note that the probability of $\pi(i)$ being a secretary is k/n. Consider any subset B of $\mathcal{F}(S_i)$ that contains all secretaries in $\mathcal{F}(S_i)$ and is of size $\min\{|\mathcal{F}(S_i)|, k\}$. B is a blocking subset: if all elements in B appear before the threshold, no element between $\tau_k + 1$ and $i - 1$ (inclusive) will be accepted. Moreover, this implies that there are no maximal elements between $\tau_k + 1$ and $i - 1$. The lemma follows observing that the probability of this event is at least $\frac{\binom{\tau_k}{\ell}}{\binom{i-1}{k}}$. □

We can now prove a concrete bound on the winning probability that closely resembles the bound in Section 6.5.

Theorem 72. *For any $k \geq 2$,*

$$\Pr[\textsc{win}] \geq k^{-\frac{1}{k-1}} - \Pr[\textsc{pass}] - O(k/n).$$

Proof. By Lemma 71,

$$\begin{aligned}
\Pr[\textsc{win}] + \Pr[\textsc{pass}] &\geq \frac{k}{n} \sum_{i=r_k+1}^{n} \frac{\tau_k \cdots (\tau_k - k + 1))}{(i-1)\cdots(i-k)} \\
&\geq \frac{k}{n}(\tau_k - k + 1)^k \cdot \sum_{i=\tau_k+1}^{n} \frac{1}{(i-k)^k} \\
&\geq \frac{k}{n}(\tau_k - k + 1)^k \cdot \int_{i=\tau_k+1}^{n+1} \frac{1}{(i-k)^k} \\
&= \frac{k}{n}(\tau_k - k + 1)^k \frac{1}{k-1} \\
&\qquad \left(\frac{1}{(\tau_k - k + 1)^{k-1}} - \frac{1}{(n-k+1)^{k-1}}\right) \\
&= \frac{\tau_k - k + 1}{n} \frac{k}{k-1} \left(1 - \left(\frac{\tau_k - k + 1}{n - k + 1}\right)^{k-1}\right) \\
&\geq \frac{\tau_k - k + 1}{n} \frac{k}{k-1} \left(1 - \left(\frac{\tau_k}{n}\right)^{k-1}\right).
\end{aligned}$$

Now, by definition, we have $\tau_k = n/k^{\frac{1}{k-1}}$. Therefore,

$$\Pr[\textsc{win}] + \Pr[\textsc{pass}] \geq \frac{\tau_k - k + 1}{n} = k^{-\frac{1}{k-1}} - \frac{k+1}{n},$$

which concludes the proof. □

Note that the only way for the algorithm to discard a secretary $\pi(i)$ with $i > \tau_k$ is if $|\mathcal{F}(S_{i+1})| > k$. This event has zero probability in the poset $\mathcal{P}_k$ consisting of k disjoint total orders.

Corollary 73. *Algorithm 6.1 succeeds with probability at least $k^{-\frac{1}{k-1}} - O(k/n)$ on any poset of width at most k. In particular, this holds true for the poset $\mathcal{P}_k$.*

Corollary 73 shows that the algorithm performs well when the poset is "thin". However, the algorithm benefits of "large" posets as well. For example, the algorithm performs well on trees or lattices, or even collections of them.

Say that a tree is m-heavy if the root has at least two children whose subtrees have size m. Let τ be the threshold used by the algorithm, and consider a $\omega(n/\tau)$-heavy tree. Then, we observe that conditioning on the secretary (the root) being after the threshold, the pareto set $\mathcal{F}(S_\tau)$ will be of size at least 2 (with probability approaching 1), and will be so until reaching the secretary. A similar argument applies for a d-dimensional lattice, that is a set of elements $\Lambda_\ell = \{\sum_{i=1}^{d} a_i \mathbf{e}_i \mid a_i \in \{0, 1, \ldots, \ell_i\}\}$ with $\mathcal{P}_{\Lambda_\ell} = \{(\mathbf{v}, \mathbf{v}') \in \Lambda_\ell^2 \mid \exists j\ \mathbf{v}_j > \mathbf{v}'_j \text{ and } \forall i\ \mathbf{v}_i \geq \mathbf{v}'_i\}$, that is the secretary is the origin. We say that Λ_ℓ is m-heavy if $\ell_i \geq m$ for all $1 \leq i \leq d$.

Observation 74. *Algorithm 6.1 with threshold τ succeeds with probability $1 - (\tau/n)^k - o(1)$ on a poset consisting of k disjoint $\omega((n/\tau)\log(1+k))$-heavy trees, as well as on a poset consisting of k disjoint $\omega((n/\tau)\log(1+kd))$-heavy d-dimensional lattices.*

6.7 Conclusions

We have presented an algorithm for the poset secretary that achieves success probability of $k^{-1/(k-1)}(1 - o_k(1))$ for a poset with k maximal elements. We also showed that no algorithm can obtain success probability better than $k^{-1/(k-1)}$. Closing the gap between the two bounds is an open problem and we indeed believe that the proposed algorithm achieves success probability $k^{-1/(k-1)}(1 - o_n(1))$ — which would substantially close the gap.

A drawback of our algorithm is that it takes in input the number k of maximal elements in the poset. It is a very interesting question whether similar bounds can be achieved with an algorithm that is completely oblivious of the poset. A possible algorithm (for which an analysis is missing) skips the first τ_1 candidates and modifies the check at line 5 of Algorithm 6.1 to $a \in \mathcal{F}_{\mathcal{P}}(S) \wedge |\mathcal{F}_{\mathcal{P}}(S)| \leq m(i)$, where $m(i) = \arg\max_{k \geq 1}\{\tau_k \leq i\}$ is the index of the largest threshold before i.

Chapter 6, in full, is a reprint of the paper “Hiring a secretary from a poset” co-authored with Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii in the proceedings of the 12th ACM Conference on Electronic Commerce, EC 2011 [KLVV11]. The dissertation author was the primary investigator and author of this paper.

Bibliography

[ABC$^+$09] 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 *FOCS*, pages 293–302, 2009.

[ADK$^+$08] Elliot Anshelevich, Anirban Dasgupta, Jon M. Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM J. Comput.*, 38(4):1602–1623, 2008.

[ADTW03] Elliot Anshelevich, Anirban Dasgupta, Éva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *STOC*, pages 511–520, 2003.

[AEED$^+$06] Susanne Albers, Stefan Eilts, Eyal Even-Dar, Yishay Mansour, and Liam Roditty. On nash equilibria for a network creation game. In *SODA*, pages 89–98, 2006.

[AF99] David J. Aldous and James A. Fill. *Reversible Markov Chains and Random Walks on Graphs.* 1999. Book in preparation.

[AKK$^+$07] Esteban Arcaute, Adam Kirsch, Ravi Kumar, David Liben-Nowell, and Sergei Vassilvitskii. On threshold behavior in query incentive networks. In *ACM Conference on Electronic Commerce*, pages 66–74, 2007.

[AKL$^+$79] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, László Lovász, and Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *FOCS*, pages 218–223, 1979.

[AMW01] Miklós Ajtai, Nimrod Megiddo, and Orli Waarts. Improved algorithms and analysis for secretary problems and generalizations. *SIAM J. Discrete Math.*, 14(1):1–27, 2001.

[AN04] Krishna B. Athreya and Peter Ney. *Branching processes.* Springer-Verlag, New York, 2004.

[AOY11] Daron Acemoglu, Asuman E. Ozdaglar, and Mehmet Ercan Yildiz. Diffusion of innovations in social networks. In *IEEE Conference on Decision and Control*, pages 2329–2334, 2011.

[APP$^+$12] Omid Amini, David Peleg, Stéphane Pérennes, Ignasi Sau, and Saket Saurabh. On the approximability of some degree-constrained subgraph problems. *Discrete Applied Mathematics*, 160(12):1661–1679, 2012.

[AW04] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics (2nd edition)*. John Wiley Interscience, March 2004.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[BBM09] Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. Improved equilibria via public service advertising. In *SODA*, pages 728–737, 2009.

[BBM10] Maria-Florina Balcan, Avrim Blum, and Yishay Mansour. Circumventing the price of anarchy: Leading dynamics to good behavior. In *ICS*, pages 200–213, 2010.

[BC91] Dimitri P. Bertsekas and David A. Castañon. Parallel synchronous and asynchronous implementations of the auction algorithm. *Parallel Computing*, 17(6-7):707–732, 1991.

[BCC$^+$10] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $o(n^{1/4})$ approximation for densest k-subgraph. In *STOC*, pages 201–210, 2010.

[BDG$^+$09] Moshe Babaioff, Michael Dinitz, Anupam Gupta, Nicole Immorlica, and Kunal Talwar. Secretary problems: weights and discounts. In *SODA*, pages 1245–1254, 2009.

[BG00] Venkatesh Bala and Sanjeev Goyal. A noncooperative model of network formation. *Econometrica*, 68(5):1181–1230, September 2000.

[BIK07] Moshe Babaioff, Nicole Immorlica, and Robert Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA*, pages 434–443, 2007.

[BIKK08] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Online auctions and generalized secretary problems. *SIGecom Exchanges*, 7(2), 2008.

[BIKK12] Christina Brandt, Nicole Immorlica, Gautam Kamath, and Robert Kleinberg. An analysis of one-dimensional schelling segregation. In *STOC*, pages 789–804, 2012.

[BJS10] Niv Buchbinder, Kamal Jain, and Mohit Singh. Secretary problems via linear programming. In *IPCO*, pages 163–176, 2010.

[BK11] Michael Brautbar and Michael Kearns. A clustering coefficient network formation game. In *SAGT*, pages 224–235, 2011.

[Blu03] Lawrence E. Blume. How noise matters. *Games and Economic Behavior*, 44(2):251–271, 2003.

[Bov06] Anton Bovier. *Statistical mechanics of disordered systems: a mathematical perspective*, volume 18. Cambridge University Press, 2006.

[BRST01] Béla Bollobás, Oliver Riordan, Joel Spencer, and Gábor Tusanády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18:279–290, 2001.

[Bru00] F. Thomas Bruss. Sum the odds to one and stop. *Annals of Probability*, 28:1384–1391, 2000.

[CCVV12] Manuel Cebrián, Lorenzo Coviello, Andrea Vattani, and Panagiotis Voulgaris. Finding red balloons with split contracts: robustness to individuals' selfishness. In *STOC*, pages 775–788, 2012.

[CE78] Karen S. Cook and Richard M. Emerson. Power, Equity and Commitment in Exchange Networks. *American Sociological Review*, 43(5):721–739, 1978.

[CEG83] Karen S. Cook, Richard M. Emerson, and Mary R. Gillmore. The Distribution of Power in Exchange Networks: Theory and Experimental Results. *The American Journal of Sociology*, 89(2):275–305, 1983.

[CF05] Colin Cooper and Alan M. Frieze. The cover time of random regular graphs. *SIAM J. Discrete Math.*, 18(4):728–740, 2005.

[CFM$^+$12] Lorenzo Coviello, Massimo Franceschetti, Mathew D. McCubbins, Ramamohan Paturi, and Andrea Vattani. Human matching behavior in social networks: An algorithmic perspective. *PLoS ONE*, 7(8):e41900, 08 2012.

[CFR09] Colin Cooper, Alan M. Frieze, and Tomasz Radzik. Multiple random walks in random regular graphs. *SIAM J. Discrete Math.*, 23(4):1738–1761, 2009.

[CGJ08] Kamalika Chaudhuri, Fan Chung Graham, and Mohammad Shoaib Jamall. A network coloring game. In *WINE*, pages 522–530, 2008.

[CJKT10] Tanmoy Chakraborty, J. Stephen Judd, Michael Kearns, and Jinsong Tan. A behavioral study of bargaining in social networks. In *ACM Conference on Electronic Commerce*, pages 243–252, 2010.

[CLP09] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. In *ICALP (2)*, pages 375–386, 2009.

[Coo04] Matthew Cook. Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40, 2004.

[CP05] Jacomo Corbo and David C. Parkes. The price of selfish behavior in bilateral network formation. In *PODC*, pages 99–107, 2005.

[DFF12] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. Asynchronous rumor spreading in preferential attachment graphs. In *SWAT*, pages 307–315, 2012.

[DM07] John R. Douceur and Thomas Moscibroda. Lottery trees: motivational deployment of networked systems. In *SIGCOMM*, pages 121–132, 2007.

[DMW03] Peter S. Dodds, Roby Muhamad, and Duncan J. Watts. An experimental study of search in global social networks. *Science*, 301(5634):827, 2003.

[DN09] Devansh Dikshit and Yadati Narahari. Truthful and quality conscious query incentive networks. In *WINE*, pages 386–397, 2009.

[DR89] Pradeep Dubey and Jonathan D. Rogawski. *Inefficiency of Nash Equilibria.* Cowles foundation discussion paper. Yale University, Cowles Foundation for Research in Economics, 1989.

[DT90] John Dombrink and William N. Thompson. *The Last Resort: Success And Failure In Campaigns For Casinos.* Nevada Studies in History and Political Science. University of Nevada Press, 1990.

[Dyn63] Eugene B. Dynkin. The optimum choice of the instant for stopping a Markov process. *Sov. Math. Dokl.*, 4:627–629, 1963.

[EKTZ11] Yuval Emek, Ron Karidi, Moshe Tennenholtz, and Aviv Zohar. Mechanisms for multi-level marketing. In *ACM Conference on Electronic Commerce*, pages 209–218, 2011.

[Eme62] Richard M. Emerson. Power-Dependence Relations. *American Sociological Review*, 27(1):31–41, 1962.

[EMPW11] Daniel P. Enemark, Mathew D. McCubbins, Ramamohan Paturi, and Nicholas Weller. Does more connectivity help groups to solve social problems. In *ACM Conference on Electronic Commerce*, pages 21–26, 2011.

[ESW12] Yuval Emek, Jasmin Smula, and Roger Wattenhofer. Stone Age Distributed Computing. In *TIK Report Number 344, Zuerich*, November 2012.

[Fer89] Thomas Ferguson. Who solved the secretary problem. *Statistical Science*, 4:282–296, 1989.

[FG09] Pierre Fraigniaud and George Giakkoupis. The effect of power-law degrees on the navigability of small worlds. In *PODC*, pages 240–249, 2009.

[FG10] Pierre Fraigniaud and George Giakkoupis. On the searchability of small-world networks with arbitrary underlying structure. In *STOC*, pages 389–398, 2010.

[FLM$^+$03] Alex Fabrikant, Ankur Luthra, Elitza N. Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *PODC*, pages 347–351, 2003.

[FPK01] Uriel Feige, David Peleg, and Guy Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[FW10] Ragnar Freij and Johan Wästlund. Partially ordered secretaries. *Electronic Communication in Probability*, 15:504–507, 2010.

[Gia11] George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *STACS*, pages 57–68, 2011.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[GK12] Sanjeev Goyal and Michael Kearns. Competitive contagion in networks. In *STOC*, pages 759–774, 2012.

[GKMN08] Nicholas Georgiou, Malgorzata Kuchta, Michal Morayne, and Jaroslaw Niemiec. On a universal best choice algorithm for partially ordered sets. *Random Struct. Algorithms*, 32(3):263–273, 2008.

[GL09] Doron Goldstein and Michael Langberg. The dense k subgraph problem. *CoRR*, abs/0912.5327, 2009.

[Gne92] A. V. Gnedin. Multicriteria extensions of the best choice problem: Sequential selection without linear order. *Contemp. Math*, 125:153–172, 1992.

[GRST10] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *Proc. 6th WINE*, pages 246–257, 2010.

[GS62] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[GS11] George Giakkoupis and Nicolas Schabanel. Optimal path search in small worlds: dimension matters. In *STOC*, pages 393–402, 2011.

[Han10] Eric Hand. Citizen science: People power. *Nature*, 466(7307):685, 2010.

[HK71] J.E. Hopcroft and R.M. Karp. A $n^{5/2}$ algorithm for maximum matchings in bipartite. In *Switching and Automata Theory, 1971., 12th Annual Symposium on*, pages 122–125. IEEE, 1971.

[HK10] J.W. Hatfield and S.D. Kominers. Matching in networks with bilateral contracts. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 119–120. ACM, 2010.

[HKP04] M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. In *Proc. 5th EC*, pages 71–80, 2004.

[HMS08] Jason D. Hartline, Vahab S. Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *WWW*, pages 189–198, 2008.

[II86] A. Israeli and A. Itai. A fast and simple randomized parallel algorithm for maximal matching. *Information Processing Letters*, 22(2):77–80, 1986.

[IJ90] Amos Israeli and Marc Jalfon. Token management schemes and random walks yield self-stabilizing mutual exclusion. In *Proceedings of the ninth annual ACM symposium on Principles of distributed computing*, PODC '90, pages 119–131, New York, NY, USA, 1990. ACM.

[IMPV10] Amos Israeli, Mathew D. McCubbins, Ramamohan Paturi, and Andrea Vattani. Low memory distributed protocols for 2-coloring. In *SSS*, pages 303–318, 2010.

[Jac05] M. Jackson. *Group Formation in Economics: Networks, Clubs and Coalitions*. G. Demange and M. Wooders, 2005.

[JK08] J. Stephen Judd and Michael Kearns. Behavioral experiments in networked trade. In *ACM Conference on Electronic Commerce*, pages 150–159, 2008.

[JKV10] S. Judd, M. Kearns, and Y. Vorobeychik. Behavioral dynamics and influence in networked coloring and consensus. *Proceedings of the National Academy of Sciences*, 107(34):14978–14982, 2010.

[JKV11] S. Judd, M. Kearns, and Y. Vorobeychik. Behavioral conflict and fairness in social networks. *Internet and Network Economics*, pages 242–253, 2011.

[JY07] Matthew O. Jackson and Leeat Yariv. Diffusion of behavior and equilibrium properties in network games. *American Economic Review*, 97(2):92–98, 2007.

[Kan10] Yashodhan Kanoria. An fptas for bargaining networks with unequal bargaining powers. In *WINE*, pages 282–293, 2010.

[KBB$^+$11a] Y. Kanoria, M. Bayati, C. Borgs, J. Chayes, and A. Montanari. Fast convergence of natural bargaining dynamics in exchange networks. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1518–1537, 2011.

[KBB$^+$11b] Yashodhan Kanoria, Mohsen Bayati, Christian Borgs, Jennifer T. Chayes, and Andrea Montanari. Fast convergence of natural bargaining dynamics in exchange networks. In *SODA*, pages 1518–1537, 2011.

[Kho01] Subhash Khot. Improved inaproximability results for maxclique, chromatic number and approximate graph coloring. In *FOCS*, pages 600–609, 2001.

[KJTW09] Michael Kearns, Stephen Judd, Jinsong Tan, and Jennifer Wortman. Behavioral experiments on biased voting in networks. *Proceedings of the National Academy of Sciences*, January 2009.

[KJV12] M. Kearns, S. Judd, and Y. Vorobeychik. Behavioral experiments on a network formation game. In *Proceedings of the ACM Conference of Electronic Commerce*, 2012.

[KKT03] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[Kle00a] Jon M. Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[Kle00b] Jon M. Kleinberg. The small-world phenomenon: an algorithm perspective. In *STOC*, pages 163–170, 2000.

[Kle01] Jon M. Kleinberg. Small-world phenomena and the dynamics of information. In *NIPS*, pages 431–438, 2001.

[Kle05] R. Kleinberg. A multiple-choice secretary problem with applications to online auctions. In *Proc. 16th SODA*, pages 630–631, 2005.

[KLM02] G. Kubicki, J. Lehel, and M. Morayne. A ratio inequality for binary trees and the best secretary. *Combinatorics, Probability, and Computing*, 11:146–161, 2002.

[KLVV11] Ravi Kumar, Silvio Lattanzi, Sergei Vassilvitskii, and Andrea Vattani. Hiring a secretary from a poset. In *ACM Conference on Electronic Commerce*, pages 39–48, 2011.

[KN10] Nagaraj Kota and Y. Narahari. Threshold behavior of incentives in social networks. In *ACM international conference on Information and knowledge management*, pages 1461–1464, 2010.

[Koz10] Jakub Kozik. Dynamic threshold strategy for universal best choice problem. In *Proc. 21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms*, pages 439–452, 2010.

[KP99] Elias Koutsoupias and Christos H. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, 1999.

[KP09] Elias Koutsoupias and Christos H. Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009.

[KR05] Jon M. Kleinberg and Prabhakar Raghavan. Query incentive networks. In *FOCS*, pages 132–141, 2005.

[KSM06] Michael Kearns, Siddharth Suri, and Nick Montfort. A behavioral study of the coloring problem on human subject networks. *Science*, 313:2006, 2006.

[KSS97] Henry Kautz, Bart Selman, and Mehul Shah. Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40:63–65, March 1997.

[KT79] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, pages 263–291, 1979.

[KT08a] Jon M. Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *STOC*, pages 295–304, 2008.

[KT08b] Jon M. Kleinberg and Éva Tardos. Balanced outcomes in social exchange networks. In *STOC*, pages 295–304, 2008.

[Kuh55] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[KUW86] R.M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in random nc. *Combinatorica*, 6(1):35–48, 1986.

[Lig85] T.M. Liggett. *Interacting particle systems*, volume 276. Springer Verlag, 1985.

[LL96] Bibb Latané and Todd L'Herrou. Spatial clustering in the conformity game: Dynamic social impact in electronic groups. *Journal of Personality and Social Psychology*, 70:1218–1230, 1996.

[LNNK$^+$05] David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11623–11628, August 2005.

[LPSP08] Z. Lotker, B. Patt-Shamir, and S. Pettie. Improved distributed approximate matching. In *Proceedings of the ACM symposium on Parallelism in algorithms and architectures*, pages 129–136, 2008.

[LRSH11] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020, 2011.

[MHB10] M. Zadimoghaddam M. H. Bateni, M. T. Hajiaghayi. Submodular secretary problem and extensions. In *Proc. 6th WINE*, pages 39–52, 2010.

[Mil63] Stanley Milgram. Behavioral study of obedience. *Journal of Abnormal and Social Psychology*, 67(4):371–378, 1963.

[Mil67] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[Mil74] Stanley Milgram. *Obedience to authority: an experimental view.* Harper & Row, 1974.

[Mil77] Stanley Milgram. *The Individual in a Social World: Essays and Experiments.* Addison Wesley Publishing Company, 1977.

[Mor98] Michal Morayne. Partial-order analogue of the secretary problem; the binary tree case. *Discrete Math*, 184:165–181, 1998.

[MPW09] Mathew D. McCubbins, Ramamohan Paturi, and Nicholas Weller. Connected Coordination: Network Structure and Group Coordination. *American Politics Research*, 37:899–920, 2009.

[MS09] A. Montanari and A. Saberi. Convergence to equilibrium in local interaction games. In *IEEE Symposium on Foundations of Computer Science*, pages 303–312. IEEE, 2009.

[MS10] Elchanan Mossel and Grant Schoenebeck. Reaching consensus on social networks. In *ICS*, pages 214–229, 2010.

[MS12] Jason R. Marden and Jeff S. Shamma. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Games and Economic Behavior*, 75(2):788–808, 2012.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[MVV11] Petros Mol, Andrea Vattani, and Panagiotis Voulgaris. The effects of diversity in aggregation games. In *ICS*, pages 509–521, 2011.

[Pap01] Christos H. Papadimitriou. Algorithms, games, and the internet. In *STOC*, pages 749–753, 2001.

[PPR$^+$11] Galen Pickard, Wei Pan, Iyad Rahwan, Manuel Cebrián, Riley Crane, Anmol Madan, and Alex Pentland. Time critical social mobilization. *Science*, 334(6055):509–512, 2011.

[PR00] David Peleg and Vitaly Rubinovich. A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction. *SIAM J. Comput.*, 30(5):1427–1442, 2000.

[Pra64] J.W. Pratt. Risk aversion in the small and in the large. *Econometrica: Journal of the Econometric Society*, pages 122–136, 1964.

[Pre99] John Preater. The best-choice problem for partially ordered objects. *Operation Research Letter*, 25:187–190, 1999.

[RS92] A.E. Roth and M.A.O. Sotomayor. *Two-sided matching: A study in game-theoretic modeling and analysis*, volume 18. Cambridge University Press, 1992.

[San06] Nicola Santoro. *Design and Analysis of Distributed Algorithms*. Wiley Series on Parallel And Distributed Computing. Wiley, 2006.

[SB76] H.A. Simon and C.I. Barnard. *Administrative behavior*, volume 3. Cambridge University Press, 1976.

[Sch71] T. C. Schelling. Dynamic models of segregation. *Journal of Mathematical Sociology*, 1(2):143–186, 1971.

[Sch78] Thomas C. Schelling. *Micromotives and Macrobehavior.* W. W. Norton & Company, October 1978.

[Sch09] Sebastian Schnettler. A structured overview of 50 years of small-world research. *Social Networks*, 31(3):165–178, 2009.

[SH88] Galen H. Sasaki and Bruce Hajek. The time complexity of maximum matching by simulated annealing. *J. ACM*, 35(2):387–403, 1988.

[Sin11] Yaron Singer. How to win friends and influence people, truthfully: Influence maximization mechanisms for social networks. In *WSDM*, 2011.

[Sot11] J. A. Soto. Matroid secretary problem in the random assignment model. In *Proc. 22nd SODA*, pages 1275–1284, 2011.

[Sur04] James Surowiecki. *The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations.* Doubleday Books, 2004.

[SW09] Yogeshwer Sharma and David P. Williamson. Stackelberg thresholds in network routing games or the value of altruism. *Games and Economic Behavior*, 67(1):174–190, 2009.

[SW11] S. Suri and D.J. Watts. Cooperation and contagion in web-based, networked public goods experiments. *PLoS One*, 6(3):e16836, 2011.

[TCG$^+$11] John C. Tang, Manuel Cebrián, Nicklaus A. Giacobe, Hyun-Woo Kim, Taemie Kim, and Douglas ”Beaker” Wickert. Reflecting on the darpa red balloon challenge. *Commun. ACM*, 54(4):78–85, 2011.

[TM69] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[Val84] Leslie G. Valiant. A theory of the learnable. In *STOC*, pages 436–445, 1984.

[Val00] Leslie G. Valiant. *Circuits of the Mind.* Oxford University Press, USA, 2000.

[Vet02] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *FOCS*, pages 416–, 2002.

[VNB66] J. Von Neumann and A.W. Burks. Theory of self-reproducing automata. 1966.

[WDN02] Duncan J. Watts, Peter S. Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296(5571):1302, 2002.

[Wil99] D. Willer. *Network Exchange Theory.* Praeger, 1999.

[WS98] D.J. Watts and S.H. Strogatz. Collective dynamics of small-worldnetworks. *Nature*, 393(6684):440–442, 1998.

[WSW12] J. Wang, S. Suri, and D.J. Watts. Cooperation and assortativity with endogenous partner selection. In *Proceedings of the ACM Conference of Electronic Commerce*, 2012.

[YS03] Bin Yu and Munindar P. Singh. Searching social networks. In *AAMAS*, pages 65–72, 2003.

[ZA04] Zhang, Jun and Van Alstyne, Marshall. SWIM: fostering social network based information search. In *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems*, volume 2, page 1568, 2004.

[Zim08] Philip Zimbardo. *The Lucifer Effect: Understanding How Good People Turn Evil.* Random House Publishing Group, 2008.

[ZSP08] Michael M. Zavlanos, Leonid Spesivtsev, and George J. Pappas. A distributed auction algorithm for the assignment problem. In *Conference on Decision and Control*, pages 1212–1217, 2008.