ABSTRACT

Title of dissertation:	COMPUTATIONAL ANALYSIS OF INTELLIGENT AGENTS: SOCIAL AND STRATEGIC SETTINGS
	Anshul Sawant, Doctor of Philosophy, 2016
Dissertation co-directed by:	Prof. V.S. Subrahmanian, Department of Computer Science Prof. Mohammad Taghi. Hajiaghayi, Department of Computer Science

The central motif of this work is prediction and optimization in presence of multiple interacting intelligent agents. We use the phrase 'intelligent agents' to imply in some sense, a 'bounded rationality', the exact meaning of which varies depending on the setting. Our agents may not be 'rational' in the classical game theoretic sense, in that they don't always optimize a global objective. Rather, they rely on heuristics, as is natural for human agents or even software agents operating in the real-world. Within this broad framework we study the problem of influence maximization in social networks where behavior of agents is myopic, but complication stems from the structure of interaction networks. In this setting, we generalize two well-known models and give new algorithms and hardness results for our models. Then we move on to models where the agents reason strategically but are faced with considerable uncertainty. For such games, we give a new solution concept and analyze a real-world game using out techniques. Finally, the richest model we consider is that of Network Cournot Competition which deals with strategic resource allocation in hypergraphs, where agents reason strategically and their interaction is specified indirectly via player's utility functions. For this model, we give the first equilibrium computability results. In all of the above problems, we assume that payoffs for the agents are known. However, for real-world games, getting the payoffs can be quite challenging. To this end, we also study the inverse problem of inferring payoffs, given game history. We propose and evaluate a data analytic framework and we show that it is fast and performant.

COMPUTATIONAL ANALYSIS OF INTELLIGENT AGENTS: SOCIAL AND STRATEGIC SETTINGS

by

Anshul Sawant

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Dr. V. S. Subrahmanian, Co-chair/Co-advisor
Dr. MohammadTaghi Hajiaghayi, Co-chair/Co-advisor
Dr. Sarit Kraus, Dept. of Computer Science, Bar-Ilan University
Dr. Dana Nau
Dr. Subramanian Raghavan, Robert H. Smith School of Business and Institute for Systems Research

Dedication

Dedicated to my mother, Dr. Satya Sawant.

Acknowledgments

To start with, I would like to thank my advisors, Prof. V.S. Subrahmanian and Prof. MohammadTaghi Hajiaghayi. Both have provided excellent guidance to me over the years. They have provided me with invaluable direction in my efforts and have been a constant source of ideas and inspiration. I would also like to thank Prof. Naveen Garg without whose guidance and inspiration, I would never have started this Ph.D. program.

Over the course of my Ph.D. I have had the opportunity to collaborate with several wonderful people. I am thankful for contributions of (in no particular order) Hamid Mahini, David Malec, Sarit Kraus, Chanhyun Kang, Melika Abolhassani, John P. Dickerson, Eric Demaine and S. Raghavan. I would like to thank Sarit Kraus, S.Raghavan and Dana Nau for serving on my dissertation committee.

I am very thankful to Jennifer Story and Fatima Bangura. They made this such a smooth sailing. Never once though the course of my Ph.D did I have to think about bureaucratic hassles.

Of course, none of this would have been any fun without friends. I would specifically like to thank Amit, Bhaskar, Kartik, Manish, Ramakrishna (guys at 5002), Meethu, Sudha (girls at 117), Ioana, Chanhyun, Eunhui, Noseong, Srijan, Koel and Kanthi (I am sure I missed a few). In addition, I had a great time with my mates at Terrapin Cricket Club, who are currently the 4^{th} ranked college cricket team in the U.S., go terps!

Finally and most importantly, I would like to thank my family. My mother, Satya, has been a bedrock of support throughout my life. She is the strongest person I know. Neeti, my wife for inspiring me to undertake this herculean task. My daughter, Avni, for being such a delight. My sister and my brother-in-law, Vidush and Vikas, for support and advice. I would also like to thank my extended family, especially, Sheela and Umed Bhaker, and Saroj and Sumer Savant. Besides being a constant source of love and support, they helped me through one of the toughest patches in my life and I would have been lost without them.

Contents

Chapter 1

Introduction

The goal of this work is the analysis and prediction of behavior of multiple interacting intelligent agents in social and strategic settings. We use the phrase 'intelligent agents' to imply in some sense, a 'bounded rationality', the exact meaning of which varies somewhat depending on the setting under consideration. However, we note that our agents may not be 'rational' in the classical game theoretic sense, in that they don't always optimize a global objective. Rather, they rely on heuristics, as is natural for human agents or even software agents operating in the real-world. We propose and analyze theoretical models for behavior of such agents in social and strategic settings. In addition, we develop data analytic frameworks for analysis of real-world strategic games.

Consider the case of a product manufacturer trying to promote a product by leveraging word-of-mouth publicity. It is reasonable to assume that perception of the product for a person is influenced by reviews from her peers. Thus, the manufacturer wants to distribute coupons among a community to increase the expected number of users. We note that the optimal solution involves shaping behavior of intelligent agents whose behavior depends not only on the environment, but also on behavior of other agents. This theme of interacting agents is common to a great variety of scenarios. In the problems we study, the models of agents and their interactions vary depending on details of the problem. In the social network setting, the agents are influenced by their peers. In the case of markets, the players influence each other by their influence on prices, whereas in strategic settings, player's actions have direct influence on payoffs of other players. Thus, we study the problem of algorithm design for interacting intelligent agents.

1.1 Algorithms for Networked Agents

Social networks are the graph of relationships within a group of individuals. They play a very important role in diffusion of influence among the interconnected individuals. In social setting, we study the problem of influence maximization – the problem of ensuring that an idea or product is adopted by maximum number of individuals, given limited resources at disposal of the planner. We study two problems related to influence maximization. They mainly differ in the aspects of diffusion the planner controls. In the first case the planner can only control the order in which the diffusion happens in the network and in the second case planner can only control the incentives for nodes to adopt the product. We study the first and second cases in Chapters 2 and 3 respectively.

In Chapter 2, we study the propagation of influence in a social network with negative feedback. Adoption or rejection of ideas, products, and technologies in a society is often governed by simultaneous propagation of positive and negative influences. Consider a planner trying to introduce an idea in different parts of a society at different times. How should the planner design a schedule considering this fact that positive reaction to the idea in early areas has a positive impact on probability of success in later areas, whereas a flopped reaction has exactly the opposite impact? We generalize a well-known economic model which has been recently used by Chierichetti et al. [28]. In this model the reaction of each area is determined by its initial preference and the reaction of early areas. We generalize previous works by studying the problem when people in different areas have various behaviors.

In Chapter 3, we study the power of fractional allocations of resources to maximize influence in a network. This work extends in a natural way the well-studied model by Kempe et al. [75], where a designer selects a (small) seed set of nodes in a social network to influence directly, this influence cascades when other nodes reach certain thresholds of neighbor influence, and the goal is to maximize the final number of influenced nodes. Despite extensive study from both practical and theoretical viewpoints, this model limits the designer to a binary choice for each node, with no way to apply intermediate levels of influence. This model captures some settings precisely, e.g. exposure to an idea or pathogen, but it fails to capture very relevant concerns in others, for example, a manufacturer promoting a new product by distributing five “20% off” coupons instead of giving away one free product.

From the relatively simple models of influence maximization, we move on to a much richer model – that of strategic decision making in hypergraphs. In this setting, the agents have a cost associated with resource allocation along hyperedges. In return, the agents derive a reward from the hyperedges. The interaction among the agents is implicitly specified by these cost and reward functions. In Chapter 4, We study such a model, which can be viewed as a generalization of the well-known Cournot model from economics. We give one of the first positive results on computability of equilibrium in this setting. Efficient computability is an important prerequisite for solution concepts [102]. Therefore, it is an important first step in analysis of such models. We give a convex formulation for the problem. This opens the possibility that that various dynamics, e.g. best-response, may converge to equilibrium. This is a direction of further research. We also contribute the first combinatorial algorithm for computing an equilibrium for the classical Cournot Oligopoly.

1.2 Real-world Strategic Games

We now move on to conceptually simple but computationally challenging strategic setting – multiplayer simultaneous games and its application to a real-world strategic game where interaction between players is explicitly specified by payoff matrices for the players. Game theory has long been used to study the strategic interactions of rational players in such games. However, computational frameworks that use real-world data to make meaningful inferences and predictions about these agents are a relatively new development in this field. In this area, we make important contributions. In Chapter 5, we develop a framework to analyze equilibria of games with multiple possible payoff matrices with no priors on these matrices. The use of game theory to model conflict has been studied by several researchers, spearheaded by Schelling. Most of these efforts assume a single payoff matrix that captures players' utilities under different assumptions about what the players will do. Our experience in counter-terrorism applications is that experts disagree on these payoffs. In order to effectively enumerate large numbers of equilibria with payoffs provided by multiple experts, we propose a novel combination of vector payoffs and well-supported ϵ-approximate equilibria. We develop bounds related to computation of these equilibria for some special cases, and give a quasi-polynomial time approximation scheme (QPTAS) for the general case when the number of players is small (which is true in many real-world applications). Leveraging this QPTAS, we give efficient algorithms to find such equilibria and experimental results showing they work well on simulated data. We then present a real-world application in which there are five parties including four governmental entities and the terrorist group Lashkar-e-Taiba[1]. The goal was to understand whether there were any pure (or mixed) equilibria in which the group's terrorist acts could be significantly reduced.

[1] Lashkar-e-Taiba, translated variously from Urdu into "Army of the Pure" or "Army of the Pious", is a prominent south Asian terrorist organization responsible for attacks in India, Kashmir, Pakistan, and Afghanistan, including the three days of attacks in 2008 in Mumbai, India, that resulted in the deaths of 166 innocent people [122, 124].

1.3 Payoff Inference

Though, most game theory assumes that payoff functions are provided as input, getting payoff matrices in strategic games (e.g. corporate negotiations, counter-terrorism operations) has proven difficult. In Chapter 6, We develop a data analytic framework for learning payoff functions of players from a given game history. The payoffs can then be analyzed in manner similar to that in Chapter 5. Ng and Russell pioneered inverse reinforcement learning (IRL) that studies the problem of learning payoffs in a non-adversarial setting. Recently, Waugh showed how a maximum entropy-based approach learns a probability distribution function (pdf) over joint actions for each player under the assumption that players are playing a correlated equilibrium. Unlike past work, we study the case where histories of past actions are sparse and where players are not fully rational. We set up a feasible system of inequalities on payoffs using regret minimization and best response dynamics. Then we evaluate three solution selection methods on this system of inequalities: (i) approximately compute a centroid of the resulting polytope (ii) soft constraints approach where we penalize violation of constraints in the objective and (iii) an SVM based approach. To test the effectiveness of our approach, we run experiments on synthetic and real-world data. We show that our methods have both good accuracy and runtimes.

Chapter 2

Scheduling a Cascade

2.1 Introduction

People's opinions are usually formed by their friends' opinions. Whenever a new concept is introduced into a society, the high correlation between people's reactions initiates an influence propagation. Under this propagation, the problem of promoting a product or an opinion depends on the problem of directing the flow of influences. As a result, a planner can develop a new idea by controlling the flow of influences in a desired way. Although there have been many attempts to understand the behavior of influence propagation in a social network, the topic is still controversial due to lack of reliable information and complex behavior of this phenomenon.

For example, one compelling approach is "seeding" which was introduced by the seminal work of Kempe, Kleinberg, and Trados [75] and is well-studied in the literature [75, 76, 77, 95]. The idea is to influence a group of people in the initial investment period and spread the desired opinion in the ultimate exploitation phase. Another approach is to use time-varying and customer-specific prices to propagate the product

[0]This is a joint work with Mohammad T. Hajiaghayi and Hamid Mahini. A version of this work appeared in *Symposium on Algorithmic Game Theory '13* [61].

(see e.g., [3, 4, 62]). All of these papers investigate the influence propagation problem when only positive influences spread into the network. However, in many real world applications people are affected by both positive and negative influences, e.g., when both consenting and dissenting opinions broadcast simultaneously.

We generalize a well-known economic model introduced by Arthur [9]. This model has been recently used by Chierichetti, Kleinberg, and Panconesi [28].

As Jon Kleinberg motivated the problem[1], assume an organization is going to develop a new idea in a society where the people in the society are grouped into n different areas. Each area consists of people living near each other with almost the same preferences.

The planner schedules to introduce a new idea in different areas at different times. Each area may accept or reject the original idea. Since areas are varied and effects of early decisions boost during the diffusion, a schedule-based strategy affects the spread of influences.

This framework closely matches to various applications from economics to social science to public health where the original idea could be a new product, a new technology, or a new belief.

Consider the spread of two opposing influences simultaneously. Both positive and adverse reactions to a single idea originate different flows of influences simultaneously. In this model, each area has an *initial preference* of $\mathcal{Y}$ or $\mathcal{N}$. The initial preference of $\mathcal{Y}$ ($\mathcal{N}$) means the area will accept (decline) the original idea when there are no network externalities. Let c_i be a non-negative number indicating how reaction of people in area i depends on the others'. We call c_i the *threshold* of area i. Assume the planner introduced the idea in area i at time s. Let $m_{\mathcal{Y}}$ and $m_{\mathcal{N}}$ be the number of areas which accept or reject the idea before time s. If $|m_{\mathcal{Y}} - m_{\mathcal{N}}| \geq c_i$ the people in area i decide based on the majority of previous adopters. It means they adopt the idea if $m_{\mathcal{Y}} - m_{\mathcal{N}} \geq c_i$ and drop it if $m_{\mathcal{N}} - m_{\mathcal{Y}} \geq c_i$. Otherwise, if $|m_{\mathcal{Y}} - m_{\mathcal{N}}| < c_i$ the

[1] via personal communication.

people in area i accept or reject the idea if the initial preference of area i is $\mathcal{Y}$ or $\mathcal{N}$ respectively. The planner does not know exact initial preferences and has only prior knowledge about them. Formally speaking, for area i the planner knows the initial preference of area i will be $\mathcal{Y}$ with probability p_i and will be $\mathcal{N}$ with probability $1-p_i$. We call p_i the *initial acceptance probability* of area i.

We consider the problem when the planner classifies different areas into various types. The classification is based on the planner's knowledge about the reaction of people living in each area. Hence, the classification is based on different features, e.g., preferences, beliefs, education, and age such that people in areas with the same type react almost the same to the new idea. It means all areas of the same type have the same threshold c_i and the same initial acceptance probability p_i. It is worth mentioning previous works only consider the problem when all areas have the same type, i.e., all p_i's and c_i's are the same [9, 28].

The planner wants to manage the flow of influences, and her *spreading strategy* is a permutation π over different areas. Her goal is to find a spreading strategy π which maximizes the expected number of adopters.

We consider both *adaptive* and *non-adaptive* spreading strategies in this chapter. In the adaptive spreading strategy, the planner can see results of earlier areas for further decisions. On the other hand, in the non-adaptive spreading strategy the planner decides about the permutation in advance.

We show the effect of a spreading strategy on the number of adopters with an example in Section 2.1.1.

2.1.1 Examples

Example 1. *Consider a society with* 3 *areas and* 3 *types. The planner prior is as follows. Initial acceptance probabilities of areas* 1, 2, *and* 3 *are* 0.2, 0.5, *and* 0.8 *respectively. Thresholds of areas* 1, 2, *and* 3 *are* 1, 2, *and* 3 *respectively (See Figure*

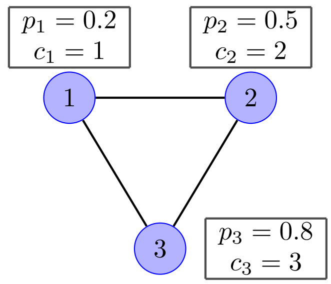


Figure 2.1: A society with 3 areas. The expected number of adopters for spreading strategy $\pi = (1, 2, 3)$ is 1.5. The expected number of adopters for spreading strategy $\pi' = (3, 1, 2)$ is 2.4.

2.1). Consider spreading strategy $\pi = (1, 2, 3)$. People in area 1 accept the idea with probability $p_1 = 0.2$. Threshold of area 2 is 2. It means people in area 2 decide based on initial rule and accept the idea with probability $p_2 = 0.5$. Threshold of area 3 is 3. Thus, people in area 3 decide based on initial rule as well and accept the idea with probability $p_3 = 0.8$. Therefore, the expected number of adopters for spreading strategy π is $p_1 + p_2 + p_3 = 1.5$. In order to see the impact of an optimal spreading strategy consider spreading strategy $\pi' = (3, 1, 2)$. People in area 3 accept the idea with probability $p_3 = 0.8$. Threshold of area 1 is 1. It means the decision of people in area 1 is correlated to the decision of people in area 3. In other word, people in area 1 follow the decision of people in area 3. Thus, there are two possible scenarios. First, both areas 3 and 1 accept the idea. The probability of this scenario is $p_3 = 0.8$. The second scenario is that both areas 3 and 1 reject the idea. The probability of the second scenario is $1 - p_3 = 0.2$. In both scenario the threshold of area 2 is hit. Hence, area 2 will accept the idea with probability $p_3 = 0.8$. Therefore, the expected number of adopters for spreading schedule π' is $3p_3 = 2.4$.

Example 2. *At the first glance, it seems a greedy approach leads us to find the best non-adaptive spreading strategy. The greedy approach is to first schedule a node with the highest probability of adopting. We find a counter-example for this greedy approach with a society with 3 areas.*

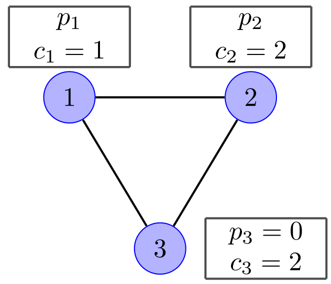


Figure 2.2: A society with 3 areas. The expected number of adopters for spreading strategy $\pi = (1, 2, 3)$ is $p_1 + p_2 + p_1 p_2$. The expected number of adopters for spreading strategy $\pi' = (2, 1, 3)$ is $3p_2$.

Consider a society with 3 areas and 3 types. Area 1 has threshold 1 and areas 2 and 3 have threshold 2. Initial acceptance probabilities are $p_1 > p_2 > p_3 = 0$ (See Figure 2.2). The greedy approach leads us to spreading strategy $\pi = (1, 2, 3)$. Assume the planner uses spreading strategy π. The probability that people in area 1 accept the idea is p_1. The threshold for area 2 is 2. Hence, they decide based on initial rule. It means the probability that people in area 2 accept the idea is p_2. At last, if both area 1 and 2 accept the idea then people in area 3 accept the idea with probability $p_1 p_2$ based on threshold rules . Otherwise, they reject it because $p_3 = 0$, i.e., area 3 has an initial preference of $\mathcal{N}$ for sure. Thus, the expected number of adopter is $p_1 + p_2 + p_1 p_2$. Now, assume the planner uses spreading strategy $\pi' = (2, 1, 3)$. Area 2 accepts the idea with probability p_2. The threshold of area 1 is 1. It means area 1 is a follower of area 2 under spreading strategy π'. Hence, there are two possibilities. Both areas 1 and 2 accept the idea with probability p_2 or both areas 1 and 2 reject the idea with probability $1 - p_2$. In both cases area 3 decides based on the threshold rule. Therefore, there are 3 adopters with probability p_2 or all areas reject the idea with probability $1 - p_2$. Hence, the expected number of adopter is $3p_2$ for spreading strategy π'. One can check spreading strategy π' is better that π for various probabilities p_1 and p_2, e.g., $p_1 = 0.4$ and $p_2 = 0.3$ or $p_1 = 0.8$ and $p_2 = 0.7$.

Example 3. *The result of Theorem 2.1.3 leads us to the following conjecture for the* *partial propagation setting.*

> *"Consider an arbitrary non-adaptive spreading strategy in the partial propagation setting. If all initial acceptance probabilities are greater/less than $\frac{1}{2}$, then adding an edge to the graph helps/hurts promoting the new product.".*

This conjecture has several consequences, e.g., a complete graph is the best graph for spreading a new idea when initial acceptance probabilities are greater than $\frac{1}{2}$. This eventuates directly Theorem 2.1.3. Surprisingly, this conjecture does not hold. We present an example with the same initial acceptance probabilities of less than $\frac{1}{2}$ such that adding a relationship between two areas increases the expected number of adopters.

Consider a society with 4 areas and only one type. Initial acceptance probabilities and thresholds for all areas are p and 1 respectively. Consider spreading strategy $\pi = (1, 2, 3, 4)$ and a society which is represented by graph G (See Figure 2.3). Areas 1, 2, and 3 decide about the idea independently and accept it with probability p. Threshold of area 4 is 1. Hence, people in area 4 accept the idea if there are at least two adopters so far. Therefore, area 4 accept the idea with probability $3p^2(1-p) + p^3$ and the expected number of adopters is $3p + 3p^2(1-p) + p^3$. Assume influences also propagate between area 1 and 2. In this case the society is represented by graph G' (See Figure 2.3). Threshold of area 2 is 1. Hence, area 2 is a follower of area 1 under spreading strategy π. Thus, there are two possibilities when area 2 is scheduled. Both area 1 and 2 accept the idea with probability p or both reject it with probability $1-p$. Area 3 decide independently and accept the idea with probability p. Threshold of area 4 is 1. Thus, area 4 is also a follower of both area 1 and 2. Therefore, the expected number of adopter is $4p$ in this case. One can check $3p+3p^2(1-p)+p^3$ is greater than $4p$ if and only if $0.5 < p < 1$. It means when $p < 0.5$ (resp., $p > 0.5$) the number of adopters increases (resp., decreases) by adding a relation to the society.

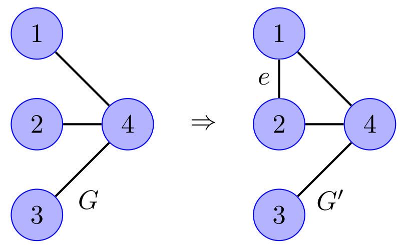


Figure 2.3: This figure represents a partial propagation setting with 4 areas. All Thresholds are equal to 1 and all initial acceptance probabilities are p. The expected number of adopters for spreading strategy $\pi = (1, 2, 3, 4)$ is $3p + 3p^2(1-p) + p^3$ for a society which is represented by graph G. The expected number of adopters for spreading strategy $\pi = (1, 2, 3, 4)$ is $4p$ for a society which is represented by graph G'. Note that $3p^2(1-p) + p^3$ is greater than p if and only if $0.5 < p < 1$

2.1.2 Related Work

We are motivated by a series of well-known studies in economics and politics literature in order to model people's behavior [9, 14, 17, 56]. Arthur first proposed a framework to analyze people's behavior in a scenario with two competing products [9]. In this model people are going to decide about one of two competing products alternatively. He studied the problem when people are affected by all previous customers, and the planner has the same prior knowledge about people's behavior, i.e., people have the same types. He demonstrated that a cascade of influences is formed when products have positive network externalities, and early decisions determine the ultimate outcome of the market. It has been showed the same cascade arises when people look at earlier decisions, not because of network externalities, but because they have limited information themselves or even have bounded rationality to process all available data [14, 17].

Chierichetti, Kleinberg, and Panconesi argued when relations between people form an arbitrary network, the outcome of an influence propagation highly depends on the order in which people make their decisions [28]. In this setting, a potential spreading strategy is an ordering of decision makers. They studied the problem of finding a spreading strategy which maximizes the expected number of adopters when people have the same type, i.e., people have the same threshold c and the same initial acceptance probability p. They proved for any n-node graph there is an adaptive spreading strategy with at least $O(np^c)$ adopters. They also showed for any n-node graph all adaptive spreading strategies result in at least (resp. at most) $\frac{n}{2}$ if initial acceptance probability is less (resp. greater) than $\frac{1}{2}$. They considered the problem on an arbitrary graph when nodes have the same type. While we mainly study the problem on a complete graph when nodes have various types, we improve their result in our setting and show the expected number of adopters for all adaptive spreading strategies is at least (resp. at most) np if initial acceptance probability is $p \geq \frac{1}{2}$ (resp. $p \leq \frac{1}{2}$). We also show the problem of designing the best spreading strategy is hard on an arbitrary graph with several types of customers. We prove it is $\#P$-complete to compute the expected number of adopters for a given spreading strategy.

The problem of designing an appropriate marketing strategy based on network externalities has been studied extensively in the computer science literature. For example, Kempe, Kleinberg, and Trados [75] studied the following question in their seminal work: How can we influence a group of people in an investment phase in order to propagate an idea in the exploitation phase? This question was introduced by Domingos and Richardson [45]. The answer to this question leads to a marketing strategy based on seeding. There are several papers that study the same problem from an algorithmic point of view, e.g., [23, 76, 95]. Hartline, Mirrokni, and Sundararajan [62] also proposed another marketing strategy based on scheduling for selling a product. Their marketing strategy is a permutation π over customers and price p_i for customer i. The seller offers the product with price p_i to customer i at time t where $t = \pi^{-1}(i)$. The goal is to find a marketing strategy which maximizes the profit of the seller. This approach is followed by several works, e.g., [3, 4, 8]. These papers study the behavior of an influence propagation when there is only one flow on influences in the network. In this chapter, we study the problem of designing a spreading strategy when both negative and positive influences propagate simultaneously.

The propagation of competitive influences has been studied in the literature (See [55] and its references). These works studied the influence propagation problem in the presence of competing influences, i.e., when two or more competing firms try to propagate their products at the same time. However we study the problem of influence propagation when there exist both positive and negative reactions to the same idea. There are also studies which consider the influence propagation problem in the presence of positive and negative influences [26, 83]. Che et al. [26] use a variant of the independent cascade model introduced in [75]. They model negative influences by allowing each person to flips her idea with a given probability q. Li et al. [83] model the negative influences by negative edges in the graph. Although they study the same problem, we use different models to capture behavior of people.

2.1.3 Our Results

We analyze an influence propagation phenomenon where two opposing flows of influences propagate through a social network. As a result, a mistake in the selection of early areas may result in propagation of negative influences. Therefore a good understanding of influence propagation dynamics seems necessary to analyze the properties of a spreading strategy. Besides the previous papers which have studied the problem with just one type [9, 28], we consider the scheduling problem with various types. Also, we mainly study the problem in a *full propagation* setting as it matches well to our motivations. In the full propagation setting news and influences propagate between every two areas. One can imagine how internet, media, and electronic devices broadcast news and influences from everywhere to everywhere. In the *partial propagation* setting news and influences do not necessarily propagate between every two areas. In the partial propagation setting the society can be modeled with a graph, where there is an edge from area i to area j if and only if influences propagate from area i to area j.

Our main focus is to analyze the problem when the planner chooses a non-adaptive spreading strategy. Consider an arbitrary non-adaptive spreading strategy when initial preferences of all areas are p. The expected number of adopters is exactly np if all areas decide independently. We demonstrate that in the presence of network influences, the expected number of adopters in greater/less than np if initial acceptance probability p is greater/less than $\frac{1}{2}$. These results have a bold message: **The influence propagation is an amplifier for an appealing idea and an attenuator for an unappealing idea.** Chierichetti, Kleinberg, and Panconesi [28] studied the problem on an arbitrary graph with only one type. They proved the number of adopters is greater/less than $\frac{n}{2}$ if initial acceptance probability p is greater/less than $\frac{1}{2}$. Theorem 2.1.3 improves their result from $\frac{n}{2}$ to np in our setting. Consider an arbitrary non-adaptive spreading strategy π in the full propagation setting. Assume all initial acceptance probabilities are equal to p. If $p \geq \frac{1}{2}$, then the expected number of adopters is at least np. Furthermore, If $p \leq \frac{1}{2}$, then the expected number of adopters is at most np.

Chierichetti, Kleinberg, and Panconesi [28] studied the problem of designing an optimum spreading strategy in the partial propagation setting. They design an approximation algorithm for the problem when the planner has the same prior knowledge about all areas, i.e., all areas have the same type. We study the same problem with more than one type. We first consider the problem in the full propagation setting. One approach is to consider a non-adaptive spreading strategy with a constant number of switches between different types. The planner has the same prior knowledge about areas with the same type. It means areas with the same type are identical for the planner. Thus any spreading strategy can be specified by types of areas rather than areas themselves. Let $\tau(i)$ be the type of area i and $\tau(\pi)$ be the sequence of types for spreading strategy π. For a give spreading strategy π a *switch* is a position k in the sequence such that $\tau(\pi(k)) \neq \tau(\pi(k+1))$. As an example consider a society with 4 areas. Areas 1 and 2 are of type 1. Areas 3 and 4 are of type 2. Then spreading strategy $\pi_1 = (1, 2, 3, 4)$ with

$\tau(\pi_1) = (1, 1, 2, 2)$ has a switch at position 2 and spreading strategy $\pi_2 = (1, 3, 2, 4)$ with $\tau(\pi_2) = (1, 2, 1, 2)$ has switches at positions 1, 2, and 3. A σ-switch spreading strategy is a spreading strategy with at most σ switches. For any constant σ, there exists a society with areas of two types such that no σ-switch spreading strategy is optimal. We construct a society with n areas with $\frac{n}{2}$ areas of type 1 and $\frac{n}{2}$ areas of type 2. We demonstrate an optimal non-adaptive spreading strategy should switch at least $\Omega(n)$ times. It means no switch-based non-adaptive spreading strategy can be optimal. We prove Theorem 2.1.3 formally in Section 2.5.

On the positive side, we analyze the problem when thresholds are drawn independently from an unknown distribution and initial acceptance probabilities are arbitrary numbers. We characterize the optimal non-adaptive spreading strategy in this case. Assume that the planner's prior knowledge about all values of c_i's is the same, i.e., all c_i's are drawn independently from the same but unknown distribution. Let initial acceptance probabilities be arbitrary numbers. Then, the best non-adaptive spreading strategy is to order all areas in non-increasing order of their initial acceptance probabilities.

We also study the problem of designing the optimum spreading strategy in the partial propagation setting with more than one types. We show it is hard to determine the expected number of adopters for a given spreading strategy. Formal speaking, we show it is $\#P$-complete to compute the expected number of adopters for a given spreading strategy π in the partial propagation setting with more than one type. This is another evidence to show the influence propagation is more complicated with more than one type. We prove Theorem 2.1.3 based on a reduction from a variation of the *network reliability* problem in Section 2.6.

In the partial propagation setting, it is $\#P$-complete to compute the expected number of adopters for a given non-adaptive spreading strategy π.

We also present a polynomial-time algorithm to compute the expected number of adopters for a given non-adaptive spreading strategy in a full propagation setting. We design an algorithm in order to simulate the amount of propagation for a given spreading strategy in Section 2.7. Consider a full propagation setting. The expected number of adopter can be computed in polynomial time for a given non-adaptive spreading strategy π.

At last we study the problem of designing the best adaptive spreading strategy. We overcome the hardness of the problem and design a polynomial-time algorithm to find the best adaptive marketing strategy in the following theorem. We describe the algorithm precisely in Section 2.8. A polynomial-time algorithm finds the best adaptive spreading strategy for a society with a constant number of types.

2.2 Notation and Preliminaries

In this section we define basic concepts and notation used throughout this chapter. We first formally define the spread of influence through a network as a stochastic process and then give the intuition behind the formal notation. We are given a graph $G = (V, E)$ with thresholds, $c_v \in \mathbb{Z}_{>0}, \forall v \in V$ and initial acceptance probabilities $p_v \in [0, 1], \forall v \in V$. Let $|V| = n$. Let d_v be the degree of vertex v. Let $N(v)$ be the set of neighboring vertices of v. Let $\boldsymbol{c}$ be the vector $(c_1, \ldots, c_n)$ and $\boldsymbol{p}$ be the vector $(p_1, \ldots, p_n)$. Given a graph $G = (V, E)$ and a permutation $\pi : V \mapsto V$, we define a discrete stochastic process, *IS* (Influence Spread) as an ordered set of random variables $(X^1, X^2, \ldots, X^n)$, where $X^t \in \Omega = \{-1, 0, 1\}^n, \forall t \in \{1, \ldots, n\}$. The random variable X_v^t denotes decision of area v at time t. If it has not yet been scheduled, $X_v^t = 0$. If it accepts the idea then $X_v^t = 1$, and if it rejects the idea then $X_v^t = -1$. Note that $X_v^t = 0$ iff $t < pi^{-1}(v)$. Let $D(v) = \sum_{u \in N(v)} X_u^{\pi^{-1}(v)}$ be the sum of decision's of v's neighbors. For simplicity in notation, we denote X_v^n by X_v.

We now briefly explain the intuition behind the notation. The input graph models the influence network of areas on which we want to schedule a cascade, with each vertex representing an area. There is an edge between two vertices if two corresponding areas influence each others decision. The influence spread process models the spread of idea acceptance and rejection for a given spreading strategy. The permutation π maps a position in spreading strategy to an area in V. For example, $\pi(1) = v$ implies that v is the first area to be scheduled. Once the area v is given a chance to accept or reject the idea at time $\pi^{-1}(v)$, $X_v^{\pi^{-1}(v)}$ is assigned a value based on v's decision and at all times t after $\pi^{-1}(v)$, $X_v^t = X_v^{\pi^{-1}(v)}$. The random variable X_v denotes whether an area v accepted or rejected the idea. We note that $X_v^t = X_v, \forall t \geq \pi^{-1}(v)$. The random variable X^t is complete snapshot of the cascade process at time t. The variable $D(v)$ is the decision variable for v. It denotes the sum of decisions of v's neighbors at the time v is scheduled in the cascade and it determines whether v decides to follow the majority decision or whether v decides based on its initial acceptance probability. The random variable I_t is the sum of decisions of all areas at time t. Thus, I_n is the variable we are interested in as it denotes the difference between number of people who accept the idea and people who reject the idea.

Let $v = \pi(t)$. Given X^{t-1}, X^t is defined as follows:

- Every area decides to accept or reject the idea exactly once when it is scheduled and its decision remains the same at all later times. Therefore $\forall i \neq \pi(t)$:
  - $X_i^t = X_i^{t-1}$
- Decision of area v is based on decision of previous areas if its threshold is reached.
  - $X_v^t = 1$ if $D(v) \geq c_v$
  - $X_v^t = -1$ if $D(v) \leq -c_v$
- If threshold of area v is not reached, then it decides to accept the idea with probability p_v, its initial acceptance probability, and decides to reject it with probability $1 - p_v$.

In partial propagation setting, we represent such a stochastic process by tuple $IS = (G, \boldsymbol{c}, \boldsymbol{p}, \pi)$. For full propagation setting, the underlying graph is a complete graph and hence we can denote the process by $(\boldsymbol{c}, \boldsymbol{p}, \pi)$. When $\boldsymbol{c}$ and $\boldsymbol{p}$ are clear from context, we denote the process simply by spreading strategy, π. We define random variable $I_t = \sum_{v \in V} X_v^t$. We denote by $q_v = 1 - p_v$ the probability that v rejects the idea based on initial preference. We denote by $Pr(A; IS)$, the probability of event A occurring under stochastic process IS. Similarly, we denote by $E(z; IS)$, the expected value of random variable z under the stochastic process IS.

2.3 A Bound on Spread of Appealing and Unappealing Ideas

Lets call an idea unappealing if its initial acceptance probability for all areas is p for some $p \leq \frac{1}{2}$. We prove in this section, that for such ideas, no strategy can boost the acceptance probability for any area above p. We note that exactly the opposite argument can be made when $p \geq \frac{1}{2}$ is the initial acceptance probability of all areas, i.e., any spreading strategy guarantees that every area accepts the idea with probability of at least p. Consider an arbitrary non-adaptive spreading strategy π in the full propagation setting. Assume all initial acceptance probabilities are equal to p. If $p \geq \frac{1}{2}$, then the expected number of adopters is at least np. Furthermore, If $p \leq \frac{1}{2}$, then the expected number of adopters is at most np.

Proof. We prove this result for the case when $p \leq \frac{1}{2}$. The other case ($p \leq \frac{1}{2}$) follows from symmetry.

To avoid confusion, we let $p_0 = p$ and use p_0 instead of the real number p throughout this proof. If we prove that any given area accepts the idea with probability of at most p_0, then from linearity of expectation, we are done. Consider an area v scheduled at time $t+1$. The probability that the area accepts or rejects the idea is given by:

$$Pr(X_v = 1) = p_0(1 - Pr(I_t \geq c_v) - Pr(I_t \leq -c_v)) + Pr(I_t \geq c_v)$$
$$Pr(X_v = -1) = (1 - p_0)(1 - Pr(I_t \geq c_v) - Pr(I_t \leq -c_v)) + Pr(I_t \leq -c_v)$$

Since $Pr(X_v = 1) + Pr(X_v = -1) = 1$, if we prove that $\frac{Pr(X_v=1)}{Pr(X_v=-1)} \leq \frac{p_0}{1-p_0}$, then we have $Pr(X_v = 1) \leq p_0$. We have:

$$\frac{Pr(X_v = 1)}{Pr(X_v = -1)} = \frac{p_0(1 - Pr(I_t \geq c_v) - Pr(I_t \leq -c_v)) + Pr(I_t \geq c_v)}{(1 - p_0)(1 - Pr(I_t \geq c_v) - Pr(I_t \leq -c_v)) + Pr(I_t \leq -c_v)}$$

We have:

$$\frac{p_0(1 - Pr(I_t \geq c_v) - Pr(I_t \leq -c_v))}{(1 - p_0)(1 - Pr(I_t \geq c_v) - Pr(I_t \leq -c_v))} = \frac{p_0}{1 - p_0}$$

We know that for any $a, b, c, d, e \in \mathbb{R}_{>0}$, if $\frac{a}{b} \leq e$ and $\frac{c}{d} \leq e$, then:

$$\frac{a + c}{b + d} \leq e \tag{2.1}$$

Therefore, if we prove that $\frac{Pr(I_t \geq c_v)}{Pr(I_t \leq -c_v)} \leq \frac{p_0}{1-p_0}$, we are done. Thus, we can prove this theorem by proving that $\frac{Pr(I_k \geq x)}{Pr(I_k \leq -x)} \leq \frac{p_0}{1-p_0}$ for all $x \in \{1 \ldots k\}, k \in \{1 \ldots n\}$. We prove this by induction on number of areas. If there is just one area, then that area decides to accept with probability p_0 (as all initial acceptance probabilities are equal to p_0). Assume if the number of areas is less than or equal to n, then $\frac{Pr(I_k \geq x)}{Pr(I_k \leq -x)} \leq \frac{p_0}{1-p_0}$ for all $x \in \{1 \ldots k\}, k \in \{1 \ldots n\}$. We prove the statement when there are $n+1$ areas. Let $par(n, x) : \mathbb{N} \times \mathbb{N} \mapsto \{0, 1\}$ be a function which is 0 if n and x have the same parity, 1 otherwise. Let v be the area scheduled at time $n+1$. Let $\nu = par(n, x)$. We now consider the following three cases.

Case 1: $1 \leq x \leq n - 2$. The event $I_{n+1} \geq x + 1$ is the union of the following two disjoint events:

1. $I_n \geq x+2$, and whatever the n^{th} area decides, I_{n+1} is at least $x+1$.

2. $I_n = x+\nu$ and $n+1^{th}$ area decides to accept.

Similarly, the event $I_{n+1} \leq -x-1$ is the union of the event $I_n \leq -x-2$ and the event — $I_n = -x-\nu$ and the $n+1^{th}$ area rejects the idea. We note that we require the par function because only one of the events $I_n = x$ and $I_n = x+1$ can occur w.p.p. depending on parities of n and x. Thus:

$$Pr(I_{n+1} \geq x+1) = Pr(I_n \geq x+2) + Pr(X_v = 1 | I_n = x+\nu) Pr(I_n = x+\nu)$$
$$Pr(I_{n+1} \leq -x-1) = Pr(I_n \leq -x-2) + Pr(X_v = -1 | I_n = -x-\nu) Pr(I_n = -x-\nu)$$

Now, if $x+\nu \geq c_v$, then $Pr(X_v = 1|I_n = x+\nu) = Pr(X_v = -1|I_n = -x-\nu) = 1$, otherwise $Pr(X_v = 1|I_n = x+\nu) = p_0 < 1-p_0 = Pr(X_v = -1|I_n = -x-\nu)$. Therefore, $Pr(X_v = 1|I_n = x+\nu) \leq Pr(X_v = -1|I_n = -x-\nu)$. Let $\beta = Pr(X_v = -1|I_n = -x-\nu)$. Using the above, we have:

$$Pr(I_{n+1} \geq x+1) \leq Pr(I_n \geq x+2) + \beta Pr(I_n = x+\nu)$$
$$Pr(I_{n+1} \leq -x-1) = Pr(I_n \leq -x-2) + \beta Pr(I_n = -x-\nu)$$

From above, we have:

$$f(\beta) = \frac{Pr(I_n \geq x+2) + \beta Pr(I_n = x+\nu)}{Pr(I_n \leq -x-2) + \beta Pr(I_n = -x-\nu)} \geq \frac{Pr(I_{n+1} \geq x+1)}{Pr(I_{n+1} \geq -x-1)} \quad (2.2)$$

The function $f(\beta)$ is either increasing or decreasing and hence has extrema at end points of its range. The maxima is $\leq \max\{\frac{Pr(I_n \geq x+2)}{Pr(I_n \leq -x-2)}, \frac{Pr(I_n \geq x+2)+Pr(I_n = x+\nu)}{Pr(I_n \leq -x-2)+Pr(I_n = -x-\nu)}\}$ because $\beta \in [0,1]$. Now $Pr(I_n \geq x+2) + Pr(I_n = x+1) + Pr(I_n = x) = Pr(I_n \geq x)$ and $Pr(I_n \leq -x-2) + Pr(I_n = -x-\nu) = Pr(I_n \leq -x)$. Thus $f \leq \max\{\frac{Pr(I_n \geq x+2)}{Pr(I_n \leq -x-2)}, \frac{Pr(I_n \geq x)}{Pr(I_n \leq -x)}\} \leq \frac{p_0}{1-p_0}$ (from induction hypothesis). From above and (2.2), $\frac{Pr(I_{n+1} \geq x+1)}{Pr(I_{n+1} \leq -x-1)} \leq \frac{p_0}{1-p_0}$.

Case 2: $x = 0$. If n is odd then $Pr(I_{n+1} \geq 1) = Pr(I_{n+1} \geq 2)$ and $Pr(I_{n+1} \leq -1) = Pr(I_{n+1} \leq -2)$ and this case is same as $x = 1$ and hence considered above. Thus, assume that n is even.

$$Pr(I_{n+1} \geq 1) = Pr(I_n \geq 2) + Pr(X_v = 1|I_n = 0)Pr(I_n = 0) \quad (2.3)$$

$$Pr(I_{n+1} \leq -1) = Pr(I_n \leq -2) + Pr(X_v = -1|I_n = 0)Pr(I_n = 0) \quad (2.4)$$

Since, if $I_n = 0$, then areas decide based on the initial acceptance probability. We have $Pr(X_v = 1|I_n = 0) = p_0$ and $Pr(X_v = -1|I_n = 0) = 1 - p_0$. Using this fact ,by dividing (2.3) and (2.4), we have:

$$\frac{Pr(I_{n+1} \geq 1)}{Pr(I_{n+1} \leq -1)} \leq \frac{Pr(I_n \geq 2) + p_0 Pr(I_n = 0)}{Pr(I_n \leq -2) + (1 - p_0)Pr(I_n = 0)}$$

From induction hypothesis, $\frac{Pr(I_n \geq 2)}{Pr(I_n \leq -2)} \leq \frac{p_0}{1-p_0}$. Thus, we conclude $\frac{Pr(I_{n+1} \geq 1)}{Pr(I_{n+1} \leq -1)} \leq \frac{p_0}{1-p_0}$ based on (2.1).

Case 3: $x \in \{n-1, n\}$. In this case $Pr(I_n \geq x + 2) = 0$, since the number of adopters can never be more than the number of total areas. Also, I_{n+1} cannot be equal to n because n and $n + 1$ don't have the same parity. Therefore, $Pr(I_{n+1} \geq n) = Pr(I_{n+1} \geq n + 1)$ and $Pr(I_{n+1} \leq -n) = Pr(I_{n+1} \leq -n - 1)$. Thus, it is enough to analyze the case $x = n$. We have:

$$Pr(I_{n+1} \geq n + 1) = Pr(X_v = 1|I_n = n)Pr(I_n = n)$$

$$Pr(I_{n+1} \leq n + 1) = Pr(X_v = -1|I_n = -n)Pr(I_n = -n)$$

Since either both decisions are made based on thresholds with probability 1 or both are made based on initial probabilities and initial acceptance probability is less than the initial rejection probability, We know that $Pr(X_v = 1|I_n = n) \leq Pr(X_v = -1|I_n = -n)$. Therefore $\frac{Pr(I_{n+1} \geq n+1)}{Pr(I_{n+1} \leq n+1)} \leq \frac{Pr(I_n = n)}{Pr(I_n = -n)}$. Now, since $Pr(I_n = n) =$

$Pr(I_n \geq n)$ and $Pr(I_n = -n) = Pr(I_n \leq -n)$, from induction hypothesis, we have $\frac{Pr(I_{n+1} \geq n+1)}{Pr(I_{n+1} \leq n+1)} \leq \frac{p_0}{1-p_0}$ and we are done. $\square$

2.4 Non-adaptive Marketing Strategy with Random Thresholds

We consider the problem of designing a non-adaptive spreading strategy when the thresholds are drawn independently from the same but unknown distribution. We show the best spreading strategy is to schedule areas in a non-increasing order of initial acceptance probabilities. We prove the optimality of the algorithm using a coupling argument. First we state and prove the following lemma which will be useful in proving Theorem 2.1.3.

Lemma 1. *Let π and π' be two spreading strategies. If $\exists k \in \mathbb{Z}_{>}0$, such that $\pi(i) = \pi'(i)$, $\forall i \geq k$ and $Pr(I_k \geq x; \pi) \geq Pr(I_k \geq x; \pi')$, $\forall x \in \mathbb{Z}$, then $E(I_n; \pi) \geq E(I_n; \pi')$.*

Proof. We prove this lemma by proving that:

$$Pr(I_{k+t} \geq x; \pi) \geq Pr(I_{k+t} \geq x; \pi'),\ \forall t \in \{1 \dots n-k\} \tag{2.5}$$

We note that the above implies $E(I_n; \pi) \geq E(I_n; \pi')$. We prove that if $Pr(I_k \geq x; \pi) \geq Pr(I_k \geq x; \pi')$ then $Pr(I_{k+1} \geq x; \pi) \geq Pr(I_{k+1} \geq x; \pi')$ for all $x \in \mathbb{Z}$. This argument can be successively applied to prove (2.5). Let $\pi(k+1) = v$. X_v will be 1 iff either $I_k \geq c_v$ and v accepts idea based on threshold rule or $-c_v < I_k < c_v$ and v decides to accept the idea based on initial acceptance probability p_v. Thus:

$$Pr(X_v = 1) = Pr(I_k \geq c_v) + Pr(-c_v < I_k < c_v)p_v$$

Substituting $Pr(-c_v < I_k < c_v) = Pr(I_k \geq -c_v + 1) - Pr(I_k \geq c_v)$, we have:

$$Pr(X_v = 1) = Pr(I_k \geq c_v) + (Pr(I_k \geq -c_v + 1) - Pr(I_k \geq c_v))p_v$$

By rearranging the terms, we get:

$$Pr(X_v = 1) = Pr(I_k \geq c_v)(1 - p_v) + Pr(I_k \geq -c_v + 1)p_v \tag{2.6}$$

We are given that $Pr(I_k \geq x; \pi) \geq Pr(I_k \geq x; \pi'),\ \forall x \in \mathbb{Z}$. From this and from (2.6), we have, $Pr(X_v = 1; \pi) \geq Pr(X_v = 1; \pi')$. Thus, $Pr(I_{k+1} \geq x; \pi) \geq Pr(I_{k+1} \geq x; \pi'),\ \forall x \in \mathbb{Z}$. □

Assume that the planner's prior knowledge about all values of c_i's is the same, i.e., all c_i's are drawn independently from the same but unknown distribution. Let initial acceptance probabilities be arbitrary numbers. Then, the best non-adaptive spreading strategy is to order all areas in non-increasing order of their initial acceptance probabilities.

Proof. Let π' be a spreading strategy where areas are scheduled in an order that is not non-increasing. Thus, there exists k such that $p_{\pi'(k)} < p_{\pi'(k+1)}$. We prove that if a new spreading strategy π is created by exchanging position of areas $\pi'(k)$ and $\pi'(k+1)$, then the expected number of people who accept the idea cannot decrease. It means the best spreading strategy is non-increasing in the initial acceptance probabilites.

To prove the theorem, we will prove that $Pr(I_{k+1} \geq x; \pi) \geq Pr(I_{k+1} \geq x; \pi')$ and the result then follows from Lemma 1. Since, the two spreading strategies are identical till time $k-1$ and therefore the random variable I_{k-1} has identical distribution under both the strategies, we can prove the above by proving that $Pr(I_{k+1} \geq I_{k-1} + y | I_{k-1}; \pi) \geq Pr(I_{k+1} \geq I_{k-1} + y | I_{k-1}; \pi')$ for all $y \in \mathbb{Z}$. We note that the only feasible values for y are in $\{-2, 0, 2\}$. Hence, if $y > 2$ then both sides of the above inequality are equal to 1 and the inequality holds. Similarly, if $y <= -2$ both sides of the inequality are equal to 1 and the inequality holds. Thus, we only need to analyze the values $y = 0$ and $y = 2$.

Now we define some notation to help with rest of the proof. Let $u = \pi'(k+1)$, $v = \pi'(k)$, and $q_i = 1 - p_i$. It means $p_v < p_u$. Let $\chi(i, j)$ be the event where i and j are indicators of decision of areas scheduled at time k and $k+1$ respectively, e.g., $\chi(1, 1)$ means that areas scheduled at time k and $k+1$ accepted the idea, whereas $\chi(1, -1)$ implies that area scheduled at time k accepted the idea, while the area scheduled at time $k+1$ rejected the idea. Let $B(y)$ be the event $I_{k+1} \geq I_{k-1} + y | I_{k-1} = z$ for some arbitrary $z \in \mathbb{Z}$. We consider the cases $I_{k-1} > 0$, $I_{k-1} < 0$ and $I_{k-1} = 0$ separately.

Case 1: $I_{k-1} = z, z > 0$. We have, $B(0) = \chi(1, 1) \cup \chi(1, -1) \cup \chi(-1, 1) = \chi(-1, -1)^c$. Since we assume $z > 0$, the thresholds $-c_u$ and $-c_v$ cannot be hit. Thus, $\chi(-1, -1)$ occurs only when both areas decide to reject the idea based on their respective initial acceptance probabilities. Thus, from chain rule of probability, it is the product of following four terms:

1. $Pr(z < c_u)$, i.e, the threshold rule does not apply and u decides based on initial acceptance probabilities.

2. u rejects the idea based on initial probability of rejection, q_u.

3. $Pr(z - 1 < c_v)$. Given u rejected the idea, $D(v)$, the decision variable for v becomes $z - 1$ and the threshold rule does not apply and v decides based on initial acceptance probabilities.

4. v rejects the idea based on initial probability of rejection, q_v.

Therefore, $Pr(\chi(-1, -1)) = Pr(z < c_u) q_u Pr(z - 1 < c_v) q_v$. Thus, $Pr(B(0); \pi) = 1 - Pr(z < c_u) q_u Pr(z - 1 < c_v) q_v$. Since, c_u and c_v are i.i.d random variables, we can write any probability of form $Pr(z \gtreqless c_u)$ or $Pr(z \gtreqless c_v)$ as $Pr(z \gtreqless x)$, where x

is an independent random variable with the same distribution as c_u and c_v. Thus,

$$Pr(B(0); \pi) = 1 - Pr(z < x)q_u Pr(z - 1 < x)q_v \tag{2.7}$$

Now, $Pr(\chi(1,1)) = Pr(X_u = 1|I_{k-1} = z)Pr(X_v = 1|I_k = z+1)$. Event $X_u = 1$ is the union of following two non-overlapping events:

1. $z \geq c_u$; u accepts the idea because of the threshold rule.

2. $z < c_u$ and u accepts the idea based on initial acceptance probability, p_u.

Thus, $Pr(X_u = 1|I_{k-1} = z) = Pr(z \geq c_u) + Pr(z < c_u)p_u$. Similarly, $Pr(X_v = c_v|I_k = z+1) = Pr(z+1 \geq c_v) + Pr(z+1 < c_v)p_v$. Therefore:

$$Pr(B(2); \pi) = (Pr(z \geq x) + Pr(z < x)p_u)(Pr(z+1 \geq x) + Pr(z+1 < x)p_v) \tag{2.8}$$

Where, we have replaced c_u and c_v by x because they are i.i.d. random variables. We can obtain corresponding probabilities for process π' by exchanging p_u and p_v. Thus, $Pr(B(0); \pi) = Pr(B(0); \pi') = 1 - Pr(z < x)q_u Pr(z - 1 < x)q_v$. We can write $Pr(B(2); \pi')$ as follows:

$$Pr(B(2); \pi') = (Pr(z \geq x) + Pr(z < x)p_v)(Pr(z+1 \geq x) + Pr(z+1 < x)p_u) \tag{2.9}$$

On the other hand $Pr(z < x) \geq Pr(z+1 < x)$ and $Pr(z+1 \geq x) \geq Pr(z \geq x)$. Comparing (2.8) and (2.9) along with these facts that $p_v < p_u$ and $Pr(z < x)Pr(z+1 \geq x) \geq Pr(z \geq x)Pr(z+1 < x)$, we get $Pr(B(2); \pi) \geq Pr(B(2); \pi')$.

Case 2: $I_{k-1} = -z, z > 0$s. By a similar analysis, we have:

$$Pr(B(2); \pi) = Pr(z < x)Pr(z - 1 < x)p_u p_v = Pr(B(2); \pi') \tag{2.10}$$

$$Pr(B(0); \pi) = 1 - (Pr(z \geq x) + Pr(z < x)q_u)(Pr(z + 1 \geq x) + Pr(z + 1 < x)q_v) \tag{2.11}$$

$$Pr(B(0); \pi') = 1 - (Pr(z \geq x) + Pr(z < x)q_v)(Pr(z + 1 \geq x) + Pr(z + 1 < x)q_u) \tag{2.12}$$

Comparing (2.11) and (2.12), we have $Pr(B(0); \pi) \geq Pr(B(0); \pi')$.

Case 3: $I_{k-1} = 0$. We have:

$$Pr(B(2); \pi) = p_u(Pr(x > 1)p_v + Pr(x = 1)) \tag{2.13}$$

$$Pr(B(0); \pi) = p_u + q_u Pr(x > 1)p_v \tag{2.14}$$

$$Pr(B(2); \pi') = p_v(Pr(x > 1)p_u + Pr(x = 1)) \tag{2.15}$$

$$Pr(B(0); \pi') = p_v + q_v Pr(x > 1)p_u \tag{2.16}$$

By comparing (2.13) with (2.15) and (2.14) with (2.16), we see that $Pr(B(2); \pi) \geq Pr(B(2); \pi')$ and $Pr(B(0); \pi) \geq Pr(B(0); \pi')$ respectively. Thus, $Pr(I_{k+1} \geq I_{k-1} + x | I_{k-1}; \pi) \geq Pr(I_{k+1} \geq I_{k-1} + x | I_{k-1}; \pi'), \forall x \in \mathbb{Z}$. □

2.5 Type Switching Approach

Consider a society with a constant number of types. One approach that might work is an algorithm that finds an optimal spreading strategy allowing for only a constant number of switches between types in a spreading strategy. We note that areas of the same type are identical from point of view of scheduling a cascade. Thus, any non-adaptive spreading strategy can be specified by specifying types of areas rather than the areas themselves. Let τ be the mapping between an area and its type. That is

$\tau(i)$ is the type of area i. Let λ be sequence of types for a given spreading strategy. Specifically, λ is a vector whose k^{th} component, $\lambda(k) = \tau(\pi(k))$. A switch is any position k in the sequence λ such that $\lambda(k) \neq \lambda(k+1)$. As an example, consider a society with four areas with two areas of type 1 and two areas of type 2. Then the type sequence $\lambda = (1, 1, 2, 2)$ has a switch at position 2 whereas $\lambda_2 = (1, 2, 1, 2)$ has switches at positions 1, 2 and 3. We define a σ-switch spreading strategy as a non-adaptive spreading strategy that has at most σ switches, where σ is a constant independent of input size. We now prove that no algorithm whose output is a σ-switch spreading strategy can be optimal.

A σ-switch spreading strategy is a spreading strategy with at most σ switches. For any constant σ, there exists a society with areas of two types such that no σ-switch spreading strategy is optimal.

Proof. The proof outline is as follows. We construct an instance of problem with $2n$ areas with two types, the number of areas of both types being n, for which an optimal spreading strategy alternates between these types. Lets call this instance S and lets call this strategy π. We prove that the expected number of adopters achieved by this optimal strategy is upper bound on number of acceptors for any input instance with areas of these two types, whatever be the number of areas of both types, given that total number of areas is $2n$, e.g., the number of areas of one type can be n_1 and the other type $2n - n_1$ for any integer n_1 between 0 and $2n$ and no strategy for this instance can exceed the expected number of adopters achieved by π for the instance of problem with n areas of each type. We then show that any σ-switch strategy for instance S of problem can be improved by changing type of one of the areas. Since, the optimal value achieved by this new strategy cannot be greater than strategy π on instance S, no σ-switch strategy can be optimal.

Consider an instance with two types $\gamma_1 = (P, 1)$ and $\gamma_2 = (P, 2)$ where $P > \frac{1}{2}$, the total number of areas is $2n$ and the number of areas of types γ_1 and γ_2 is n

each. Let π be a spreading strategy for which the type sequence of areas is given by $\lambda = (\gamma_1, \gamma_2, \ldots, \gamma_1, \gamma_2)$, i.e., every area at odd position is of type γ_1 and every area at even position is of type γ_2. Let the expected number of areas which accept the idea for this spreading strategy be α. Now consider an instance where the total number of areas is the same but the number of areas of type γ_1 is n_1 and number of areas of type γ_2 is $2n - n_1$ for some arbitrary natural number n_1 such that $0 \leq n_1 \leq n_2$. For this instance, let the expeted number of areas which accept the idea given an optimal spreading strategy be β. We now prove that $\alpha \geq \beta$. If we have no restriction on the number of areas of each type, then for any $t = 0 \mod 2$, the areas to be scheduled at time $t+1$ and $t+2$ can be of types (γ_1, γ_1), (γ_1, γ_2), (γ_2, γ_1) or (γ_2, γ_2). We prove that $\alpha \geq \beta$ by proving that it is better to schedule areas of type γ_1 and γ_2 at times $t+1$ and $t+2$ respectively. If $|I_t| \geq 2$, then we are indifferent between all spreading strategies because in this case all the areas will decide based on the threshold rule. Thus, if we can prove that (γ_1, γ_2) is a best choice for types at times $t+1$ and $t+2$ when $|I_t| < 2$, we are done. Since t is even, the only feasible value of $|I_t| \leq 2$ is $I_t = 0$. Thus, this is the only case we need to analyze. Let ρ be the tuple of types of areas scheduled at times $t+1$ and $t+2$. Let χ be the tuple indicating decisions of areas scheduled at times $t+1$ and $t+2$. Now we analyze the probabilties with which the four possible values of χ are realized for each of the four possible values of ρ when $I_t = 0$. Let number of areas to be scheduled after time t be m.

Case 1: $\rho = (\gamma_1, \gamma_1)$ or (γ_2, γ_1)

In this case, the first area decides based on its initial acceptance probability and the second area follows the decision of the first area.

$$Pr(\chi = (1,1)) = P$$
$$Pr(\chi = (1,-1)) = 0$$
$$Pr(\chi = (-1,1)) = 0$$
$$Pr(\chi = (-1,-1)) = 1 - P$$

The expected number of areas which accept the idea after time t in this case is mP, as all areas follow the decision of area scheduled at time $t + 1$.

Case 2: $\rho = (\gamma_1, \gamma_2)$ or (γ_2, γ_2)

In this case, both the areas decide based on their initial acceptance probability.

$$Pr(\chi = (1,1)) = P^2 \tag{2.17}$$
$$Pr(\chi = (1,-1)) = P(1 - P) \tag{2.18}$$
$$Pr(\chi = (-1,1)) = P(1 - P) \tag{2.19}$$
$$Pr(\chi = (-1,-1)) = (1 - P)^2 \tag{2.20}$$

From (2.17), with probability P^2, all areas after time t will accept the idea. If for any time t', we are given that $I_{t'} = 0$, then we can treat the subsequent areas as the starting point of a new spreading strategy. Thus, if $I_{t+2} = 0$, then from Theorem 2.1.3 (given that $P > \frac{1}{2}$), the expected number of adopters for any future spreading strategy is at least $(m - 2)P$. Hence, from (2.18) and (2.19), with probability $2P(1 - P)$ the expected number of areas that will accept after time t is at least $1 + (m - 2)P$. Therefore, in this case, the expected number of areas that accept after time t is at least $mP^2 + 2P(1-P)(1+(m-2)P)$. Thus, we are done if we prove that $mP^2 + 2P(1-$

$P)(1+(m-2)P)$ is greater than mP.

$$mP^2 + 2P(1-P)(1+(m-2)P) - mP = P(1-P)(-m+2(1+(m-2)P))$$

Thus, it is enough to prove that $2(1+(m-2)P) - m > 0$. We have:

$$2(1+(m-2)P) - m = (2P-1)(m-2)$$

Since $P > \frac{1}{2}$, $2P - 1 > 0$. Thus, for all $m > 2$, it is strictly better to schedule an area of type γ_2 at time $t+2$. If an area of type γ_2 is scheduled at time $t+2$, then it is equivalent to schedule an area of either type at time $t+1$. Thus, given that there is at least one more area to follow at time $t+3$, it is best to schedule areas of type γ_1 and γ_2 respectively at times $t+1$ and $t+2$ at any arbitrary time $t = 0 \mod 2$. Also, such a schedule is strictly better, all other things begin same, than the schedule where, areas of type γ_1 are scheduled at times $t+1$ and $t+2$. This fact is important as we use this later in the proof. If there are no more areas to follow, then we are indifferent to all the four options. Hence, the expected number of adopters achieved by π is an upper bound on number of acceptors for any input instance with areas of these two types whatever be the number of areas of both types

The final part of this proof is by contradiction. Let the the number of areas in the input instance of problem be $2n$ with n areas each of types $\gamma_1 = (P, 1)$ and $\gamma_2 = (P, 2)$. Consider a σ-switch strategy. Choose $n \geq 4(\sigma + 1)$. Thus, every σ-switch strategy will have at least four consecutive areas of type γ_1. Let a σ-switch strategy, π', be an optimal one. Therefore, there will exist a time t in π' such that $t = 0 \mod 2$, $\tau(\pi'(t+1)) = \gamma_1$, $\tau(\pi'(t+2)) = \gamma_1$ and at least one more area will be scheduled after time $t+2$. As explained earlier, the expected number of adopters in this case is strictly less than expected number of adopters if we schedule an area of type γ_2 at time $t+2$, which, as proved above, is at most the expected number of adopters for a strategy with type sequence $(\gamma_1, \gamma_2, \ldots, \gamma_1, \gamma_2)$. Therefore, strategy π is not optimal. This is a contradiction and no σ-switch strategy can be optimal for the given instance. □

2.6 Hardness Result

We prove that problem of computing expected number of adopters for a given spreading strategy in the partial propagation setting is $\#P$-complete. This result applies even when the input graphs are planer with a maximum degree of 3 and have only 4 different types of vertices. We prove this by reduction from a version of the network reliability problem that is known to be $\#P$-complete ([110]). In the network reliability problem, a directed graph G and probability $0 \leq p \leq 1$ are given. Nodes fail independently with probability $1-p$. Therefore, each node is present in the surviving subgraph with probability p. We achieve the reduction by simulating the $s-t$ network reliability problem by designing an instance of cascade scheduling problem where, probability of an area v accepting an idea is exactly equal to a path existing in the surviving sub-graph from the source to vertex v. Before proceeding to details of the proof, we give some definitions below.

Definition 1. *Given a directed graph G with source s, terminal t, and a probability $1-p, 0 \leq p < 1$ of nodes failing independently, the (s,t)*-connectedness reliability *of G, $R(G, s, t; p)$, is defined as the probability that there is at least one path from s to t such that none of the vertices falling on the path have failed.*

Definition 2. AST *is the problem of computing $R(G, s, t; p)$ when G is an acyclic directed (s,t)-planar graph with each vertex having degree at most three. We denote an instance of AST on graph G as $AST(G, s, t, p)$.*

Definition 3. *Given an influence spread process, $S = (G, \mathbf{c}, \mathbf{p}, \pi)$ on G with a source node s and a target node t,* IST *is the problem of computing $Pr(X_t = 1; S)$ given that $\pi(1) = s$ and $Pr(X_s = 1) = 1$. We denote an instance of IST by $IST(G, \mathbf{c}, \mathbf{p}, \pi, s, t)$.*

We will reduce an instance of AST to an instance of IST (Probability of Influence Spread to T).

Given an instance of AST, $AST(G = (V, E), s, t, p)$ we now construct an instance of IST, $IST(G' = (V', E'), \mathbf{c}, \mathbf{p}, \pi, s, t)$ for which $R(G, s, t; p) = Pr(X_t = 1)$. Let d_v^{in} be the indegree of $v \in V$ in G. For every vertex $v \in V - \{s\}$, we add three vertices to graph G'. Lets denote them by b_v, the blocking vertex of v, f_v, the forwarding vertex for v and v', which corresponds to the original vertex v. The rationale for nomenclature will become apparent later. For every edge (u, v) in E, we add an edge $\{u', b_v\}$ in E'. In addition, we add edges $\{b_v, v'\}$ and $\{f_v, v'\}$ to E'. The acceptance probabilities and thresholds are set as follows: $p_{v'} = 0, p_{f_v} = p, p_{b_v} = 1 \; \forall v \in V - \{s\}, p_{s'} = p$. $c_v = 2, c_{b_v} = d_v^{in} \; \forall v \in V - \{s\}$. Threshold $c_{s'}$ is irrelevant and can be any arbitrary value greater than 0 since it is the first vertex to be scheduled. Thresholds c_{f_v} can also be any arbitrary value greater than 0 since no neighbor of f_v is scheduled before f_v. Let $\pi' : V \mapsto V$ be any topological ordering on V where, s is the first node and t is the last node. Then π is constructed as follows:

$$
\begin{aligned}
\pi^{-1}(s') =& 1 \\
\pi^{-1}(v') =& 3\pi'^{-1}(v) - 2 \; \forall v \in V - \{s\} \\
\pi^{-1}(b_v) =& 3\pi'^{-1}(v) - 4 \; \forall v \in V - \{s\} \\
\pi^{-1}(f_v) =& 3\pi'^{-1}(t) - 3 \; \forall v \in V - \{s\}
\end{aligned}
$$

The above construction of π can be interpreted as follows. Source remains the first vertex to be scheduled. A vertex v is split into three vertices — v', b_v and f_v. In place of v, these three vertices are consecutively scheduled in order b_v, f_v and v', e.g., if $\pi' = (s, v, t)$, then $\pi = (s', b_v, f_v, v', b_t, f_t, t')$.

Let IS be the influence spread process $(G', \mathbf{c}, \mathbf{p}, \pi)$. Now, we prove the following lemmas which relate the probability of existence of a path of operative vertices between

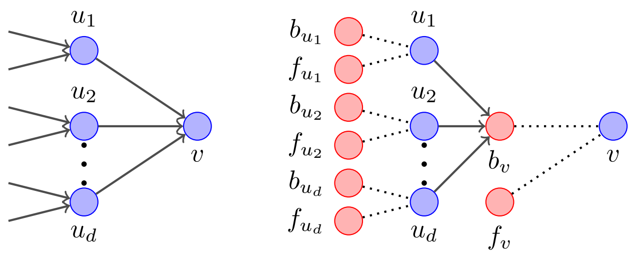


Figure 2.4: Reduction from Network Reliability on a DAG to Computing Expected Number of Influenced Nodes – The diagram on left is a part of DAG with probability of failure of each node equal to $(1 - p)$. The diagram on right is corresponding part of graph that represents an influence spread stochastic process the models the given network reliability problem where $p_{b_v} = 1$, $c_{b_v} = d$, $p_{f_v} = p$,$p_{v'} = 0$, and $c_{v'} = 2$.

s and v in G and the probability that area v accepts the idea in the influence spread process IS.

We first prove that computing the expecte number of vertices in graph to which s has a path with operating vertices is $\#P$-complete. We then use this to prove the main theorem.

Lemma 2. *Consider an instance of AST, $AST(G = (V, E), s, t, p)$. Then computing the expected number of vertices in graph to which s has a path with operating vertices is $\#P$-complete.*

Proof. Let $a(G, s)$ be the expected number of vertices in the graph to which s has a path with operating vertices in G. Let $b(G, s, t)$ be probability that there is a path of operating vertices from s to t in G. We note that t has no outgoing edges. Lets assume that $a(G, s)$ can be computed in time polynomial in $|G|$. Let $G' = G - \{t\}$. Deletion of t does not change probability of survival of any path whose destination is not t. Therefore $a(G', s) = \sum_{u \in V - \{t\}} b(G, s, u)$. Thus, $a(G, s) - a(G', s) = b(G, s, t)$. This is a contradiction because this implies that $b(G, s, t)$ can be computed in time polynomial in $|G|$. □

The proof of the main theorem of this section is organized as follows. We first prove that the probability of an area v' accepting an idea is exactly equal to probability of a path existing from s to v. Then, we use this fact along with Lemma 2 to prove the main result.   In the partial propagation setting, it is $\#P$-complete to compute the expected number of adopters for a given non-adaptive spreading strategy π.

Proof. Let $AST(G = (V, E), s, t, p)$ be an instance of AST problem. Let $S(G' = (V', E'), \mathbf{c}, \mathbf{p}, \pi)$ be an influence spread process with G', c_v, p_v and π as defined above. Then an area $v \neq s, t$ accepts the idea with probability p iff at least one of its predecessors in G also accepts the idea.

Let $P(v)$ be the set of predecessors of v in G. We note that in IS, by construction of π and G', vertices in $P(v)$ are exactly the neighbors of b_v that are scheduled before b_v. Area b_v is immediately followed by f_v and f_v by v. Also, by construction of G', b_v and f_v are neighbors of v and v has no other neighbors. Area f_v's only neighbor is v.

If no vertex in $P(v)$ accepts the idea, then $D(b_v) = -d_v^{in} = -c_{b_v}$ and thus, $Pr(b_v = -1|$ no vertex in $P(v)$ accepts the idea $) = 1$ and therefore, b_v rejects the idea. Since, threshold of v is $c_v = 2$, v decides based on threshold if and only if both its neighbors either accept or reject the idea. Therefore if b_v rejects the idea, then if f_v accepts the idea, then v does not accept the idea because it decides to reject the idea based on its initial acceptance probability as $p_v = 0$. If $X_{f_v} = -1$, then also v does not accept the idea because it reject the idea based on threshold rule, because both its neighbors rejected this idea. Thus, if none of the vertices in $P(v)$ accept the idea then v does not accept the idea.

If any area in $P(v)$ accepts the idea then $-c_{b_v} = -d_v^{in} < D(b_v) < d_v^{in} = c_{b_v}$ and b_v accepts the idea because its initial acceptance probability, $p_{b_v} = 1$. Now, if f_v accepts the idea then v also accepts because $c_v = 2$ and if f_v rejects the idea, then v does not accept the idea because it decides to reject it on basis of its initial acceptance probability, $p_v = 0$. Since, no neighbor of f_v is scheduled before f_v, f_v accepts the idea independently at random with its initial acceptance probability $p_{f_v} = p$. Therefore, given that at least one vertex in set $P(v)$ accepts the idea, v accepts the idea with probability p.

Now, by principal of deferred decisions, process of finding a path of operating vertices from s to t in the network reliability problem, can be simulated as follows. Let π be any topological ordering on vertices of G. Let $L(i)$ be the i^{th} layer (excluding layer containing just the source vertex, s) in topologically sorted G. Then probability that a path to $u \in L(1)$ exists is p because we let each vertex in this layer fail independently with probability $1-p$. For vertex v in any subsequent layer, if there exists a path to any of vertices in $P(v)$, the set of predecessors of v, then we let v fail independently with probability $1-p$. If no path to any of predecessors of v exists, then no path to v can exist and it is immaterial whether v fails or not. Thus, we let v fail with probability 1. As explained above, this is exactly the process simulated by $IS(G', c_v, p_v, \pi)$. Thus, computing $Pr(X_t = 1)$ is $\#P$-complete.

However, we need to prove hardness of computing $\Lambda = \sum_{u \in V'} Pr(X_u = 1)$. If we can prove that from Λ we can compute the expected number of vertices in graph to which s has a path, say $\alpha = \sum_{v \in V} Pr(X_{v'} = 1)$, then from Lemma 2, we are done.

Since $\forall v \in V, Pr(X_{v'} = 1) = Pr(X_{b_v} = 1) \cdot Pr(X_{f_v} = 1) = Pr(X_{b_v} = 1) \cdot p$ and $Pr(X_{f_v}) = p$, we have:

$$\Lambda = \sum_{v \in V} (Pr(X_{v'} = 1) + Pr(X_{b_v} = 1) + Pr(X_{f_v} = 1)) = \sum_{v \in V} (Pr(X_{v'} = 1) + \frac{Pr(X_{v'} = 1)}{p} + p)$$

From above, we can easily compute α. Hence, the claim follows. □

We note that AST is $\#P$-complete even when degrees of vertices of the input graph is constrained to be 3. Thus, indegree of a node (through which a path from s to t can pass) has to be 1 or 2. If p is the survival probability of a vertex in the AST problem instance, then the possible types of areas in the corresponding instance of IST are in $\{(1,1),(1,2),(p,1),(0,2)\}$, where the first two types correspond to blocking nodes in G, the forwarding nodes are of type $(p,1)$ and the vertices corresponding to original vertices are of type $(0,2)$. Thus, IST is hard on graphs with maximum degree constrained to 3 and number of types constrained to 4.

2.7 Computing Expected Number of Adopters

Here we give an algorithm to compute $E(I_n)$, given a spreading strategy π with thresholds given by vector $\boldsymbol{c}$ and initial probabilities of acceptance given by vector $\boldsymbol{p}$. Let Y_k be the number of 1 decisions among vertices in $\{\pi(1),\pi(2),\ldots,\pi(k)\}$. We note that $I_k = 2Y_k - k$. Since $E(I_n) = \sum_{i\in\{1\ldots n\}} xPr(I_n = x)$, we are interested in computing $Pr(I_n = x)$, $\forall x \in \{-n\ldots n\}$. Consider a full propagation setting. The expected number of adopter can be computed in polynomial time for a given non-adaptive spreading strategy π. Let A be a $n \times (2n+1)$ matrix where $A[k,x] = Pr(I_k = x), k \in \{1\ldots n\}, x \in \{-n\ldots n\}$. Let $v = \pi(k)$. The following recurrence might be used to arrive at a dynamic programming formulation:

$$A[k,x] \leftarrow Pr(X_v^k = 1)A[k-1,x-1] + Pr(X_v^k = -1)A[k-1,x+1]$$

However, one needs to be careful when computing $Pr(X_v^k = 1)$ because it is dependent of I_{k-1}. Thus, in the correct recurrence we must have $Pr(X_v^k = 1|I_{k-1} = x-1)$ and $Pr(X_v^k = -1|I_{k+1} = x+1)$ instead of $Pr(X_v^k = 1)$ and $Pr(X_v^k = -1)$ respectively. Below we derive the dynamic program keeping this subtelty in mind. Let

$v = \pi(k+1)$. We have:

$$Pr(I_{k+1} = x+1 | I_k = x) = \begin{cases} p_v & \text{if } -c_v < x < c_v \\ 1 & \text{if } x \geq c_v \\ 0 & \text{otherwise} \end{cases}$$
$$Pr(I_{k+1} = x-1 | I_k = x) = 1 - Pr(I_{k+1} = x+1 | I_k = x)$$

We have:

$$\begin{aligned} Pr(I_{k+1} = x) =& Pr(I_{k+1} = x | I_k = x-1) Pr(I_k = x-1) \\ &+ Pr(I_{k+1} = x | I_k = x+1) Pr(I_k = x+1) \end{aligned}$$

The above relation suggests a dynamic program for computing $E(I_n)$. The matrix A is initialized with $A[1,1] = p_{\pi(1)}, A[1,-1] = 1 - A[1,1], A[1,0] = 0, A[k,x] = 0,\ \forall x > k, A[k,x] = 0,\ \forall x < -k$. When $|x| < n, k > 1$, then any $A[k,x]$ depends on $A[k-1, x+1]$ and $A[k-1, x+1]$ and we get the recurrence:

$$\begin{aligned} A[k,x] \leftarrow & Pr(I_k = x | I_{k-1} = x-1) A[k-1, x-1] \\ &+ Pr(I_k = x | I_{k-1} = x+1) A[k-1, x+1] \end{aligned}$$

From A, $E(I_n)$ can be computed as follows:

$$E(I_n) = \sum_{i \in \{1 \ldots n\}} x Pr(I_n = x) = \sum_{i \in \{1 \ldots n\}} i A[n,i]$$

2.8 Adaptive Marketing Strategy

In this section we propose a dynamic program for computing best adaptive spreading strategy and thus, prove Theorem 2.1.3. Here we give dynamic program when there are two types of areas. This can be extended to any constant number of types. Let $B(n_1, n_2, k)$ be the expected number of areas that adopt the product for a best ordering where n_1 is number of areas of type 1 and n_2 is the number of areas of type 2 in the market k is sum of decisions of vertices that have been scheduled so far. We note that deployment number k is equal to difference of number of yes decisions and no decisions. Let thresholds and initial acceptance probabilities for vertices of type i be c_i and p_i. At any given time in the strategy, let B_i be the best possible result if an area of type i is scheduled next. Depending on value of k, we have the following cases (cases 2 and 4 will not occur if $c_1 = c_2$):

1. $n_1 = 0 \vee n_2 = 0$: If all areas are of the same type, then all spreading strategies are equivalent and we can choose any arbitraty spreading strategy for the remaining areas.

2. $c_1 \leq k < c_2$: In this case, areas of type 1 will accept the idea **w.p.** 1. Areas of type 2 will accept the idea with probability p_2 and reject it with probability $1 - p_2$.

$$\begin{aligned} B_1 =& 1 + B(n_1 - 1, n_2, k + 1) \\ B_2 =& p_2 + p_2 B(n_1, n_2 - 1, k + 1) + (1 - p_2) B(n_1, n_2 - 1, k - 1) \\ B(n_1, n_2, k) =& \max\{B_1, B_2\} \end{aligned}$$

3. $-c_1 < k < c_1$: In this case, both types of areas will decide to accept or reject the idea on basis of initial acceptance probabilities. Therefore:

$$\begin{aligned} B_1 =& p_1 + p_1 B(n_1 - 1, n_2, k + 1) + (1 - p_1) B(n_1 - 1, n_2, k - 1) \\ B_2 =& p_2 + p_2 B(n_1, n_2 - 1, k + 1) + (1 - p_2) B(n_1, n_2 - 1, k - 1) \\ B(n_1, n_2, k) =& \max\{B_1, B_2\} \end{aligned}$$

4. $-c_2 < k \leq -c_1$: In this case, areas of type 1 will reject the idea with probability 1 and areas of type 2 will accept the idea with probability p_2.

$$
\begin{aligned}
B_1 =& B(n_1 - 1, n_2, k + 1) \\
B_2 =& p_2 + p_2 B(n_1, n_2 - 1, k + 1) + (1 - p_2) B(n_1, n_2 - 1, k - 1) \\
B(n_1, n_2, k) =& \max\{B_1, B_2\}
\end{aligned}
$$

5. $k \leq -c_2$: In this case, both types of areas will reject the idea. Therefore:

$$B(n_1, n_2, k) = 0$$

6. $k \geq cc_2$: In this case, both types of areas will reject the idea. Therefore:

$$B(n_1, n_2, k) = n_1 + n_2$$

This can easily be extended to any constant number of types. The time complexity with t types is $O(n^{t+1})$.

Chapter 3

Influence Maximization with Partial Incentives

3.1 Introduction

The ideas we are exposed to and the choices we make are significantly affected by our social context. It has long been studied how social networks – i.e. who we interact with in a variety of contexts – impact the choices we make, and how ideas and behaviors can spread through such networks [17, 56, 118, 130]. With the advent of the Internet, and websites such as Facebook and Google Plus devoted to the forming and maintaining of social networks, this effect becomes ever more evident. Individuals are linked together in ways that are more readily apparent and widespread than ever before, and accordingly understanding how social networks affect the behaviors and actions that spread through a society becomes ever more important.

A key question in this area is understanding how such a behavioral cascade can start. For example, for a company that wishes to introduce a new product but has

[0]This is a joint work with E. D. Demaine, M. T. Hajiaghayi, H. Mahini, D. L. Malec, S. Raghavan and M. Zadimoghadam. A version of this work appeared in *WWW '14* [39].

a limited promotional budget, it becomes critical to understand how to target their promotional efforts in order to generate awareness among as many people as possible. A well-studied model of this is the Influence Maximization problem, as introduced by Kempe et al. [75]. In the Influence Maximization problem, an optimizer wishes to find a small set of individuals to influence, such that this influence will cascade and grow through the social network to the maximum extent possible. For example, if a company wants to introduce a new piece of software, and believes that friends of users are likely to become users themselves, how should they allocate free copies of their software in order to maximize the size of their eventual user base?

Since the introduction of the Influence Maximization problem by Kempe et al. [75], there has been a great deal of interest and follow up work in the model. A particular driving force for applying this model has been the growth of large-scale social networks on the Internet. While Kempe et al. [75] give a greedy algorithm for approximating the Influence Maximization problem, it requires costly simulation at every step; thus, while their solution provides a good benchmark, a key area of research has been on finding practical, fast algorithms that themselves provide good approximations to the greedy algorithm [18, 23, 24, 25, 82]. The practical, applied nature of the motivating settings means that even small gains in performance (either runtime or approximation factor) are critical, especially on large, real-world instances.

We believe that the standard formulation of the Influence Maximization problem, however, misses a critical aspect of practical applications. In particular, it forces a binary choice upon the optimizer: when starting a cascade, an optimizer may choose for each individual whether to apply no influence or maximal influence, but has no choice in between. While this is reasonable for some settings – e.g. exposure to an idea or pathogen – it is far less reasonable for others of practical importance. For example, a company promoting a new product may find that giving away ten free copies is far less effective than offering a discount of ten percent to a hundred people. We propose a *fractional* version of the problem where the optimizer has the freedom to split influence across individuals as they see fit.

To make this concrete, consider the following problem an optimizer might face. Say that an optimizer feels there is some small, well-connected group whose adoption of their product is critical to success, but only has enough promotion budget remaining to influence one third of the group directly. In the original version of Influence Maximization, the optimizer is forced to decide which third of the group to focus on. We believe it is more natural to assume they have the flexibility to try applying uniform influence to the group, say offering everyone a discount of one third on the price of their product, or in fact any combination of these two approaches. While our results are preliminary, we feel our proposed model addresses some very practical and very real concerns with practical applications of Influence Maximization, and offers many opportunities for important future research.

3.2 Results

In this work, our main goal is to understand how our proposed fractional version of the Influence Maximization problem differs from the integral version proposed by Kempe et al. [75]. We consider this question from both a theoretical and an empirical perspective. On the theoretical side, we shall see that unlike many problems, the fractional version of Influence Maximization appears to retain essentially the same hardness as the fractional version. Furthermore, we give examples where the objective values for the fractional and integral versions can differ significantly. Nevertheless, we are able to extend the main positive result for Influence Maximization to the fractional case, namely submodularity results of Mossel and Roch [95] which give a very general extension of the results of Kempe et al. [75]. On the empirical side, we simulate the main algorithms and heuristics on real-world social network data, and find that the computed solutions are substantially more efficient in the fractional setting.

Our main theoretical result shows that the positive results of Mossel and Roch [95] extend to our proposed fractional model. Their result showed that in the integral case, when influence between agents is submodular, so too is the objective function in Influence Maximization. We show that for a continuous anolog of submodularity[1] the same results holds for our fractional case. We first consider a discretized version of the fractional Influence Maximization Problem, where each vertex can be assigned a weight that is a multiple of some discretization parameter $\epsilon = \frac{1}{N}$. Then, we consider the final influenced set by choosing a weighted seed set S, where the weight of each element is a multiple of ε. We show the fractional Influence Maximization objection is a submodular function of S for any $N \geq 1$ (Theorem 4). We further extend this result to the continuous case (Theorem 5). We note that this result does not follow simply by relating the fractional objective function to the integral one and interpolating or other similar methods; instead, we need to use a non-trivial reduction to the generalization of the influence maximization problem given by Mossel and Roch [95]. Not only does this result show that our problem admits a greedy solution with good approximation guarantee, it furthermore gives us hope that we can readily adapt the large body of work on efficient heuristics for the integral case to our problem and achieve good results.

In addition to showing the submodularity of the objective persists from the integral case to the fractional case, we show that the hardness of the integral case persists as well. In the case of fixed thresholds, we show that all of the hardness results of [75] extend readily to the fractional case. In particular, we show that for the fractional version of linear influence model, even an $n^{1-\epsilon}$ approximation algorithm is NP-hard to achieve. We first prove NP-hardness of the problem by a reduction from the independent set problem (Theorem 8) and then strengthen the result to prove inapproximability (Corollary 9). In addition, when thresholds are assumed to be independent and uniformly distributed in $[0, 1]$, we show that it is NP-Hard to achieve any better than

[1] We note this is neither of the two most common continuous extensions of submodularity, namely the multinear and Lovász extensions.

$1 - 1/e$ approximation in the Triggering model introduced by Kempe et al. [75]. This holds even for the simple case where triggering sets are deterministic and have constant sizes, and shows that even for this simple case the greedy approximation is tight, just as in the integral case. An important aspect of all of these reductions is that they use very simple DAGs, with only two layers of vertices.

Our last set of results focus on the special case where the network is a DAG. Here, we focus on the linear influence model with uniform thresholds. In this case, we see that we can easily compute the expected influence from any single node via dynamic programming; this closely resembles a previous result for the integral case [25]. In the fractional case, this gives us a sort of linearity result. Namely, if we are careful to avoid interference between the influences we place on nodes, we can conclude that the objective is essentially linear in the seed set. While the conditions on this theorem seem strong at first glance, it has a very powerful implication: all of the hardness results we presented involved choosing optimal seed sets from among the sources in a DAG, and this theorem says that with uniform thresholds the greedy algorithm finds the *optimal* such seed set.

3.3 Related Work

Several works in economics, sociology and political science have studied and modeled behaviors arising from information and influence cascades in social networks. Some of the earliest models were proposed by Granovetter [56] and Schelling [118]. Since then many such models have been studied and proposed in literature [17, 114, 130].

The advent of social networking platforms such as Facebook, Twitter and Flickr has provided researchers with unprecendented data about social interactions, albeit in a virtual setting. The question of monetization of this data is critically important for the entities that provide these platforms and the entities that want to leverage this data to engineer effective marketing campaigns. The above two factors have generated huge interest in algorithmic aspects of these systems.

A question of central importance is to recognize "important individuals" in a social netowrk. Domingos and Richardson [45, 112] were the first to propose heuristics for selection of customers on a network for marketing. Both these works focus on evaluating customers based on their intrinsic and network value. The network value is assumed to be generated by a customer influencing other customers in her social network to buy the product. In a seminal paper, Kempe et al. [75] give an approximation algorithm for selection of influential nodes under the linear threshold (LT) model. Mossel and Roch [95] generalized the results of Kempe et al. [75] to cases where the activation functions are monotone and submodular. Gunnec and Raghavan [58] were the first to discuss fractional incentives (they refer to these as partial incentives/inducements) in the context of a product design problem. They consider a fractional version of the target set selection problem (i.e., fixed thresholds, fractional incentives, a linear influence model, with the goal of minimizing the fractional incentives paid out so that all nodes in the graph are influenced). They provide an integer programming model, and show that when the neighbors of a node have equal influence on it, the problem is polynomially solvable via a greedy algorithm [57, 58, 59].

Some recent works have directly tackled the question of revenue maximization in social networks by leveraging differential pricing to monetize positive externalities arising due to adoption of product by neighbors of a customer [4, 8, 46, 62]. Other works have focused on finding faster algorithms for the target set selection problem ([23, 24, 25, 82]). A very recent theoretical result in this direction is an $O(\frac{(m+n)\log(n)}{\epsilon^3})$ algorithm giving an approximation guarantee of $1 - \frac{1}{e} - \epsilon$ [18]. While Leskovec et al. [82] do not compare their algorithm directly with the greedy algorithm of Kempe et al. [75], the heuristics in other papers ([23, 24, 25]) approach the performance of the greedy algorithm quite closely. For example, in one of the papers [24], the proposed heuristic achieves an influence spread of approximately 95% of the influence spread achieved by the greedy algorithm. An intersting fact on the flip side is that none of the heuristics beat the greedy algorithm (which itself is a heuristic) for even a single dataset.

3.4 Model

Integral Influence Model We begin by describing the model used for propagation of influence in social networks used in [95]; it captures the model described in [75] as a special case. Unlike the latter, the edges in this model are given only implicitly. Here, the graph is given by a vertex set V and an explicit description of how nodes influence each other. Specifically, for each vertex $v \in V$, we are given a function $f_v : 2^V \to [0, 1]$ specifying the amount of influence each subset $S \subseteq V$ exerts on v. We denote the set of all influence functions by $\mathcal{F} = \{f_v\}_{v \in V}$.

Given a graph specified by $(V, \mathcal{F})$, we want to understand how influence propagates in this graph. The spread of influence is (again) modeled by a process that runs in stages. In addition to the influence function f_v, each vertex v has a threshold $\theta_v \in [0, 1]$ representing how resistant it is to being influenced. If the currently activated set of vertices is $S \subseteq V$ at a given stage, then each $v \in V \setminus S$ becomes activated in the next stage if and only if $f_v(S) \geq \theta_v$. Our goal is to compare the total influence different sets have on the graph as a whole; we measure this by comparing the results of allowing the spreading process to run to completion given any pair of sets to use at the initial activated sets. To represent the fact that some (sets of) vertices may be more important than others, we define a weight function $w : 2^V \to \mathbb{R}_+$ on subsets of V. We then can define the value of a set S formally as follows. Given an initial activated set S_0, let $S_1^\Theta, S_2^\Theta, \ldots, S_n^\Theta$ be the activated sets after $1, 2, \ldots, n = |V|$ stages of our spreading process, when $\Theta = (\theta_v)_{v \in V}$ is our vector of thresholds. Then we want to understand the value of $w(S_n^\Theta)$ when we set $S_0 = S$. Note this may depend strongly on Θ; to that end, we assume that each threshold is independently distributed as $\theta_v \sim \mathcal{U}[0, 1]$. Then, we are interested in understanding the structure of the function $\sigma : 2^V \to \mathbb{R}_+$ given by

$$\sigma(S) = \mathop{\mathrm{E}}_{\Theta}[\ w(S_n^{\Theta}) \mid S_0 = S\ ].$$

Fractional Influence Model One shortcoming of the model described above is that it completely separates the effects of directly applied influence from other agents in the network. In particular, note that every agent in the social network is either explicitly activated by the optimizer (and influence from other agents is irrelevant), or is activated by influence from other agents with no (direct) involvement from the optimizer. It seems natural to suppose, however, that it is possible for agents to become activated by a mixture of direct influence from the designer and influence from other agents. For example, if we have chosen to activate a set S of agents, and know that this set strongly influences some other agent, we should be able to apply some intermediate level of influence with the assurance that this additional agent will become active eventually. To this end, we propose the following modification of the model. Rather than selecting a set S of nodes to activate, the optimizer specifies a vector $\mathbf{x} \in [0, 1]^n$ indexed by V, where x_v indicates the amount of direct influence we apply to v. We assume that this direct influence is additive with influence from other nodes in the network, and so a set S causes v to be activated if and only if $f_v(S) + x_v \geq \theta_v$. Here, we assume that no nodes are initially activated, that is $S_0 = \emptyset$. Note, however, that even without contributions from other nodes, our directly-applied influence can cause activations. For example, it is easy to see that

$$S_1^{\Theta} = \{v \in V : x_v \geq \theta_v\}.$$

An important point, however, is that our process is not simply a matter of selecting an initial activated set at random with marginal probabilities $\mathbf{x}$, since for any node v not initially activated, our direct influence x_v makes it easier for other nodes in the network to activate it in later rounds. Lastly, we observe that this model captures the originally discussed model as a special case, since selecting sets to initially activate corresponds exactly with choosing $\mathbf{x} \in \{0,1\}^n$, just with a single-round delay in the process. To this end, hereafter we refer to the original model as the integral influence model, and this new model as the fractional influence model. As before, we want to understand the structure of the expected value of the final influenced set as a function of how we apply influence to nodes in a graph. We extend our function to $\sigma : [0,1]^n \to \mathbb{R}_+$ by

$$\sigma(\mathbf{x}) = \mathop{\mathrm{E}}_{\Theta}[\ w(S_n^{\Theta}) \mid \text{we apply direct influences } \mathbf{x}\ ].$$

Linear Influence Model This is a special case of the fractional influence model. In the linear variant of the problem, our influence functions are computed as follows. We are given a digraph $G = (V, E)$ and a weight function $\mathbf{w}$ on edges. We use $\delta^-(v)$ and $\delta^+(v)$ to denote the sets of nodes with edges to and edges from v, respectively. Then, we denote the influence function f_v for v by

$$f_v(S) = \sum_{u \in S \cap \delta^-(v)} w_{uv}.$$

In our model, we assume that $\sum_{u \in \delta^-(v)} w_{uv} \leq 1$ for every vertex $v \in V$.

non-Stochastic Thresholds In all these models discussed above, we make the stochastic assumption that the thresholds of nodes are independently distributed as $\theta_v \sim \mathcal{U}[0,1]$. One natural question is whether the same results hold when we are facing arbitrary thresholds. We answer this question negatively by providing hardness results in Section 3.8 when the thresholds are given as part of the instance.

3.5 Reduction

In order to state the main result of [95], we need to define the following properties for set functions. Given a set N and a function $f : 2^N \to \mathbb{R}$, we say that:

- f is *normalized* if $f(\emptyset) = 0$;
- f is *monotone* if $f(S) \leq f(T)$ for any $S \subseteq T \subseteq N$; and
- f is *submodular* if $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$ for any $S \subseteq T \subseteq N$ and $x \in N \setminus T$.

We say that a collection of functions satisfies the above properties if every function in the collection does. With the above definitions in hand, we are now ready to state the following result of [95].

Theorem 3. *Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of our problem. If both w and $\mathcal{F}$ are normalized, monotone, and submodular, then σ is as well.*

We want to extend Theorem 3 to the fractional influence model. We show that, for arbitrarily fine discretizations of $[0, 1]$, any instance of our problem considered on the discretized space can be reduced to an instance of the original problem. Fix $N \in \mathbb{Z}_+$, and let $\delta = 1/N > 0$ be our discretization parameter. Let $\Delta = \{0, \delta, 2\delta, \ldots, 1\}$. We consider the function σ restricted to the domain Δ^n. Lastly, let δ_v be the vector with δ in the component corresponding to v, and 0 in all other components. Then we can state our desired properties for σ as follows:

- we say σ is *normalized* if $\sigma(\mathbf{0}) = 0$;
- we say σ is *monotone* if $\mathbf{x} \leq \mathbf{y}$ implies $\sigma(\mathbf{x}) \leq \sigma(\mathbf{y})$; and
- we say σ is *submodular* if for any $\mathbf{x} \leq \mathbf{y}$, and any $v \in V$, either $y_v = 1$ or $\sigma(\mathbf{x} + \delta_v) - \sigma(\mathbf{x}) \geq \sigma(\mathbf{y} + \delta_v) - \sigma(\mathbf{y})$,

where all comparisons and additions between vectors above are componentwise. We get the following extension of Theorem 3.

Theorem 4. *Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of our problem. If both w and $\mathcal{F}$ are normalized, monotone, and submodular, then for any discretization Δ^n of $[0,1]^n$ (as defined above), σ is normalized, monotone, and submodular on Δ^n.*

Proof. We prove this by reducing the (discretized) fractional problem for $\mathcal{I}$ to an instance of the integral influence problem and then applying Theorem 3. We begin by modifying $\mathcal{I}$ to produce a new instance $\hat{\mathcal{I}} = (\hat{V}, \hat{\mathcal{F}}, \hat{w})$. Then, we show that $\hat{\mathcal{F}}$ and $\hat{w}$ will retain the properties of normalization, monotonicity, and submodularity. Lastly, we showing a mapping from (discretized) fractional activations for $\mathcal{I}$ to integral activations for $\hat{\mathcal{I}}$ such that objective values are preserved, and our desired fractional set function properties for σ correspond exactly to their integral counterparts for the objective function $\hat{\sigma}$ for $\hat{\mathcal{I}}$. The result then follows immediately from Theorem 3.

We begin by constructing the instance $\hat{\mathcal{I}}$. The key idea is that we can simulate fractional activation with integral activation by adding a set of dummy activator nodes for each original node; each activator node applies an incremental amount of pressure on its associated original node. Then, for each original node we just need to add the influence from activator node to that from other original nodes, and truncate the sum to one. Fortunately, both of the aforementioned operations preserve the desired properties. Lastly, in order to avoid the activator nodes interfering with objective values, we simply need to give them weight zero. With this intuition in mind, we now define $\hat{\mathcal{I}} = (\hat{V}, \hat{\mathcal{F}}, \hat{w})$ formally.

First, we construct $\hat{V}$. For each node $v \in V$, create a set $A_v = \{v^1, v^2, \ldots, v^{1/\delta}\}$ of activator nodes for v. Our node set in the new instance is

$$\hat{V} = V \cup \left(\bigcup_{v \in V} A_v \right).$$

We now proceed to define the functions $\hat{f}_{\hat{v}}$ for each $\hat{v} \in \hat{V}$. If $\hat{v}$ is an activator node for some $v \in V$, we simply set $\hat{f}_{\hat{v}} \equiv 0$; otherwise, $\hat{v} \in V$ and we set

$$\hat{f}_{\hat{v}}(S) = \min\left(f_{\hat{v}}(S \cap V) + \delta|S \cap A_{\hat{v}}|, 1\right)$$

for each $S \subseteq \hat{V}$. Lastly, we set

$$\hat{w}(S) = w(S \cap V)$$

for all $S \subseteq \hat{V}$. Together, these make up our modified instance $\hat{\mathcal{I}}$.

We now show that since w and $\mathcal{F}$ are normalized, monotone, and submodular, $\hat{w}$ and $\hat{\mathcal{F}}$ will be as well. We begin with $\hat{w}$, since it is the simpler of the two. Now, $\hat{w}$ is clearly normalized since $\hat{w}(\emptyset) = w(\emptyset)$. Similarly, for any $S \subseteq T \subseteq \hat{V}$, we have that $S \cap V \subseteq T \cap V$, and so

$$\hat{w}(S) = w(S \cap V) \leq w(T \cap V) = \hat{w}(T),$$

by the submodularity of w. Lastly, let $\hat{u} \in \hat{V} \setminus T$. If $\hat{u} \in V$, then

$$\begin{aligned} \hat{w}(S \cup \{\hat{u}\}) - \hat{w}(S) &= w((S \cap V) \cup \{\hat{u}\}) - \hat{w}(S \cap V) \\ &\geq w((T \cap V) \cup \{\hat{u}\}) - \hat{w}(T \cap V) \\ &= \hat{w}(T \cup \{\hat{u}\}) - \hat{w}(T), \end{aligned}$$

since w is submodular. On the other hand, if $\hat{u} \notin V$, we immediately get that

$$\hat{w}(S \cup \{\hat{u}\}) - \hat{w}(S) = 0 = \hat{w}(T \cup \{\hat{u}\}) - \hat{w}(T).$$

Thus, we can see that $\hat{w}$ is normalized, monotone, and submodular.

Next, we show that $\hat{\mathcal{F}}$ is normalized, monotone, and submodular. For $\hat{v} \in \hat{V} \setminus V$, it follows trivially since $\hat{\mathcal{F}}$ is identically 0. In the case that $\hat{V} \in V$, it is less immediate, and we consider each of the properties below.

- $\hat{f}_{\hat{v}}$ normalized. This follows by computing that

$$\begin{aligned}\hat{f}_{\hat{v}}(\emptyset) &= \min\left(f_{\hat{v}}(V \cap \emptyset) + \delta|A_{\hat{v}} \cap \emptyset|, 1\right) \\ &= \min\left(f_{\hat{v}}(\emptyset) + \delta|\emptyset|, 1\right) = 0,\end{aligned}$$

  since $f_{\hat{v}}$ is normalized.

- $\hat{f}_{\hat{v}}$ monotone. Let $S \subseteq T \subseteq \hat{V}$. Then we have both $S \cap V \subseteq T \cap V$ and $S \cap A_{\hat{v}} \subseteq T \cap A_{\hat{v}}$. Thus, we can see that

$$\begin{aligned}f_{\hat{v}}(V \cap S) &\leq f_{\hat{v}}(V \cap T) \\ |A_{\hat{v}} \cap S| &\leq |A_{\hat{v}} \cap T|,\end{aligned}$$

  where the former follows by the monotonicity of $f_{\hat{v}}$. Combining these, we get that

$$f_{\hat{v}}(V \cap S) + \delta|A_{\hat{v}} \cap S| \leq f_{\hat{v}}(V \cap T) + \delta|A_{\hat{v}} \cap T|.$$

  Thus, we may conclude that $\hat{f}_{\hat{v}}(S) \leq \hat{f}_{\hat{v}}(T)$, since it follows from the above inequality that

$$f_{\hat{v}}(V \cap S) + \delta|A_{\hat{v}} \cap S| > \min\left(f_{\hat{v}}(V \cap T) + \delta|A_{\hat{v}} \cap T|, 1\right)$$

  implies

$$\begin{aligned}&\min \quad \left(f_{\hat{v}}(V \cap S) + \delta|A_{\hat{v}} \cap S|, 1\right) = 1 \\ = &\min \quad \left(f_{\hat{v}}(V \cap T) + \delta|A_{\hat{v}} \cap T|, 1\right).\end{aligned}$$

- $\hat{f}_{\hat{v}}$ submodular. Let $S \subseteq T \subseteq \hat{V}$, and $\hat{u} \in \hat{V} \setminus T$. Now, we have three cases, depending on the choice of $\hat{u}$. If $\hat{u} \in V$, we have that $\hat{u} \notin A_{\hat{v}}$, and so

$$\begin{aligned}(f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) - (f_{\hat{v}}(S) - \delta|S \cap A_{\hat{v}}|) \\ = f_{\hat{v}}(S \cup \{\hat{u}\}) - f_{\hat{v}}(S);\end{aligned}$$

and

$$\begin{aligned}(f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|) - (f_{\hat{v}}(T) - \delta|T \cap A_{\hat{v}}|) \\ = f_{\hat{v}}(T \cup \{\hat{u}\}) - f_{\hat{v}}(T).\end{aligned}$$

Thus, the submodularity of $f_{\hat{v}}$ implies the former is greater than or equal to the latter. On the other hand, if $\hat{u} \in A_{\hat{v}}$, then $\hat{u} \notin V$, and we can see that

$$\begin{aligned}(f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) = \delta = \\ = (f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|) .\end{aligned}$$

Lastly, if $\hat{u} \notin V \cup A_{\hat{v}}$, we can immediately see that

$$\begin{aligned}(f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) = 0 = \\ = (f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|) .\end{aligned}$$

Thus, in every case, we may conclude that

$$\begin{aligned}(f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) \geq \\ \geq (f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|) ,\end{aligned}$$

and hence (by the same reasoning as for monotonicity) we may conclude that

$$\hat{f}_{\hat{v}}(S \cup \{\hat{u}\}) - \hat{f}_{\hat{v}}(S) \geq \hat{f}_{\hat{v}}(T \cup \{\hat{u}\}) - \hat{f}_{\hat{v}}(T).$$

Thus, we can see that $\hat{\mathcal{F}}$ is normalized, monotone, and submodular on $\hat{V}$, exactly as desired.

As such, we can apply Theorem 3 to our function and get that for our modified instance $\hat{\mathcal{I}} = (\hat{V}, \hat{\mathcal{F}}, \hat{w})$, the corresponding function $\hat{\sigma}$ must normalized, monotone, and submodular. All that remains is to demonstrate our claimed mapping from (discretized) fractional activations for $\mathcal{I}$ to integral activations for $\hat{\mathcal{I}}$.

We do so as follows. For each $v \in V$ and each $d \in \Delta$, let $A_v^d = \{v^1, v^2, \ldots, v^d\}$. Then, given the vector $\mathbf{x} \in \Delta^n$, we set

$$S^{\mathbf{x}} = \bigcup_{v \in V} A_v^{x_v},$$

where x_v is the component of $\mathbf{x}$ corresponding to the node v.

We first show that under this definition we have that $\sigma(\mathbf{x}) = \hat{\sigma}(S^{\mathbf{x}})$. In fact, as we will see the sets influenced will be the same not just in expectation, but for every set of thresholds Θ for the vertices V. Note that in the modified setting $\hat{\mathcal{I}}$ we also have thresholds for each vertex in $\hat{V} \setminus V$; however, since we chose $\hat{f}_{\hat{v}} \equiv 0$ for all $\hat{v} \in \hat{V} \setminus V$, and thresholds are independent draws from $\mathcal{U}[0, 1]$, we can see that with probability 1 we have $\hat{f}_{\hat{v}}(S) < \theta_{\hat{v}}$ for all S and all $\hat{v} \in \hat{V} \setminus V$. Thus, in the following discussion we do not bother to fix these thresholds, as their precise values have no effect on the spread of influence.

Fix some vector Θ of thresholds for the vertices in V. Let $S_1^{\Theta}, \ldots, S_n^{\Theta}$ and $\hat{S}_1^{\Theta}, \ldots, \hat{S}_n^{\Theta}$ be the influenced sets in each round in the setting $\mathcal{I}$ with influence vector $\mathbf{x}$ and in the setting $\hat{\mathcal{I}}$ with influence set $S^{\mathbf{x}}$, respectively. We show by induction that for all $i = 0, 1, \ldots, n$, $\hat{S}_i^{\Theta} \cap V = S_i^{\Theta}$. By the definition of $\hat{w}$, this immediately implies that $w(S_n^\Theta) = \hat{w}(\hat{S}_n^\Theta)$, and the claim follows. We prove our claim by induction. For $i = 0$, the equality follows simply by our definitions of the processes, since $S_0 = \emptyset$ and $\hat{S}_0 = S^{\mathbf{x}}$. Now, assuming the claim holds for $i-1$, we need to show that it holds for i. By our definition of the processes, we know that

$$S_i^\Theta = S_{i-1}^\Theta \cup \{v \in V \setminus S_{i-1}^\Theta : f_v(S_{i-1}^\Theta) + x_v \geq \theta_v\};$$

similarly, we have that

$$\hat{S}_i^\Theta = \hat{S}_{i-1}^\Theta \cup \{\hat{v} \in \hat{V} \setminus \hat{S}_{i-1}^\Theta : \hat{f}_{\hat{v}}(\hat{S}_{i-1}^\Theta) \geq \theta_{\hat{v}}\}.$$

Recall, however, that for all $\hat{v} \in \hat{V} \setminus V$, we have that $\hat{f}_{\hat{v}} \equiv 0$, and it follows that $\hat{S}_i^\Theta \setminus V = S^{\mathbf{x}}$ for all i. Thus, we can rewrite the second equality above as

$$\hat{S}_i^\Theta = \hat{S}_{i-1}^\Theta \cup \{v \in V \setminus \hat{S}_{i-1}^\Theta : \hat{f}_v(\hat{S}_{i-1}^\Theta) \geq \theta_v\}.$$

Consider an arbitrary $v \in V \setminus S_{i-1}^\Theta = V \setminus \hat{S}_{i-1}^\Theta$. Now, we know that $v \in \hat{S}_i^\Theta$ if and only if

$$\theta_v \leq \hat{f}_v(\hat{S}_{i-1}^\Theta) = \min(f_v(\hat{S}_{i-1}^\Theta \cap V) + \delta|\hat{S}_{i-1}^\Theta \cap A_v|, 1)$$

Recall, however, that $\hat{S}_{i-1}^\Theta \cap V = S_{i-1}^\Theta$ by assumption. Furthermore, we can compute that

$$|\hat{S}_{i-1}^\Theta \cap A_v| = |S^{\mathbf{x}} \cap A_v| = |A_v^{x_v}| = |\{v^1, \ldots, v^{x_v}\}| = x_v/\delta.$$

Thus, since we know that $\theta_v \leq 1$ always, we can conclude that $v \in \hat{S}_i^\Theta \cap V$ if and only if

$$\theta_v \leq f_v(S_{i-1}^\Theta) + x_v,$$

which is precisely the condition for including v in S_i^Θ. Thus, we can conclude that

$\hat{S}_i^{\Theta} \cap V = S_i^{\Theta}$.

We have now shown that for all vectors of thresholds Θ for vertices in V, with probability 1 we have that $\hat{S}_i^{\Theta} \cap V = S_i^{\Theta}$ for $i = 0, 1, \ldots, n$. In particular, note that $\hat{S}_n^{\Theta} \cap V = S_n^{\Theta}$, and so $\hat{w}(\hat{S}_n^{\Theta}) = w(S_n^{\Theta})$. Thus, we may conclude that $\hat{\sigma}(S^{\mathbf{x}}) = \sigma(\mathbf{x})$.

Lastly, we need to show that for our given mapping from (discretized) fractional activation vectors $\mathbf{x}$ to set $S^{\mathbf{x}}$, we have that the desired properties for σ are satisfied if the corresponding properties are satisfied for $\hat{\sigma}$. So we assume that $\hat{\sigma}$ is normalized, monotone, and submodular (as, in fact, it must be by the above argument and Theorem 3), and show that σ is as well. First, note that $\mathbf{x} = \mathbf{0}$ implies $S^{\mathbf{x}} = \emptyset$, and so $\sigma(\mathbf{x}) = \hat{\sigma}(\emptyset) = 0$. Second, let $\mathbf{x}, \mathbf{y} \in \Delta^n$ such that $\mathbf{x} \leq \mathbf{y}$ componentwise. Then we can see that $S^{\mathbf{x}} \subseteq S^{\mathbf{y}}$ and so

$$\sigma(\mathbf{x}) = \hat{\sigma}(S^{\mathbf{x}}) \leq \hat{\sigma}(S^{\mathbf{y}}) = \sigma(\mathbf{y}).$$

Finally, pick some $v \in V$ such that $y_v < 1$. Recall our definition of $\hat{f}_{\hat{v}}$; by inspection, we can see that we have $\hat{f}_{\hat{v}}(S) = \hat{f}_{\hat{v}}(T)$ any time both $S \cap V = T \cap V$ and $|S \cap A_v| = |T \cap A_v|$, for any $S, T \in \hat{V}$. Thus, we can see that $S^{\mathbf{x}+\delta_i} = S^{\mathbf{x}} \cup \{v^{x_v+1}\}$ and $S^{\mathbf{y}} = S^{\mathbf{y}} \cup \{v^{y_v+1}\}$. So we have

$$\begin{aligned}
\sigma(\mathbf{x} + \delta_i) - \sigma(\mathbf{x}) &= \hat{\sigma}(S^{\mathbf{x}} \cup \{v^{x_v+1}\}) - \hat{\sigma}(S^{\mathbf{x}}) \\
&= \hat{\sigma}(S^{\mathbf{x}} \cup \{v^{y_v+1}\}) - \hat{\sigma}(S^{\mathbf{x}}) \\
&\geq \hat{\sigma}(S^{\mathbf{y}} \cup \{v^{y_v+1}\}) - \hat{\sigma}(S^{\mathbf{y}}) \\
&= \sigma(\mathbf{y} + \delta_i) - \sigma(\mathbf{y}).
\end{aligned}$$

Thus, σ has exactly the claimed properties on Δ^n, and the theorem follows. □

In fact, we can use the same technique as achieve the following extension to fully continuous versions of our properties. We define the following properties for σ on the continuous domain $[0,1]^n$:

- we say σ is *normalized* if $\sigma(\mathbf{0}) = 0$;
- we say σ is *monotone* if $\mathbf{x} \le \mathbf{y}$ implies $\sigma(\mathbf{x}) \le \sigma(\mathbf{y})$; and
- we say σ is *submodular* if for any $\mathbf{x} \le \mathbf{y}$, any $v \in V$, and for any $\varepsilon > 0$ such that $y_v + \varepsilon \le 1$, we have that $\sigma(\mathbf{x} + \varepsilon_v) - \sigma(\mathbf{x}) \ge \sigma(\mathbf{y} + \varepsilon_v) - \sigma(\mathbf{y})$,

where ε_v is the vector with a value of ε in the coordinate corresponding to v and a value of 0 in all other coordinates. As before, all comparisons and additions between vectors above are componentwise. The same techniques immediately give us the following theorem.

Theorem 5. *Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of our problem. If both w and $\mathcal{F}$ are normalized, monotone, and submodular, then σ is normalized, monotone, and submodular on $[0,1]^n$.*

Proof. We use the exact same technique as in the proof of Theorem 4. The only difference is how we define the activator nodes in our modified instance $\hat{\mathcal{I}}$. Here, rather than trying to model the entire domain, we simply focus on the points we want to verify our properties on. To that end, fix some $x \le y$, as well as an $\varepsilon > 0$ and some $v \in V$. We will only define three activator nodes here: $a^{\mathbf{x}}$, $a^{\mathbf{x}-ys}$, and a^{ε}. The first two contributes amounts of $x_{v'}$ and $(y_{v'} - x_{v'}) \ge 0$, respectively, to the modified influence function for vertex $v' \in V$. The last contributes an amount of ε to the influence function for vertex v, and makes no contribution to any other influence functions. As before, all influence functions get capped at one. Our modified weight function is defined exactly as before, and it is easy to see that exactly the same argument will imply that the modified weight and influence functions will be normalized, monotone, and submodular, allowing us to apply Theorem 3.

Thus, all that remains is to relate the function values between the original and modified instances. Note, however, that here it is even simpler than in the discretized case. If $\hat{\sigma}$ is the objective for σ, it clear to see that:

$$
\begin{aligned}
\sigma(0) &= \hat{\sigma}(\emptyset) \\
\sigma(\mathbf{x}) &= \hat{\sigma}(\{a^{\mathbf{x}}\}) \\
\sigma(\mathbf{y}) &= \hat{\sigma}(\{a^{\mathbf{x}}, a^{\mathbf{y}-\mathbf{x}}\}) \\
\sigma(\mathbf{x} + \varepsilon_V) &= \hat{\sigma}(\{a^{\mathbf{x}}, a^{\varepsilon}\}) \\
\sigma(\mathbf{y} + \varepsilon_v) &= \hat{\sigma}(\{a^{\mathbf{x}}, a^{\mathbf{y}-\mathbf{x}}, a^{\varepsilon}\})
\end{aligned}
$$

The above equalities make it clear that the desired qualities for σ will follow immediately from their discrete counterparts for $\hat{\sigma}$. □

Theorem 6. *Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of our problem. Then for any discretization Δ^n of $[0,1]^n$ (as defined above), if σ is normalized, monotone, and submodular on Δ^n, we have that*

$$\max_{\substack{\mathbf{x} \in \Delta^n: \\ \|\mathbf{x}\|_1 \leq K}} \sigma(\mathbf{x}) \geq (1 - \delta n) \max_{\substack{\mathbf{x} \in [0,1]^n: \\ \|\mathbf{x}\|_1 \leq K}} \sigma(\mathbf{x}),$$

for any K.

Proof. Let $\mathbf{x}^*$ be an optimal solution to our problem on $[0,1]^n$, i.e. we have

$$\operatorname*{argmax}_{\mathbf{x} \in [0,1]^n : \|\mathbf{x}\|_1 \leq K} \sigma(\mathbf{x}).$$

Let $\bar{\mathbf{x}}^*$ be the result of rounding $\mathbf{x}^*$ up componentwise to the nearest element of Δ^n. Formally, we define $\bar{\mathbf{x}}^*$ by $\bar{\mathbf{x}}^*_v = \min\{d \in \Delta : d \geq \mathbf{x}^*\}$. Note that by monotonicity, we must have that $\sigma(\bar{\mathbf{x}}^*) \geq \sigma(\mathbf{x}^*)$; we also have that $\|\bar{\mathbf{x}}^*\|_1 \leq \|\mathbf{x}^*\|_1 + \delta n$. Now, consider constructing $\bar{\mathbf{x}}^*$ greedily by adding δ to a single coordinate in each step. Formally, set

$\mathbf{x}^0 = \mathbf{0}$, and for each $i = 1, 2, \ldots, \|\bar{\mathbf{x}}^*\|_1/\delta$ set

$$\mathbf{x}^i = \mathbf{x}^{i-1} + \delta_v \text{ for some } v \in \underset{v:\, \mathbf{x}_v^{i-1}<1}{\operatorname{argmax}} (\sigma(\mathbf{x}^{i-1} + \delta_v) = \sigma(\mathbf{x}^{i-1})),$$

where (as before) δ_v is a vector with δ in the component corresponding to v and 0 in all other components. Note that the submodularity of σ implies that $\sigma(\mathbf{x}^i) - \sigma(\mathbf{x}^{i-1})$ is decreasing in i. An immediate consequence of this is that, for any i, we have that

$$\sigma(\mathbf{x}^i) \geq \frac{i}{\|\bar{\mathbf{x}}^*\|_1} \sigma(\bar{\mathbf{x}}^*).$$

Invoking the above for $i = K/\delta$ we get that

$$\sigma(\mathbf{x}^{K/\delta}) \geq \frac{K/\delta}{\|\bar{\mathbf{x}}^*\|_1} \sigma(\bar{\mathbf{x}}^*) \geq \frac{K}{K + \delta n} \sigma(\bar{\mathbf{x}}^*) \geq (1 - \delta n)\sigma(\bar{\mathbf{x}}^*).$$

We observe that $\|\mathbf{x}^{K/\delta}\|_1 = K$, and $\mathbf{x}^{K/\delta} \in \Delta^n$, and so the desired theorem follows.

□

3.6 DAGs

In this section, we focus on the case of linear influences, and argue that some aspects of the problem become simpler on DAGs. Similar to the fractional influence model, our goal is to pick an influence vector $\mathbf{x} \in [0, 1]^{|V|}$ indexed by V to maximize

$$\sigma(\mathbf{x}) = \mathop{\mathrm{E}}_{\Theta}[\ |S_n^\Theta| \mid \text{we apply direct influences } \mathbf{x}\ ],$$

where $S_1^\Theta, \ldots, S_n^\Theta$ is the sequence of sets of nodes activated under thresholds Θ and direct influence $\mathbf{x}$. We sometimes abuse notation and use $\sigma(S)$ to denote obj applied to the characteristic vector of the set $S \in 2^V$.

Given a DAG $G = (V, E)$ and a fractional influence vector $\mathbf{x} \in [0, 1]^{|V|}$ indexed by V, we define the sets

$$I(\mathbf{x}) = \{v \in V : x_v > 0\}, \text{ and}$$
$$S(\mathbf{x}) = \{v \in V : x_v + \textstyle\sum_{u \in \delta^-(v)} w_{uv} > 1\},$$

as the sets of nodes *influenced* by $\mathbf{x}$ and *(over-)saturated* by $\mathbf{x}$. Note that $S(\mathbf{x}) \subseteq I(\mathbf{x})$. We get the following theorem.

Theorem 7. *Given a DAG G and influence vector $\mathbf{x}$, if G contains no path from an element of $I(\mathbf{x})$ to any element of $S(\mathbf{x})$, then we have that*

$$\sigma(\mathbf{x}) = \textstyle\sum_{v \in V} x_v \sigma(\mathbf{1}_v).$$

Proof. We prove this by induction on the number of vertices. In the case that V contains only a single vertex, the claim is trivial. Otherwise, let $G = (V, E)$ and $\mathbf{x}$ satisfy our assumptions, with $|V| = n > 1$, and assume out claim holds for any DAG with $(n-1)$ or fewer nodes. Let $s \in V$ be a source vertex (i.e. have in-degree 0) in G. Now, if $s \notin I(\mathbf{x})$, we know that s is never activated. Let $\hat{\sigma}$ and $\hat{\mathbf{x}}$ be σ on G restricted to $V \setminus s$ and $\mathbf{x}$ restricted to $V \setminus s$, respectively, and observe that we may apply our induction hypothesis to $\hat{\sigma}(\hat{\mathbf{x}})$ since removing s from G cannot cause any of the requirements for our theorem to become violated. Thus, we can see that

$$\sigma(\mathbf{x}) = \hat{\sigma}(\hat{\mathbf{x}}) = \sum_{v \in V \setminus s} x_v \hat{\sigma}(\mathbf{1}_v) = \sum_{v \in V} x_v \sigma(\mathbf{1}_v),$$

since $x_s = 0$.

Now, assume that $s \in I(\mathbf{x})$. Recall that by our conditions on G, therefore, we know that G contains no path from s to any elements of $S(\mathbf{x})$. One critical implication of this is that none of the nodes in $\delta^+(s)$ have paths to elements of $S(\mathbf{x})$ either, and so we made apply influence to them without violating the assumptions of our inductive hypothesis, as long as we are careful not to add so much weight that they become saturated.

In order to prove our claim, we focus on G restricted to $V \setminus \{s\}$, call it $\hat{G}$. Let $\hat{\sigma}$ be σ over $\hat{G}$, and consider the following two influence vectors for $\hat{G}$. Define $\hat{\mathbf{x}}$ to simply be the restriction of $\mathbf{x}$ to $\hat{G}$; define $\hat{\mathbf{y}}$ by $y_v = w_{sv}$ if $v \in \delta^+(s)$ and 0 otherwise. Letting $\hat{I}$ and $\hat{Sat}$ be I and S, respectively, restricted to $\hat{G}$, we can see that

$$\left.\begin{array}{l}\hat{I}(\hat{\mathbf{x}}), \hat{I}(\hat{\mathbf{y}}), \hat{I}(\hat{\mathbf{x}}+\hat{\mathbf{y}}) \subseteq I(\mathbf{x}) \cup \delta^+(s), \text{ and} \\ \hat{S}(\hat{\mathbf{x}}), \hat{S}(\hat{\mathbf{y}}), \hat{S}(\hat{\mathbf{x}}+\hat{\mathbf{y}}) \subseteq S(\mathbf{x}).\end{array}\right\} \tag{3.1}$$

The observation that gives the above is that, compared to $\mathbf{x}$, the only vertices with increased influence applied to them are the elements of $\delta^+(s)$, and the amounts of these increases are precisely balanced by the removal of s (and its outgoing edges) from $\hat{G}$. In particular, note that for any $v \in V \setminus \{s\}$, by our definition of $]\hat{ys}$ we have that

$$x_v + \sum_{u \in \delta^-(v)} w_{uv} = \hat{x_v} + \hat{y_v} + \sum_{\substack{u \in \delta^-(v) \\ u \neq s}} w_{uv}.$$

As previously noted G contains no paths from an element of $\delta^+(s)$ to any element of $S(\mathbf{x})$; this combined with (3.1) allows us to conclude that we may apply our induction hypothesis to $\hat{G}$ with any of $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, or $\hat{\mathbf{x}} + \hat{\mathbf{y}}$. We proceed by showing that for any vector Θ of thresholds for G (and its restriction to $\hat{G}$), we have that the set activated under $\mathbf{x}$ in G always corresponds closely to one of the sets activated by $\hat{\mathbf{x}}$ or $(\hat{\mathbf{x}} + \hat{\mathbf{y}})$ in $\hat{G}$. To that end, fix any vector Θ. We consider the cases where $x_s \geq \theta_s$ and $x_s < \theta_s$ separately.

We begin with the case where $x_s < \theta_s$, since it is the simpler of the two. Let $S_0^\Theta, \ldots, S_n^\Theta$ and $\hat{S}_0^\Theta, \ldots, \hat{S}_n^\Theta$ denote the sets activated in G under $\mathbf{x}$ and in $\hat{G}$ under $\hat{\mathbf{x}}$, respectively, in stages $0, \ldots, n$. Note that since s is a source, and $x_s < \theta_s$, we know that $s \notin S_i^\Theta$ for all i. However, this means that every node in $V \setminus \{s\}$ has both the same direct influence applied to it under $\mathbf{x}$ and $\hat{\mathbf{x}}$, and the same amount of influence applied by any activated set in both G and $\hat{G}$. So we can immediately see that since $S_0^\Theta = \emptyset = \hat{S}_0^\Theta$, by induction we will have that $S_i^\Theta = \hat{S}_i^\Theta$ for all i, and in particular for $i = n$.

The case where $x_s \geq \theta_s$ requires more care. Let $S_0^\Theta, \ldots, S_n^\Theta$ and $\hat{S}_0^\Theta, \ldots, \hat{S}_n^\Theta$ denote the sets activated in G under $\mathbf{x}$ and in $\hat{G}$ under $\hat{\mathbf{x}} + \hat{\mathbf{y}}$, respectively, in stages $0, \ldots, n$. Note that our assumption implies that s will be activated by our direct influence in the first round, and so we have $s \in \hat{S}_i^\Theta$ for all $i \geq 1$. Fix some $v \in V$, $v \neq s$, and let $f_v(S)$ and $\hat{f}_v(S)$ denote the total influence – both direct and cascading – applied to in G and $\hat{G}$, respectively, when the current active set is S. Then, we can see that for any $S \subseteq V \setminus \{s\}$ we have that

$$\hat{f}_v(S) = \hat{x_v} + \hat{y_v} + \sum_{\substack{u \in \delta^-(v) \\ u \in S}} w_{uv} = x_v + \sum_{\substack{u \in \delta^-(v) \\ u \in S \cup \{s\}}} w_{uv} = f_v(S \cup \{s\}). \tag{3.2}$$

Furthermore, note that both f_v and $\hat{f}_v$ are always monotone nondecreasing. While we cannot show that $S_i^\Theta = \hat{S}_i^\Theta$ for all i in this case, we will instead show that $S_i^\Theta \setminus \{s\} \subseteq \hat{S}_i^\Theta \subseteq S_{i+1}^\Theta \setminus \{s\}$ for all $i = 0, \ldots, n-1$. Recall that the propagation of influence converges by n steps. That is, if we continued the process for an additional step to produce activated sets S_{n+1}^Θ and $\hat{S}_{n+1}^\Theta$, we would have that $S_{n+1}^\Theta = S_n^\Theta$ and $\hat{S}_{n+1}^\Theta = \hat{S}_n^\Theta$. However, our claim would extend to this extra stage as well, and so we conclude that we must have that $S_n^\Theta = \hat{S}_n^\Theta \cup \{s\}$. We prove our claim inductively. First, observe that it holds trivially for $i = 0$, since we have $S_0^\Theta = \hat{S}_0^\Theta = \emptyset$, and previously observed that $s \in S_1^\Theta$. Now, the claim holds for some i. Note, however, that by (3.2) and monotonicity we must have that for all $v \in V$, $v \neq s$

$$\begin{aligned} f_v(S_i^\Theta) &= \hat{f}_v(S_i^\Theta \setminus \{s\}) \leq \hat{f}_v(\hat{S}_i^\Theta) \\ &\leq \hat{f}_v(S_{i+1}^\Theta \setminus \{s\}) = f_v(S_{i+1}^\Theta). \end{aligned}$$

But from the above, we can see that $S_{i+1} \setminus \{s\} \subseteq \hat{S}^{\Theta}_{i+1} \subseteq S^{\Theta}_{i+2} \setminus \{s\}$ since such a v in included in each of the above sets if and only if $f_v(S^{\Theta}_i)$, $\hat{f}_v(\hat{S}^{\Theta}_i)$, or $f_v(S^{\Theta}_{i+1})$, respectively, exceeds θ_v.

Thus, by observing that θ_s is an independent draw from $\mathcal{U}[0, 1]$, we can see that taking expectations over Θ and conditioning on which of θ_s and x_s is larger, gives us that

$$\begin{aligned}\sigma(\mathbf{x}) &= (1 - x_s)\hat{\sigma}(\hat{\mathbf{x}}) + x_s(1 + \hat{\sigma}(\hat{\mathbf{x}} + \hat{\mathbf{y}})) \\ &= \sum_{\substack{v \in V \\ v \neq s}} x_v \sigma(\mathbf{1}_v) + x_s(1 + \hat{\sigma}(\hat{\mathbf{y}})).\end{aligned}$$

We complete our proof by observing that, in fact, $\sigma(\mathbf{1}_s)$ is precisely equal to $1 + \hat{\sigma}(\hat{\mathbf{y}})$. We can see this by once again coupling the activated sets under any vector Θ of thresholds. In particular, let $S^{\Theta}_0, \ldots, S^{\Theta}_n$ and $\hat{S}^{\Theta}_0, \ldots, \hat{S}^{\Theta}_n$ denote the sets activated in G under $\mathbf{1}_s$ and in $\hat{G}$ under $\hat{\mathbf{y}}$, respectively, in stages $0, \ldots, n$. Arguments identical to those made above allow us to conclude that for all i, we have that $S^{\Theta}_{i+1} = \hat{S}^{\Theta}_i \cup \{s\}$. Thus, by again noting that influence cascades converge after n steps we see that $S^{\Theta}_n = \hat{S}^{\Theta}_n \cup \{s\}$, and taking expectations with respect to Θ gives precisely the desired equality. □

We may also express our optimization problem on DAGs in the integral case as the following MIP:

$$\text{maximize} \sum_v (X_v + Y_v) \text{ subject to}$$

$$\begin{aligned}
X_v + Y_v &\leq 1 & \forall v \\
Y_v - \sum_{u \in \delta^-(v)} w_{uv}(Y_u + X_u) &\leq 0 & \forall v \\
\sum_v X_v &\leq K & \\
X_v &\in \{0,1\} & \forall v \\
Y_v &\in [0,1] & \forall v
\end{aligned}$$

3.7 Examples

Example 4. *Consider solving our problem on a directed graph consisting of a single (one-directional) cycle with n vertices. Assume that every edge has weight $1 - K/n$, and that thresholds on nodes are drawn from $\mathcal{U}[0,1]$. We consider the optimal integral and fractional influence to apply.*

In the fractional case, consider applying influence of exactly K/n to every node. Note that for any node, the amount of influence we apply directly plus the weight on its sole incoming edge sum to 1. Thus, any time a node's predecessor on the cycle is becomes activated, the node will becomes activated as well. Inductively, we can then see that any time at least one node is activated in the cycle, every node will eventually become activated. This means that the expected number of activated nodes under this *strategy is precisely*

$$
\begin{aligned}
&n \cdot \Pr[\textit{At least one node activates}] \\
&= n(1 - \Pr[\textit{No nodes activate}]) \\
&= n(1 - \Pr[\textit{Every node's threshold is above } K/n]) \\
&= n(1 - (1 - K/n)^n).
\end{aligned}
$$

In the integral case, however, we cannot spread our influence as evenly. Note that each node we activate has some chance to activate the nodes following it in the cycle; however, any cascade must stop once we reach the next node we directly activated. If we have an interval of length ℓ *between directly activated nodes (including the initial node we activate directly in the length), we can see that the expected number of nodes activated in the interval is*

$$
\begin{aligned}
&\sum_{i=1}^{\ell} \Pr[\textit{Node } i \textit{ in the interval is activated}] \\
&= \sum_{i=1}^{\ell} \Pr[\textit{Nodes } 2, 3, \ldots, i \textit{ have thresholds below } 1 - K/n] \\
&= 1 + (1 - K/n) + (1 - K/n)^2 + \cdots + (1 - K/n)^{(\ell-1)} \\
&= \frac{1 - (1 - K/n)^{\ell}}{K/n}.
\end{aligned}
$$

While this tells us the expected value for a single interval, we want to know the expected value summed over all intervals. Observing from the sum form that the benefit of adding another node to an interval is strictly decreasing in the length of the interval, we can see that we should always make the lengths of the intervals as close to equal as possible. Noting that the lengths of the intervals always sum to n*, then, we can see* *that the total number of nodes activated in expectation is bounded by*

$$K\frac{1-(1-K/n)^{n/K}}{K/n} = n(1-(1-K/n))^{n/K}.$$

Note, however, that if we choose $K \approx \ln n$, we get that

$$\frac{1-(1-K/n)^{n/K}}{1-(1-K/n)^{n}} \approx 1-1/e.$$

3.8 Harndess

In this section, we present NP-Hardness and inapproximability results in the linear influence model. We assume that thresholds are not chosen from a distribution, and they are fixed and given as part of the input. We note that this is the main assumption that makes our problem intractable, and to achieve reasonable algorithms, one has to make some stochastic (distributional) assumptions on the thresholds. In Section 3.4, we introduced the linear influence model as a special case of fractional influence model, but it makes sense to define it as a special case of integral influence model as well. In the fractional linear influence model, we are allowed to apply any influence vector $\mathbf{x} \in [0,1]^n$ on nodes. By restricting the influence vector $\mathbf{x}$ to be in $\{0,1\}^n$ (a binary vector), we achieve the integral version of linear influence model. Our hardness results in Theorem 8, and Corollary 9 work for both fractional and integral versions of linear influence model. We start by proving that the linear influence model is NP-Hard with a reduction from independent set problem in Theorem 8. We strengthen this hardness result in Corollary 9 by showing that an $n^{1-\epsilon}$ approximation algorithm for the linear influence problem yields an exact algorithm for it as well for any constant $\epsilon > 0$, and therefore even an $n^{1-\epsilon}$ approximation algorithm is NP-Hard to achieve. At the end, we show the that it is NP-Hard to achieve any better than $1-1/e$ approximation in the Triggering model which is introduced in [75]. We will elaborate on the Triggering

Model and this hardness result at the end of this section.

Theorem 8. *If we allow arbitrary, fixed thresholds, it is NP-Hard to compute for a given instance of the integral linear influence problem (G, k, T) (graph G, budget k, and a target goal T) whether or not there exists a set S of k vertices in G such that $\sigma(S) \geq T$. Furthermore, the same holds in the factional version of the problem (instead of a set S of size k, we should look for a influence vector with ℓ_1 norm equal to k in the fractional case).*

Proof. We show hardness by reducing from Independent Set. Given a problem instance (G, k) of IS, we construct a two-layer DAG as follows. Let $G = (V, E)$ denote the vertices and (undirected) edges of G. The first layer L_1 consists of one vertex for every vertex $v \in V$; we abuse notation and refer to the vertex in L_1 corresponding to $v \in V$ as v as well. The second layer contains vertices based on the edges in E. For each unordered pair of vertices $\{u, v\}$ in V, we add vertices to the second layer L_2 based on whether $\{u, v\}$ is an edge in G: if $\{u, v\} \in E$, then we add a single vertex to L_2 with (directed) edges from each of $u, v \in L_1$ to it; if $\{u, v\} \notin E$, then we add two vertices to L_2, and add (directed) edges going from $u \in L_1$ to the first of these and from $v \in L_1$ to the second of these. We set all activation thresholds and all edge weights in our new DAG to $1/2$. We claim that there exists a set $S \subseteq L_1 \cup L_2$ satisfying $|S| \leq k$ and $\sigma(S) \geq kn$ if and only if G has an independent set of size k.

First, we note that in our constructed DAG, sets $S \subseteq L_1$ always dominate sets containing elements outside of L_1, in the sense that for any $T \subseteq L_1 \cup L_2$ there always exists a set $S \subseteq L_1$ such that $|S| \leq |T|$ and $\sigma(S) \geq \sigma(T)$. Consider an arbitrary such T. Now, consider any vertex $v \in T \cap L_2$. By construction, there exists some $u \in L_1$ such that (u, v) is an edge in our DAG. Note that $|T \setminus \{v\} \cup \{u\}| \leq |T|$ and $\sigma(T \setminus \{v\} \cup \{u\}) \geq \sigma(T)$. Thus, if we repeatedly replace T with $T \setminus \{v\} \cup \{u\}$ for each such v, we eventually with have the desired set S.

With the above observation in hand, we can be assured that there exists set S of k

vertices in our constructed DAG such that $\sigma(S) \geq nk$ if and only if there exists such an $S \subseteq L_1$. Recall how we constructed the second layer of our DAG: each vertex $v \in L_1$ has precisely $(n-1)$ neighbors; and two vertices $u, v \in L_1$ share a neighbor if and only if they are neighbors in the original graph G, in which case they have exactly one shared neighbor. Thus, we can see that for any set of vertices $S \subseteq L_1$, we have that

$$\sigma(S) = n|S| - |\{\{u, v\} \in E : u, v \in S\}|.$$

Thus, we can see that for any set $S \subseteq L_1$, we have $\sigma(S) \geq n|S|$ if and only if $\{u, v\} \notin E$ for any $u, v \in S$, i.e. S is an independent set in G. The main claim follows.

Furthermore, recall that in our constructed DAG, every edge weight and threshold was exactly equal to $1/2$. It is not hard to see, therefore, that in the fractional case it is never optimal to place an amount of influence on a vertex other than 0 or $1/2$. It follows, therefore, that there is a $1-1$ correspondence between optimal optimal solutions in the integral case with budget k and in the fractional case with budget $k/2$. Thus, as claimed, the hardness extends to the fractional case. □

Corollary 9. *If we allow arbitrary, fixed thresholds, it is NP-Hard to approximate the linear influence problem to within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$. Furthermore, the same holds for the fractional version of our problem.*

Proof. We show that given an instance (G, k) of Target Set Selection, and a target T, we can construct a new instance (G', k), such that if we can approximate the optimal solution for the new instance (G', k) to within a factor of $n^{1-\varepsilon}$, then we can tell whether the original instance (G, k) had a solution with objective value at least T. The claim then follows by applying Theorem 8.

Fix some $\delta > 0$. Let n be the number of vertices in G. Note that we must have that $0 < k < T \leq n$, since if any one of these inequalities fails to hold, the question of whether or not (G, k) has a solution with objective value at least T can be answered trivially. Let $N = \lceil (2n^2)^{1/\delta} \rceil$; we construct G' from G by adding N identical new vertices to it. Let v be one of our new vertices. For every vertex u that was present in G, we add an edge from u to v in G', with weight $1/n$. We set the threshold of v to be precisely T/n.

Consider what the optimal objective value in (G, k) implies about the optimal objective value in (G', k). If there exists some solution to the former providing objective value at least T, then we can see that the same solution will activate every one of the new vertices in G' as well, and so produce an objective value of at least $T + N$. On the other hand, assume every solution to G has objective value strictly less than T. Note that in the case of fixed thresholds, the activation process is deterministic, and so we may conclude that every solution has objective value at most $T - 1$. Now, this means that no matter what choices we make in G' about the vertices inherited from G, every one of the new vertices will require at least $1/n$ additional influence to become activated. Thus, no solution for (G', k) can achieve objective value greater than $(T - 1) + kn$, in either the integral or fractional case. By our choice of N, however, we can then conclude that the optimal solution for (G', k) has value at least

$$T + N > N \geq (2n^2)^{1/\delta} > (T - 1 + kn)^{1/\delta},$$

the value of the optimal solution for (G', k) in the latter case raised to the power of $1/\delta$. Thus, for any fixed $\varepsilon > 0$, we can choose an appropriate $\delta > 0$ such the new instance (G', k) has increased in size only polynomially from (G, k), but applying an $n^{1-\varepsilon}$ approximation to (G', k) will allow us to distinguish whether or not (G, k) had a solution with objective value at least T, exactly as desired. □

Before stating Theorem 10, we should define the triggering model introduced in [75]. In this model, each node v independently chooses a random triggering set T_v

according to some distribution over subsets of its neighbors. To start the process, we target a set A for initial activation. After this initial iteration, an inactive node v becomes active in step t if it has a neighor in its chosen triggering set T_v that is active at time $t-1$. For our purposes, the distributions of triggering sets have support size one (deterministic triggering sets). We also show that our hardness result even holds when the size of these sets is two.

Theorem 10. *It is NP-Hard to approximate linear influence problem to within any factor better than* $1-1/e$*, even in the Triggering model where triggering sets have size at most* 2*.*

Proof. We prove this by reducing from the Max Coverage problem, which is NP-Hard to approximate within any factor better than $1-1/e$. Let $(\mathcal{S}, k)$ be an instance of Max Coverage, where $\mathcal{S} = \{S_1, \ldots, S_m\}$ and $S_j \subseteq [n]$ for each $j = 1, \ldots, m$. We begin by showing a reduction to an instance of Target Set Selection in the Triggering model; later, we argue that we can do so while ensuring that triggering sets have size at most 2.

We construct a two layer DAG instance of Target Set Selection as follows. First, fix a large integer N; we will pick the exact value of N later. The first layer L_1 will contain m vertices, each corresponding to one of the sets in $\mathcal{S}$. The second layer L_2 contains nN vertices, N of which correspond to each $i \in [n]$. We add directed edges from the vertex in L_1 corresponding to S_j to all N vertices in L_2 corresponding to i for each $i \in S_j$. We set all thresholds and weights in the DAG to 1. Note that this corresponds exactly to the triggering model, where each vertex in the first layer has an empty triggering set and each vertex in the second layer has a triggering set consisting of exactly the nodes in L_1 corresponding to S_j that contain it. This completes the description of the reduction.

Now, we consider the maximal influence we can achieve by selecting k vertices in our constructed DAG. First, we note that we may assume without loss of generality that we only consider choosing vertices from L_1. This is because we can only improve the number of activated sets by replacing any vertex from L_2 with a vertex that has an edge to it; if we have already selected all such vertices, then we can simply replacing it with an arbitrary vertex from L_1 and be no worse off. Note, however, that if we select some set $W \subseteq L_1$ of vertices to activate, we will have that

$$\sigma(W) = |W| + N|\cup_{j\in W} S_j|.$$

Let $W^* \in \operatorname{argmax}_{W \subset L_1 : |W| \leq k} \sigma(W)$. Now, if we have an α-approximation algorithm for Target Set Selection, we can find some $W \subseteq L_1$ such that $|W| \leq k$ and $\sigma(W) \geq \alpha\sigma(W^*)$. But this means that

$$|W| + N|\cup_{j\in W} S_j| \geq \alpha \left(|W^*| + N|\cup_{j\in W^*} S_j|\right),$$

which implies that

$$|\cup_{j\in W} S_j| \geq \alpha|\cup_{j\in W^*} S_j| - m/N. \tag{3.3}$$

Thus, for any $\varepsilon > 0$, by picking $N = \lceil m/\varepsilon \rceil$ we can use our α-approximation algorithm for Target Set Selection to produce an α-approximation for Max Coverage with an additive loss of ε. Since the objective value for our problem is integral, we may therefore conclude it is NP-Hard to approximate Target Set Selection within a factor of $1 - 1/e$.

In the above reduction, our targeting sets could be as large as m. We know show that we can, in fact, ensure that no targeting set has size greater than 2. In particular, the key insight is that activation effectively functions as an OR-gate over the targeting set. We can easily replace an OR-gate with fan-in of f by a tree of at most $\log(f)$ OR-gates, each with fan-in 2. It is easy to see that if we add such trees of OR-gates before L_2, we increase the loss term in Equation (3.3) to at most $m \log(m)/N$. We can easily offset this by increasing N appropriately, and so retain our conclusion even when targeting sets have size at most 2. □

Example 5. *The following example shows that when thresholds are fixed, the optimal objective values in the fractional and integral cases can differ by as much as a factor of n, where n is the number of vertices in the graph. The instance we consider is a DAG consisting of a single, directed path of n vertices. Each edge in the path has weight $1/(n+1)$, and every vertex on the path has threshold $2/(n+1)$. Note that since thresholds are strictly greater than edge weights, and every vertex, being on a simple path, has in degree at most one, it is impossible for a vertex to be activated without some direct influence being applied to it.*

Consider our problem on the above graph with budget 1. In the integral case, we cannot activate more than a single vertex – as previously observed, no vertex can be activated without direct application of influence, and with a budget of 1 we can only affect on vertex directly. On the other hand, in the fractional case the following strategy guarantees that all vertices are activated. Apply $2/(n+1)$ influence to the earliest vertex, and $1/(n+1)$ influence to the remaining $(n-1)$ vertices. Now, this is sufficient to activate the earliest vertex directly; furthermore, every other vertex has sufficient direct influence that it will activate as long as the vertex before it on the path does. Thus, a simple induction proves the claim, and we can see that the optimal integral and fractional solutions differ in objective value by a factor of n.

3.9 Experimental Results

Datasets. We use the following real-world networks for evaluating our claims. The statistical information regarding these real-world networks are in Table 3.1.

- NetHEPT: This is an academic collaboration network. NetHEPT is based on

Networks	# of nodes	# of edges	Ave. deg.	Directed
NetHEPT	15233	58891	7.73	No
NetPHY	37154	231584	12.46	No
Facebook	4039	88234	21.84	No
Amazon	262111	1234877	4.71	Yes

Table 3.1: real-world networks

"High Energy Physics - Theory" section of the e-print arXiv[2] with papers from 1991 to 2003. In this network, nodes represent authors and edges represent co-authorship relations. This network is available at `http://research.microsoft.com/en-us/people/weic/graphdata.zip`.

- NetPHY: This is another academic collaboration network. NetPHY is taken from the full "Physics" section of the e-print arXiv. In this network, nodes represent authors and edges represent co-authorship relations. Graph is available at `http://research.microsoft.com/en-us/people/weic/graphdata.zip`.

- Facebook: This network represents *friend list* (*circle*) from Facebook. The data is available on `http://snap.stanford.edu/data/egonets-Facebook.html`.

- Amazon: This network is produced by crawling Amazon website based on the following observation: customers who bought product i also bought product j. In this network, nodes represent products and there is a direct edge from node i to node j if product i is frequently co-purchased with product j. This network is based on Amazon data in March 2003. The data is available at `http://snap.stanford.edu/data/amazon0302.html`.

Algorithms. We compare the following algorithms in this study. The first three algorithms are for the integral influence model, and the last three algorithms work for

[2] `http://www.arXiv.org`

the fractional influence model. Note that we need efficient algorithms which are fast enough so we can run them on a real-world network.

- DegreeInt: A simple greedy algorithm which selects nodes with the largest degrees. This method has been used in [23, 75] as well.
- DiscountInt: This is a variant of DegreeInt. This algorithm selects node u with the highest degree in each step. Moreover, after adding node u to the seed set, the algorithm decreases the degrees of neighbors of u by one. This method was also evaluated in [23].
- RandomInt: This algorithm just randomly adds B nodes to the seed set, i.e., by spending 1 on each of them. We use this algorithm as a baseline in our comparisons which is also used in [23, 24, 75]
- DegreeFrac: This algorithm selects each node fractionally proportional to its degree. In particular, this algorithm spends $\min\{\frac{Bd_i^-}{m}, 1\}$ on node i where B is the budget, d_i^- is the out-degree of node i, and m is the total number of edges[3].
- DiscountFrac: A heuristic for the fractional case according to Algorithm 1. Let $\Gamma_v^-(A)$ be the total sum of the weight of edges from node v to set A, and $\Gamma_v^+(A)$ be the total sum of the weight of edges from set A to node v. This algorithm starts with an empty seed set S, and in each step it adds node $v \notin S$ with the maximum $\Gamma_v^-(V-S)$ to seed set S by spending $\max\{0, 1-\Gamma_v^+(S)\}$ on node v. Note that in each step the total influence from the current seed set S to node v is $\Gamma_v^+(S)$, and it is enough to spend $1-\Gamma_v^+(S)$ for adding node v to the current seed set S. Note that no node would pay a positive amount, and the algorithm spends $\max\{0, 1-\Gamma_v^+(S)\}$ on node v.
- UniformFrac: This algorithm distributes the budget equally among all nodes. We use this algorithm as another baseline in our comparisons.

Algorithm 1 DiscountFrac

Input: Graph $G = (V, E)$ and budget B
Output: Influencing vector $\mathbf{x}$

$S \leftarrow 0$
$b \leftarrow B$
$\mathbf{x} \leftarrow \mathbf{0}$
while $b > 0$ **do**
  $u \leftarrow \text{argmax}_{v \in V-S}\{\Gamma_v^-(V-S)\}$
  $x_u \leftarrow \min\{b, \max\{0, 1 - \Gamma_u^+(S)\}\}$
  $b \leftarrow b - x_u$
  $S \leftarrow S \cup \{u\}$
end while
return $\mathbf{x}$

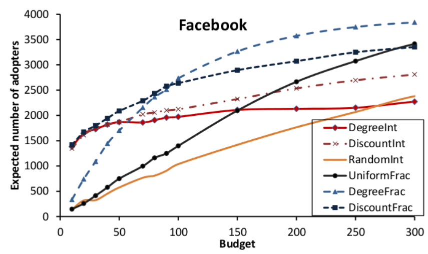
All these heuristic algorithms are fast and are designed for running on large real-world networks. In particular, algorithms DegreeInt and DegreeFrac only need the degree nodes. The running time of algorithms DiscountInt and DiscountFrac are $O(n \log n + m)$ using a heap. Algorithm DiscountInt was proposed as an efficient algorithm for the integral influence model in [23].At last we note that algorithms RandomInt and UniformFrac are linear-time algorithms. It also has been shown that the performance of DiscountInt almost matches the performance of the greedy algorithm which maximizes a submodular function [23]. Hence, it seems DiscountInt is an appropriate candidate for evaluating the power of the integral influence model.

Results. We have implemented all algorithms in $C++$ and run all experiments on a server with two 6-core/12-thread 3.46 GHz Intex Xeon X5690 CPUs, with 48 GB 1.3 GHz RAM. We run all of the aforementioned algorithms for finding the activation vector/set, and compute the performance of each algorithm by running 10000 simulations and taking the average of the number of adopters.

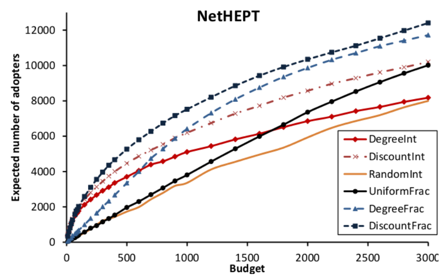
We first examine the performance of a fractional activation vector in the *weighted cascade model*. In the weighted cascade model the weight of the edge from u to v is $\frac{1}{d_v^-}$, where d_v^- is the in-degree of node v. Note that in the weighted cascade model the

[3] If the graph is undirected we should use $2m$ instead of m

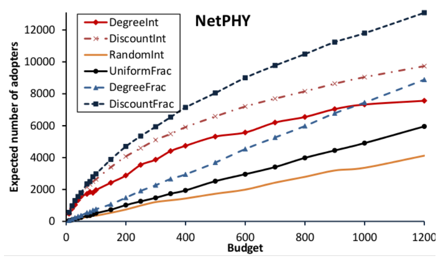
total sum of weight of incoming edges of each node is $\sum_{uv} w_{uv} = \sum_{uv} \frac{1}{d_v^-} = 1$. This model was proposed by Kempe, Kleinberg, and Tardos [75], and it has been used in the literature, e.g., see [23, 24, 25]. We run all algorithms on aforementioned real-world networks, and compare their performance with various budget in Figure 3.1.



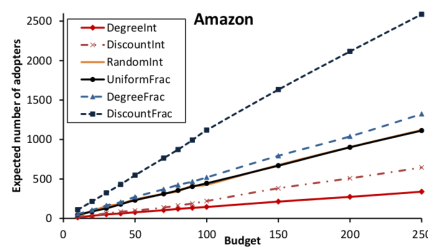
(a) Facebook



(b) NetHEPT



(c) NetPHY

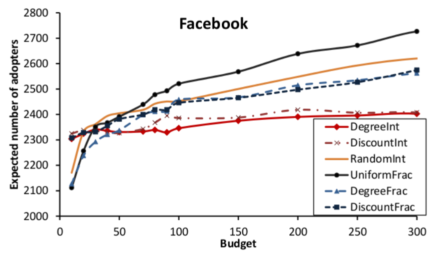


(d) Amazon

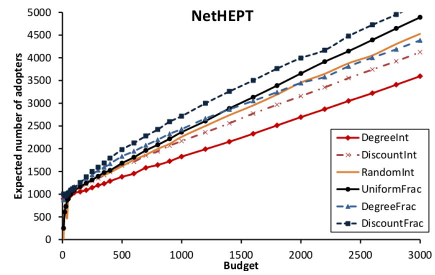
Figure 3.1: Performance of different algorithms on Facebook, NetHEPT, NetPHY, and Amazon. The weights of edges are defined based on the weighted cascade model.

We then compare the performance of various algorithms when the weight of edges are determined by *TRIVALENCY* model. In the TRIVALENCY model the weight of each edge is chosen uniformly at random from the set $\{0.001, 0.01, 0.1\}$, where 0.001, 0.01, and 0.1 represent low, medium, and high influences. Note that in this model the total sum of the weights of incoming edges of each node may be greater than 1. This model and its variants have been used in [23, 24, 75]. We run all proposed algorithms on real-world networks when their weights are defined by TRIVALENCY model. Results are shown in Figure 3.2.

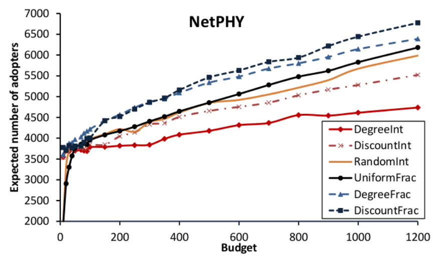
Discussion. In most of the plots, algorithms for the fractional influence model do



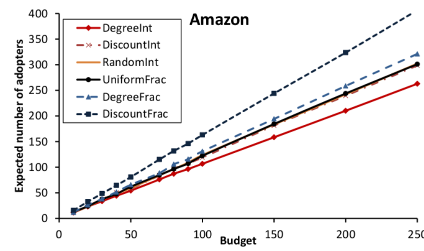
(a) Facebook



(b) NetHEPT



(c) NetPHY



(d) Amazon

Figure 3.2: Performance of different algorithms on Facebook, NetHEPT, NetPHY, and Amazon. The weights of edges are defined based on the TRIVALENCY model.

substantially better than algorithms for the integral influence model. Overall, for most datasets, `DiscountFrac` is the best algorithm, with the only exception being the Facebook dataset. As a simple metric of the power of the fractional model versus the integral model, we consider the pointwise performance gain of fractional model algorithms versus the integral model algorithms. i.e., for a given budget, we compute the ratio of expected number of adopters for the fractional model with the most adopters and the expected number of adopters for the integral model algorithm with the most adopters. Depending on the dataset, we get a mean pointwise performace gain between 3.4% (Facebook dataset, TRIVALENCY model) and 142.7 % (Amazon dataset, weighted cascade model) with the mean being 31.5% and the median being 15.7 % over all the datasets and both models (weighted cascade and TRIVALENCY). Among the heuristics presented for the integral model, `DiscountInt` is probably the best. If we compare just it to its fractional adaptation, `DiscountFrac` ,we get a similar picture: the range of average performace gain is between 9.1 % (Facebook, TRIVALENT model) and 397.6 % (Amazon, weighted cascade model) with a mean of 64.1 % and a median of 15.6 %.

In summary, the experimental results clearly demonstrate that the fractional model leads to a significantly higher number of adopters across a wide range of budgets on diverse datasets.

Chapter 4

Network Cournot Competition

4.1 Introduction

In this chapter we study selling a utility with a distribution network, e.g., natural gas, water and electricity, in several markets when the clearing price of each market is determined by its supply and demand. The distribution network fragments the market into different regional markets with their own prices. Therefore, the relations between suppliers and submarkets form a complex network [36, 37, 49, 53, 69, 98, 131]. For example, a market with access to only one supplier suffers a monopolistic price, while a market having access to multiple suppliers enjoys a lower price as a result of the price competition.

Antoine Augustin Cournot introduced the first model for studying the duopoly competition in 1838. He proposed a model where two individuals own different springs of water, and sell it independently. Each individual decides on the amount of water to supply, and then the aggregate water supply determines the market price through an inverse demand function. Cournot characterizes the unique equilibrium outcome of

[0] This is a joint work with M. Abolhassani, M. H. Bateni, M. T. Hajiaghayi and H. Mahini. A version of this work appeared in *WINE '15* [1].

the market when both suppliers have the same marginal costs of production, and the inverse demand function is linear. He argued that in the unique equilibrium outcome, the market price is above the marginal cost.

Joseph Bertrand 1883 criticized the Cournot model, where the strategy of each player is the quantity to supply, and in turn suggested to consider prices, rather than quantities, as strategies. In the Bertrand model each firm chooses a price for a homogeneous good, and the firm announcing the lowest price gets all the market share. Since the firm with the lowest price receives all the demand, each firm has incentive to price below the current market price unless the market price matches its cost. Therefore, in an equilibrium outcome of the Bertrand model, assuming all marginal costs are the same and there are at least two competitors in the market, the market price will be equal to the marginal cost.

The Cournot and Bertrand models are two basic tools for investigating the competitive market price, and have attracted much interest for modeling real markets; see, e.g., [36, 37, 53, 131]. While these are two extreme models for analyzing the price competition, it is hard to say which one is essentially better than the other. In particular, the predictive power of each strongly depends on the nature of the market, and varies from application to application. For example, the Bertrand model explains the situation where firms literally set prices, e.g., the cellphone market, the laptop market, and the TV market. On the other hand, Cournot's approach would be suitable for modeling markets like those of crude oil, natural gas, and electricity, where firms decide about quantities rather than prices.

There are several attempts to find equilibrium outcomes of the Cournot or Bertrand competitions in the oligopolistic setting, where a small number of firms compete in only one market; see, e.g., [60, 64, 80, 105, 121, 132]. Nevertheless, it is not entirely clear what equilibrium outcomes of these games are when firms compete over more than one market. In this chapter, we investigate the problem of finding equilibrium outcomes of the Cournot competition in a network setting where there are several markets for a homogeneous good and each market is accessible to a subset of firms.

4.1.1 Example

We start with the following warm-up example. This is a basic example for the Cournot competition in the network setting. It consists of three scenarios. We assume firm $i \in \{A, B\}$ produces quantity q_{ij} of the good in market $j \in \{1, 2\}$. Let $\mathbf{q}$ be the vector of all quantities.

Scenario 1 Consider the Cournot competition in an oligopolistic setting with two firms and one market (see Figure 4.1). Let $p(\mathbf{q}) = 1 - q_{A1} - q_{B1}$ be the market price (the inverse demand function), and $c_i(\mathbf{q}) = \frac{1}{2}q_{i1}^2$ be the cost of production for firm $i \in \{A, B\}$. The profit of a firm is what it gets by selling all the quantities of good it produces in all markets minus its cost of production. Therefore, the profit of firm i denoted by $\pi_i(\mathbf{q})$ is $q_{i1}(1-q_{A1}-q_{B1})-\frac{1}{2}q_{i1}^2$. In a Nash equilibrium of the game, each firm maximizes its profit assuming its opponent does not change its strategy. Hence, the unique Nash equilibrium of the game can be found by solving the set of equations $\frac{\partial \pi_A}{\partial q_{A1}} = \frac{\partial \pi_B}{\partial q_{B1}} = 0$. So $q_{A1} = q_{B1} = \frac{1}{4}$ is the unique Nash equilibrium where $p(\mathbf{q}) = \frac{1}{2}$, and $\pi_A(\mathbf{q}) = \pi_B(\mathbf{q}) = 0.9375$.

Scenario 2 We construct the second scenario by splitting the market in the previous scenario into two identical markets such that both firms have access to both markets (see Figure 4.1). Since the demand is divided between two identical markets, the price for market j would be $p_j(\mathbf{q}) = 1-2q_{Aj}-2q_{Bj}$, i.e., the clearance price of each market is the same as the clearance price of the market in Scenario 1, when the supply is half of the supply of the market in Scenario 1. In this scenario, the profit of firm $i \in \{A, B\}$ is $\pi_i(\mathbf{q}) = \sum_j q_{ij}(1-2q_{Aj}-2q_{Bj})-\frac{1}{2}(q_{i1}+q_{i2})^2$. Any Nash equilibrium of this game satisfies the set of equations $\frac{\partial \pi_A}{\partial q_{A1}} = \frac{\partial \pi_A}{\partial q_{A2}} = \frac{\partial \pi_B}{\partial q_{B1}} = \frac{\partial \pi_B}{\partial q_{B2}} = 0$. By finding the unique solution to this set of equations, one

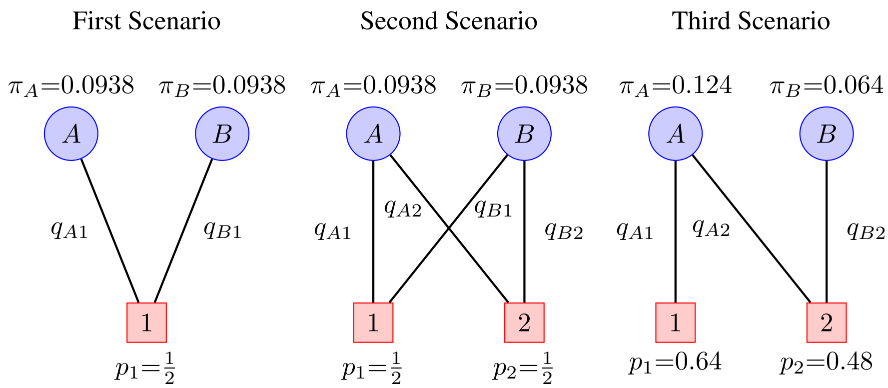


Figure 4.1: This figure represents the three scenarios of our example. Vector $\mathbf{q} = (\frac{1}{4}, \frac{1}{4})$ represents the unique equilibrium in the first scenario. Vector $\mathbf{q} = (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$ is the unique equilibrium of the second scenario. Finally, Vector $\mathbf{q} = (0.18, 0.1, 0.16)$ is the unique equilibrium in the third scenario.

can verify that $\mathbf{q} = (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$ is the unique equilibrium of the game where $p_1(\mathbf{q}) = p_2(\mathbf{q}) = \frac{1}{2}$, and $\pi_A(\mathbf{q}) = \pi_B(\mathbf{q}) = 0.09375$. Since we artificially split the market into two identical markets, this equilibrium is, not surprisingly, the same as the equilibrium in the previous scenario.

Scenario 3 Consider the previous scenario, and suppose firm 2 has no access to the first market (see Figure 4.1). Let the demand functions and the cost functions be the same as the previous scenario. The profits of firms 1 and 2 can be written as follows:

$$\begin{aligned}
\pi_A(\mathbf{q}) &= q_{A1}(1 - 2q_{A1}) + q_{A2}(1 - 2q_{A2} - 2q_{B2}) - \frac{1}{2}(q_{A1} + q_{A2})^2, \\
\pi_B(\mathbf{q}) &= q_{B2}(1 - 2q_{A2} - 2q_{B2}) - \frac{1}{2}q_{B2}^2.
\end{aligned}$$

The unique equilibrium outcome of the game is found by solving the set of equations $\frac{\partial \pi_A}{\partial q_{A1}} = \frac{\partial \pi_A}{\partial q_{A2}} = \frac{\partial \pi_B}{\partial q_{B2}} = 0$. One can verify that vector $\mathbf{q} = (q_{A1}, q_{A2}, q_{B2}) = (0.18, 0.1, 0.16)$ is the unique equilibrium outcome of the game where $p_1(\mathbf{q}) = 0.64$, $p_2(\mathbf{q}) = 0.48$, $\pi_A(\mathbf{q}) = 0.124$, and $\pi_B(\mathbf{q}) = 0.064$.

The following are a few observations worth mentioning.

- Firm A has more power in this scenario due to having a captive market[1].
- The equilibrium price of market 1 is higher than the equilibrium price in the previous scenarios.
- The position of firm B affects its profit. Since it has no access to market 1, it is not as powerful as firm A.
- The equilibrium price of market 2 is smaller than the equilibrium price in the previous scenarios.

4.1.2 Related Work

There are several papers that investigate the Cournot competition in an oligopolistic setting (see, e.g., [60, 64, 80, 121, 132]). In spite of these works, little is known about the Cournot competition in a network. Ilkılıç [66] studies the Cournot competition in a network setting, and considers a network of firms and markets where each firm chooses a quantity to supply in each accessible market. He studies the competition when the inverse demand functions are linear and the cost functions are quadratic (functions of the total production). In this study, we consider the same model when the cost functions and the demand functions may have quite general forms. We show the game with linear inverse demand functions is a potential game and therefore has a unique equilibrium outcome. Furthermore, we present two polynomial-time algorithms for finding an equilibrium outcome for a wide range of cost functions and demand functions. While we investigate the Cournot competition in networks, there is a recent paper which considers the Bertrand competition in network setting [13], albeit in a much more restricted case of only two firms competing in each market.

The final price of each market in the Cournot competition is a market clearing price; i.e, the final price is set such that the market becomes clear. Finding a market

[1] A captive market is one in which consumers have limited options and the seller has a monopoly power.

clearance equilibrium is a well-established problem, and there are several papers which propose polynomial-time algorithms for computing equilibriums of markets in which the price of each good is defined as the price in which the market clears. Examples of such markets include Arrow-Debreu market and its special case Fisher market (see related work on these markets [40, 41, 42, 47, 54, 68, 104]). Devanur and Vazirani [41] design an approximation scheme which computes the market clearing prices for the Arrow-Debreu market, and Ghiyasvand and Orlin [54] improve the running time of the algorithm. The first polynomial-time algorithm for finding an Arrow-Debreu market equilibrium is proposed by Jain [68] for a special case with linear utilities. The Fisher market, a special case of the Arrow-Debreu market, attracted a lot of attention as well. Eisenberg and Gale [47] present the first polynomial-time algorithm by transferring the problem to a concave cost maximization problem. Devanur et al. [42] design the first combinatorial algorithm which runs in polynomial time and finds the market clearance equilibrium when the utility functions are linear. This result is later improved by Orlin [104].

For the sake of completeness, we refer to recent works in the computer science literature [51, 67], which investigate the Cournot competition in an oligopolistic setting. Immorlica et al. [67] study a coalition formation game in a Cournot oligopoly. In this setting, firms form coalitions, and the utility of each coalition, which is equally divided between its members, is determined by the equilibrium of a Cournot competition between coalitions. They prove the price of anarchy, which is the ratio between the social welfare of the worse stable partition and the social optimum, is $\Theta(n^{2/5})$ where n is the number of firms. Fiat et al. [51] consider a Cournot competition where agents may decide to be non-myopic. In particular, they define two principal strategies to maximize revenue and profit (revenue minus cost) respectively. Note that in the classic Cournot competition all agents want to maximize their profit. However, in their study each agent first chooses its principal strategy and then acts accordingly. The authors prove

Table 4.1: Summary of Results

Cost functions	**Inverse demand functions**	**Running time**	**Technique**
Convex	Linear	$O(E^3)$	Convex optimization, formulation as an ordinal potential game
Convex	Strongly monotone marginal revenue function[2]	poly(E)	Reduction to a nonlinear complementarity problem
Convex, separable	Concave	$O(n \log^2 Q_{\max})$	Supermodular optimization, nested binary search

this game has a pure Nash equilibrium and the best response dynamics will converge to an equilibrium. They also show the equilibrium price in this game is lower than the equilibrium price in the standard Cournot competition.

4.1.3 Results and techniques

We consider the problem of Cournot competition on a network of markets and firms for different classes of cost and inverse demand functions. Adding these two dimensions to the classical Cournot competition which only involves a single market and basic cost and inverse demand functions yields an engaging but complicated problem which needs advanced techniques for analyzing. For simplicity of notation we model the competition by a bipartite graph rather than a hypergraph: vertices on one side denote the firms, and vertices on the other side denote the markets. An edge between a firm and a market demonstrates willingness of the firm to compete in that specific market. The complexity of finding the equilibrium, in addition to the number of markets and firms, depends on the classes that inverse demand and production cost functions belong to.

We summarize our results in Table 4.1.

In the above table, E denotes the number of edges of the bipartite graph, n de-

[2]Marginal revenue function is the vector function which maps production quantities for an edge to marginal revenue along that edge.

notes the number of firms, and $Q_{\max}$ denotes the maximum possible total quantity in the oligopoly network at any equilibrium. In our results we assume the inverse demand functions are nonincreasing functions of total production in the market. This is the basic assumption in the classical Cournot Competition model: As the price in the market increases, it is reasonable to believe that the buyers drop out of the market and demand for the product decreases. The classical Cournot Competition model as well as many previous works on Cournot Competition model assumes linearity of the inverse demand function [66, 67]. In fact there is little work on generalizing the inverse demand function in this model. The second and third row of the above table shows we have developed efficient algorithms for more general inverse demand functions satisfying concavity rather than linearity. This can be accounted as a big achievement. The assumption of monotonicity of the inverse demand function is a standard assumption in Economics [6, 7, 93]. We assume cost functions to be convex which is the case in many works related to both Cournot Competition and Bertrand Network [81, 135]. In a previous work [66], the author considered Cournot Competition on a network of firms and markets; however, assumed that inverse demand functions are linear and all the cost functions are quadratic function of the total production by the firm in all markets which is quite restrictive. Most of the results in other related works in Cournot Competition and Bertrand Network require linearity of the cost functions [13, 67]. A brief summary of our results presented in three sections is given below.

Linear Inverse Demand Functions

In case inverse demand functions are linear and production costs are convex, we present a fast and efficient algorithm to obtain the equilibrium. This approach works by showing that Network Cournot Competition belongs to a class of games called *ordinal potential games*. In such games, the collective strategy of the independent players is to maximize a single potential function. The potential function is carefully designed in such a way that changes made by one player reflects in the same way in the potential function as in their own utility function. We design a potential function for the game, which depends on the network structure, and show how it captures this property. Moreover, in the case where the cost functions are convex, we prove concavity of this designed potential function (Theorem 16) concluding convex optimization methods can be employed to find the optimum and hence, the equilibrium of the original Cournot competition. We also discuss uniqueness of equilibria in case the cost functions are strictly concave. Our result in this section is specifically interesting since we find the unique equilibrium of the game. We prove the following theorems in Section 4.3.

Theorem 11. *The Network Cournot Competition with linear inverse demand functions forms an ordinal potential game.*

Theorem 12. *Our designed potential function for the Network Cournot Competition with linear inverse demand functions is concave provided that the cost functions are convex. Furthermore, the potential function is strictly concave if the cost functions are strictly convex, and hence the equilibria for the game is unique. In addition, a polynomial-time algorithm finds the optimum of the potential function which describes the market clearance prices.*

The general case

Since the above approach does not work for nonlinear inverse demand functions, we design another interesting but more involved algorithm to capture more general forms of inverse demand functions. We show that an equilibrium of the game can be computed in polynomial time if the production cost functions are convex and the revenue function is monotone. Moreover, we show under strict monotonicity of the revenue function, the solution is unqiue, and therefore our results in this section is structural; i.e. we find the one and only equilibria. For convergence guarantee we also need Lipschitz condition on derivatives of inverse demand and cost functions. We start the section by modeling our problem as a complementarity problem. Then we prove how holding the aforementioned conditions for cost and revenue functions yields satisfying *Scaled Lipschitz Condition* (SLC) and semidefiniteness for matrices of derivatives of the profit function. SLC is a standard condition widely used in convergence analysis for scalar and vector optimization [142]. Finally , we present our algorithm, and show how meeting these new conditions by inverse demand and cost functions helps us to guarantee polynomial running time of our algorithm. We also give examples of classes of inverse demand functions satisfying the above conditions. These include many families of inverse demand functions including quadratic functions, cubic functions and entropy functions. The following theorem is the main result of Section 4.4 which summarizes the performance of our algorithm.

Theorem 13. *A solution to the Network Cournot Competition can be found in polynomial number of iterations under the following conditions:*

1. *The cost functions are (strongly) convex.*
2. *The marginal revenue function is (strongly[3]) monotone.*
3. *Ther first derivative of cost functions and inverse demand functions and the second derivative of inverse demand functions are Lipschitz continuous.*

Furthermore, the solution is unique assuming only the first condition. Therefore, our algorithm finds the unique equilibrium of NCC.

Cournot oligopoly

Another reasonable model for considering cost functions of the firms is the case where the cost of production in a market depends only on the quantity produced by the firm in that specific market (and not on quantities produced by this firm in other markets).

[3]For at least one of the first two conditions, strong version of condition should be satisfied, i.e., either cost functions should be strongly convex or the marginal revenue function should be strongly monotone.

In other words, the firms have completely independent sections for producing different goods in various markets, and there is no correlation between cost of production in separate markets. Interestingly, in this case the competitions are separable; i.e. equilibrium for Network Cournot Competition can be found by finding the quantities at equilibrium for each market individually. This motivates us for considering Cournot game where the firms compete over a single market. We present a new algorithm for computing equilibrium quantities produced by firms in a Cournot oligopoly, i.e., when the firms compete over a single market. Cournot Oligopoly is a well-known model in Economics, and computation of its Cournot Equilibrium has been subject to a lot of attention. It has been considered in many works including [21, 78, 91, 103, 127] to name a few. The earlier attempts for calculating equilibrium for a general class of inverse demand and cost functions are mainly based on solving a Linear Complementarity Problem or a Variational Inequality. These settings can be then turned into convex optimization problems of size $O(n)$ where n is the number of firms. This means the runtime of the earlier works cannot be better than $O(n^3)$ which is the best performance for convex optimization [20]. We give a novel combinatorial algorithm for this important problem when the quantities produced are integral. We limit our search to integral quantities for two reasons. First, in real-world all commodities and products are traded in integral units. Second, this algorithm can easily be adapted to compute approximate Cournot-Nash equilibrium for the continuous case and since the quantities at equilibrium may not be rational numbers, this is the best we can hope for. Our algorithm runs in time $O(n \log^2(Q_{\max}))$ where $Q_{\max}$ is an upper bound on total quantity produced at equilibrium. Our approach relies on the fact that profit functions are supermodular when the inverse demand function is nonincreasing and the cost functions are convex. We leverage the supermodularity of inverse demand functions and Topkis' Monotonicity Theorem [128] to design a nested binary search algorithm. The following is the main result of Section 4.5.

Theorem 14. *A polynomial-time algorithm successfully computes the quantities produced by each firm at an equilibrium of the Cournot oligopoly if the inverse demand function is non-increasing, and the cost functions are convex. In addition, the algorithm runs in $O(n \log^2(Q_{\max}))$ where $Q_{\max}$ is the maximum possible total quantity in the oligopoly network at any equilibrium.*

4.2 Notations

Suppose we have a set of n firms denoted by $\mathcal{F}$ and a set of m markets denoted by $\mathcal{M}$. A single good is produced in each of these markets. Each firm might or might not be able to supply a particular market. A bipartite graph is used to demonstrate these relations. In this graph, the markets are denoted by the numbers $1, 2, \ldots, m$ on one side, and the firms are denoted by the numbers $1, 2, \ldots, n$ on the other side. For simplicity, throughout the chapter we use the notation $i \in \mathcal{M}$ meaning the market i, and $j \in \mathcal{F}$ meaning firm j. For firm $j \in \mathcal{F}$ and market $i \in \mathcal{M}$ there exists an edge between the corresponding vertices in the bipartite graph if and only if firm j is able to produce the good in market i. This edge will be denoted (i, j). The set of edges of the graph is denoted by $\mathcal{E}$, and the number of edges in the graph is shown by E. For each market $i \in \mathcal{M}$, the set of vertices $N_{\mathcal{M}}(i)$ is the set of firms that this market is connected to in the graph. Similarly, $N_{\mathcal{F}}(j)$ denotes the set of neighbors of firms j among markets. The edges in $\mathcal{E}$ are sorted and numbered $1, \ldots, E$, first based on the number of their corresponding market and then based on the number of their corresponding firm. More formally, edge $(i, j) \in \mathcal{E}$ is ranked above edge $(l, k) \in \mathcal{E}$ if $i < l$ or $i = l$ and $j < k$. The quantity of the good that firm j produces in market i is denoted by q_{ij}. The vector $\mathbf{q}$ is an $E \times 1$ vector that contains all the quantities produced over the edges of the graph in the same order that the edges are numbered.

The demand for good i, denoted D_i, is the sum of the total quantity of this good produced by all firms, i.e., $D_i = \sum_{j \in N_{\mathcal{M}}(i)} q_{ij}$. The price of good i, denoted by the function $P_i(D_i)$, is only a decreasing function of total demand for this good and not the individual quantities produced by each firm in this market. For a firm j, the vector $\vec{s_j}$ denotes the strategy of firm j, which is the vector of all quantites produced by this firm in the markets $N_{\mathcal{F}}(j)$. Firm $j \in \mathcal{F}$ has a cost function related to its strategy denoted by $c_j(\vec{s_j})$. The profit that firm j makes is equal to the total money that it obtains by selling its production minus its cost of production. More formally, the profit of firm j, denoted by π_j, is

$$\pi_j = \sum_{i \in N_{\mathcal{F}}(j)} P_i(D_i) q_{ij} - c_j(\vec{s_j}). \tag{4.1}$$

4.3 Cournot competition and potential games

In this section, we design an efficient algorithm for the case where the price functions are linear. More specifically, we design an innovative *potential function* that captures the changes of all the utility functions simultaneously, and therefore, show how finding the quantities at the equilibrium would be equivalent to finding the set of quantities that maximizes this function. We use the notion of *potential games* as introduced in Monderer and Shapley [94]. In that paper, the authors introduce *ordinal potential games* as the set of games for which there exists a *potential function* P^* such that the pure strategy equilibrium set of the game coincides with the pure strategy equilibrium set of a game where every party's utility function is P^*.

In this section, we design a function for the Network Cournot Competition and show how this function is a potenial function for the problem if the price functions are linear. Interestingly, this holds for any cost function meaning Network Cournot Competition with arbitrary cost functions is an ordinal potential game as long as the price functions are linear. Furthermore, we show when the cost functions are convex, our designed potential function is concave, and hence any convex optimization method can find the equilibrium of such a Network Cournot Competition. In case cost functions are strictly convex, the potential function is strictly concave. We restate a well known theorem in this section to conclude that the convex optimization in this case has a unique solution, and therefore the equilibria that we find in this case is the one and only equilibria of the game.

Definition 4. *A game is said to be an* ordinal potential game *if the incentive of all players to change their strategy can be expressed using a single global function called the potential function. More formally, a game with n players and utility function u_i for player $i \in \{1, \ldots, n\}$ is called ordinal potential with potential function P^* if for all the strategy profiles $q \in \mathbb{R}^n$ and every strategy x_i of player i the following holds:*

$$u_i(x_i, q_{-i}) - u_i(q_i, q_{-i}) > 0 \ \textit{ iff } \ P^*(x_i, q_{-i}) - P^*(q_i, q_{-i}) > 0.$$

An equivalent definition of an ordinal potential game is a game for which a potential funciton P^* exists such that the following holds for all strategy profiles $q \in \mathbb{R}^n$ and for each player i.

$$\frac{\partial u_i}{\partial q_i} = \frac{\partial P^*}{\partial q_i}.$$

In other words, for each strategy profile q, any change in the strategy of player i has the same impact on its utility function as on the game's potential function.

The pure strategy equilibrium set of any ordinal potential game coincides with the pure strategy equilibrium set of a game with the potential function P^* as all parties' utility function.

Theorem 15. *The Network Cournot Competition with linear price functions is an ordinal potential game.*

Proof. Let $P_i(D_i) = \alpha_i - \beta_i D_i$ be the linear price function for market $i \in \mathcal{M}$ where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are constants determined by the properties of market i. Note that this function is decreasing with respect to D_i. Here we want to introduce a potential function P^*, and show that $\frac{\partial \pi_j}{\partial q_{ij}} = \frac{\partial P^*}{\partial q_{ij}}$ holds $\forall (i,j) \in \mathcal{E}$. The utility function of firm j is

$$\pi_j = \sum_{i \in N_{\mathcal{F}}(j)} \left(\alpha_i - \beta_i \sum_{k \in N_{\mathcal{F}}(j)} q_{kj} \right) q_{ij} - c_j(\vec{s_j}),$$

and taking partial derivative with respect to q_{ij} yields

$$\frac{\partial \pi_j}{\partial q_{ij}} = \alpha_i - \beta_i \sum_{k \in N_{\mathcal{F}}(j)} q_{kj} - \beta_i q_{ij} - \frac{\partial c_j(\vec{s_j})}{\partial q_{ij}}.$$

We define P^* to be

$$P^* = \sum_{i \in \mathcal{M}} \left[\alpha_i \sum_{j \in N_{\mathcal{M}}(i)} q_{ij} - \beta_i \sum_{j \in N_{\mathcal{M}}(i)} q_{ij}^2 - \beta_i \sum_{\substack{k \leq j \\ k,j \in N_{\mathcal{M}}(i)}} q_{ij} q_{ik} - \sum_{j \in N_{\mathcal{M}}(i)} \frac{c_j(\vec{s_j})}{|N_{\mathcal{F}}(j)|} \right],$$

whose partial derivative with respect to q_{ij} is

$$\begin{aligned}
\frac{\partial P^*}{\partial q_{ij}} &= \alpha_i - 2\beta_i q_{ij} - \frac{\partial}{\partial q_{ij}} \left(\beta_i \sum_{\substack{l \leq m \\ l,m \in N_{\mathcal{M}}(i)}} q_{il} q_{im} \right) - \frac{\partial c_j(\vec{s_j})}{\partial q_{ij}} \\
&= \alpha_i - 2\beta_i q_{ij} - \beta_i (D_i - q_{ij}) - \frac{\partial c_j(\vec{s_j})}{\partial q_{ij}} \\
&= \frac{\partial \pi_j}{\partial q_{ij}}.
\end{aligned}$$

Since this holds for each $i \in \mathcal{M}$ and each $j \in \mathcal{F}$, the Network Cournot Competition is an ordinal potential game. □

We can efficiently compute the equilibrium of the game if the potential function P^* is easy to optimize. Below we prove that this function is concave.

Theorem 16. *The potential function P^* from the previous theorem is concave provided that the cost functions of the firms are convex. Moreover, if the cost functions are strictly convex then the potential function is strictly concave.*

Proof. The proof goes by decomposing P^* into pieces that are concave. We first define f for one specific market i as

$$f = \sum_{j \in N_{\mathcal{M}}(i)} q_{ij}^2 + \sum_{\substack{k \leq j \\ k,j \in N_{\mathcal{M}}(i)}} q_{ij} q_{ik},$$

and prove that it is convex.

Recall that $\mathbf{q}$ is an $E \times 1$ vector of all the quantities of good produced over the existing edges of the graph. We can write $f = \mathbf{q}^T M \mathbf{q}$ where M is an $E \times E$ matrix with all elements on its diagonal equal to 1 and all other elements equal to $\frac{1}{\sqrt{2}}$:

$$M = \begin{bmatrix} 1 & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 1 & \cdots & \frac{1}{\sqrt{2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & 1 \end{bmatrix}.$$

To show that f is convex, it suffices to prove that M is positive semidefinite, by finding a matrix R such that $M = R^T R$. Consider the following $(E+1) \times E$ matrix:

$$R = \begin{bmatrix} c & c & \cdots & c \\ a & 0 & \cdots & 0 \\ 0 & a & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a \end{bmatrix},$$

where a, c are set below. Let R_i be the i-th column of R. We have $R_i \cdot R_i = a^2 + c^2$ and $R_i \cdot R_j = c^2$ for $i \neq j$.

Setting $c = 2^{-\frac{1}{4}}$ and $a = \sqrt{1 - c}$ yields $M = R^T R$, showing that M is positive semidefinite, hence the convexity of f.

The following expression for a fixed market $i \in \mathcal{M}$, sum of three concave functions, is also concave.

$$\alpha_i \sum_{j \in N_{\mathcal{M}}(i)} q_{ij} - \beta_i \Bigg(\sum_{j \in N_{\mathcal{M}}(i)} q_{ij}^2 + \sum_{\substack{k \leq j \\ k,j \in N_{\mathcal{M}}(i)}} q_{ij} q_{ik} \Bigg) - \sum_{j \in N_{\mathcal{M}}(i)} \frac{c_j(\vec{s_j})}{|N_{\mathcal{F}}(j)|}.$$

Summing over all markets proves concavity of P^*. Note that if a function is the sum of a concave function and a strictly concave function, then it is strictly concave itself. Therefore, since f is concave, we can conclude strictly concavity of P^* under the assumption that the cost functions are strictly convex. □

The following well-known theorem discusses the uniqueness of the solution to a convex optimization problem.

Theorem 17. *Let $F : \mathcal{K} \to \mathbb{R}^n$ be a strictly concave and continuous function for some finite convex space $\mathcal{K} \in \mathbb{R}^n$. Then the following convex optimization problem has a unique solution.*

$$\max f(x) \;\; s.t. \;\; x \in \mathcal{K}. \tag{4.2}$$

By Theorem 16, if the cost functions are strictly convex then the potential function is strictly concave and hence, by Theorem 17 the equilibrium of the game is unique.

Let $ConvexP(\mathcal{E}, (\alpha_1, \dots, \alpha_m), (\beta_1, \dots, \beta_m), (c_1, \dots, c_n))$ be the following convex optimization program:

$$\min \quad -\sum_{i \in \mathcal{M}} \Bigg[\alpha_i \sum_{j \in N_{\mathcal{M}}(i)} q_{ij} - \beta_i \sum_{j \in N_{\mathcal{M}}(i)} q_{ij}^2 - \beta_i \sum_{\substack{k \leq j \\ k,j \in N_{\mathcal{M}}(i)}} q_{ij} q_{ik} - \sum_{j \in N_{\mathcal{M}}(i)} \frac{c_j(\vec{s_j})}{|N_{\mathcal{F}}(j)|} \Bigg] \tag{4.3}$$

$$\text{subject to} \qquad q_{ij} \geq 0 \;\; \forall (i,j) \in \mathcal{E}.$$

Note that in this optimization program we are trying to maximize P^* for a bipartite graph with set of edges $\mathcal{E}$, linear price functions characterized by the pair (α_i, β_i) for each market $i \in \mathcal{M}$, and cost functions c_j for each firm $j \in \mathcal{F}$.

Algorithm 2 Compute quantities at equilibrium for the Network Cournot Competition.

procedure COURNOT-POTENTIAL($\mathcal{E}$, c_j, (α_i, β_i)) ▷ Set of edges, cost functions and price functions

Use a convex optimization algorithm to solve

$$ConvexP(\mathcal{E}, (\alpha_1, \dots, \alpha_m), (\beta_1, \dots, \beta_m), (c_1, \dots, c_n)).$$

and return the vector **q** of equilibrium quantities.

end procedure

The above algorithm has a time complexity equal to the time complexity of a convex optimization algorithm with E variables. The best such algorithm has a running time $O(E^3)$[20].

4.4 Finding equilibrium for cournot game with general cost and inverse demand functions

In this section, we formulate an algorithm for a much more general class of price and cost functions. Our algorithm is based on reduction of Network Cournot Competition (NCC) to a polynomial time solvable class of Non-linear Complementarity Problem (NLCP). First in Subsection 4.4.1, we introduce our marginal profit function as the vector of partial derivatives of all firms with respect to the quantities that they produce. Then in Subsection 4.4.2, we show how this marginal profit function can help us to reduce NCC to a general NLCP. We also discuss uniqueness of equilibrium in this situation which yields the fact that solving NLCP would give us the one and only equilibrium of this problem. Unfortunately, in its most general form, NLCP is computationally intractable. However, for a large class of functions, these problems are polynomial time solvable. Most of the rest of this section is dedicated to proving the fact that NCC is polytime solvable on vast and important array of price and cost functions. In Subsection 4.4.3, we rigorously define the conditions under which NLCP is polynomial time solvable. We present our algorithm in this subsection along with a theorem which shows it converges in polynomial number of steps. To show the conditions that we introduce for convergence of our algorithm in polynomial time are not restrictive, we give a discussion in Subsection 4.4.4 on the functions satisfying these conditions, and show they hold for a wide range of price functions.

Assumptions Throughout the rest of this section we assume that the price functions are decreasing and concave and the cost functions are strongly convex (*The notion of strongly convex is to be defined later*). We also assume that for each firm there is a finite quantity at which extra production ceases to be profitable even if that is the only firm operating in the market. Thus, all production quantities and consequently quantities supplied to markets by firms are finite. In addition, we assume Lipschitz continuity and finiteness of the first and the second derivatives of price and cost functions. We note that these Lipschitz continuity assumptions are very common for convergence analysis in convex optimization [20] and finiteness assumptions are implied by Lipschitz continuity. In addition, they are not very restrictive as we don't expect unbounded fluctuation in costs and prices with change in supply. For sake of brevity, we use the terms inverse demand function and price function interchangeably.

4.4.1 Marginal profit function

For the rest of this section, we assume that P_i and c_i are twice differentiable functions of quantities. We define f_{ij} for a firm j and a market i such that $(i, j) \in \mathcal{E}$ as follows.

$$f_{ij} = -\frac{\partial \pi_j}{\partial q_{ij}} = -P_i(D_i) - \frac{\partial P_i(D_i)}{\partial q_{ij}} q_{ij} + \frac{\partial c_j}{\partial q_{ij}}. \tag{4.4}$$

Recall that the price function of a market is only a function of the total production in that market and not the individual quantities produced by individual firms. Thus $\frac{\partial P_i(D_i)}{\partial q_{ij}} = \frac{\partial P_i(D_i)}{\partial q_{ik}}$ $\forall j, k \in N_{\mathcal{M}}(i)$. Therefore, we replace these terms by $P_i'(D_i)$.

$$f_{ij} = -P_i(D_i) - P_i'(D_i)q_{ij} + \frac{\partial c_j}{\partial q_{ij}}. \tag{4.5}$$

Let vector F be the vector of all f_{ij}'s corresponding to the edges of the graph in the same format that we defined the vector q. That is f_{ij} corresponding to $(i, j) \in \mathcal{E}$ appears above f_{lk} corresponding to edge $(l, k) \in \mathcal{E}$ iff $i < l$ or $i = l$ and $j < k$. Note that F is a function of q.

Moreover, we separate the part representing marginal revenue from the part representing marginal cost in function F. More formally, we split F into two functions R and S such that $F = R + S$, and the element corresponding to the edge $(i, j) \in \mathcal{E}$ in the *marginal revenue fuction* $R(q)$ is:

$$r_{ij} = -\frac{\partial \pi_j}{\partial q_{ij}} = -P_i(D_i) - P'(D_i)q_{ij},$$

whereas for the *marginal cost function* $S(q)$, it is:

$$s_{ij} = \frac{\partial c_j}{\partial q_{ij}}.$$

4.4.2 Non-linear complementarity problem

In this subsection we formally define the non-linear complementarity problem (NLCP), and prove our problem is a NLCP.

Definition 5. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuously differentiable function on $\mathbb{R}^n_+$. The complementarity problem seeks a vector $x \in \mathbb{R}^n$ that satisfies the following con-*

straints:

$$\begin{aligned} x, F(x) &\geq 0, \\ x^T F(x) &= 0. \end{aligned} \tag{4.6}$$

Theorem 18. *The problem of finding the vector q at equilibrium in the Cournot game is a complementarity problem.*

Proof. Let q^* be the vector of the quantities at equilibrium. All quantities must be nonnegative at all times; i.e., $q^* \geq 0$. It suffices to show $F(q^*) \geq 0$ and ${q^*}^T F(q^*) = 0$. At equilibrium, no party benefits from changing its strategy, in particular, its production quantities. For each edge $(i, j) \in \mathcal{E}$, if the corresponding quantity q^*_{ij} is positive, then q^*_{ij} is a local maxima for π_j; i.e., $f_{ij}(q^*) = -\left.\frac{\partial \pi_j}{\partial q_{ij}}\right|_{q^*} = 0$. On the other hand, if $q^*_{ij} = 0$, then $\left.\frac{\partial \pi_j}{\partial q_{ij}}\right|_{q^*}$ cannot be positive, since, if it is, firm j would benefit by increasing the quantity q_{ij} to a small amount ϵ. Therefore, $\left.\frac{\partial \pi}{\partial q_{ij}}\right|_{q^*}$ is always nonpositive or equivalently $f_{ij}(q^*) \geq 0$, i.e., $F(q^*) \geq 0$. Also, as we mentioned above, a nonzero q^*_{ij} is a local maximum for π_j; i.e., $f_{ij}(q^*) = -\left.\frac{\partial \pi}{\partial q_{ij}}\right|_{q^*} = 0$. Hence, either $q^*_{ij} = 0$ or $f_{ij}(q^*) = 0$; thus, $q^*_{ij} f_{ij}(q^*) = 0$. This yields $\sum_{(i,j)\in\mathcal{E}} q^*_{ij} f_{ij}(q^*) = {q^*}^T F(q^*) = 0$. □

Definition 6. *$F : \mathcal{K} \to \mathbb{R}^n$ is said to be strictly monotone at x^* if*

$$\langle (F(x) - F(x^*))^T, x - x^* \rangle \geq 0, \forall x \in \mathcal{K}. \tag{4.7}$$

F is said to be strictly monotone if it is monotone at any $x^* \in \mathcal{K}$. Equivalently, F is strictly monotone if the jacobian matrix is positive definite.

The following theorem is a well known theorem for Complementarity Problems.

Theorem 19. *Let $F : \mathcal{K} \to \mathbb{R}^n$ be a continuous and strictly monotone function with a point $x \in \mathcal{K}$ such that $F(x) \geq 0$ (i.e. there exists a potential solution to the CP). Then* *the Complementarity Problem introduced in (4.6) characterized by function F has a unique solution.*

Hence, the Complementarity Problem characterized by function F introduced element by element in (4.4) has a unique solution under the assumption that the revenue function is strongly monotone (special case of strictly monotone). Note that the marginal profit function or F in our case is non-negative in at least one point. Otherwise, no firm has any incentive to produce in any market and the equilibrium is when all production quantities are equal to zero. In the next subsection, we aim to find this unique equilibrium of the NCC problem.

4.4.3 Designing a polynomial-time algorithm

In this subsection, we introduce Algorithm 3 for finding equilibrium of NCC, and show it converges in polynomial time by Theorem 27. This theorem requires the marginal profit function to satisfy Scaled Lipschitz Condition(SLC) and monotonicity. We first introduce SLC, and show how the marginal profit function satisfies SLC and montonicty by Lemmas 20 to 26. We argue the conditions that the cost and price functions should have in order for the marginal profit function to satisfy SLC and monotonicity in Lemma 26. Finally, in Theorem 27, we show convergence of our algorithm in polynomial time.

Before introducing the next theorem, we explain what the Jacobians ∇R, ∇S, and ∇F are for the Cournot game. First note that these are $E \times E$ matrices. Let $(i, j) \in \mathcal{E}$ and $(l, k) \in \mathcal{E}$ be two edges of the graph. Let e_1 denote the index of edge (i, j), and e_2 denote the index of edge (l, k) in the graph as we discussed in the first section. Then the element in row e_1 and column e_2 of matrix ∇R, denoted $\nabla R_{e_1 e_2}$, is equal to $\frac{\partial r_{ij}}{\partial q_{lk}}$. We name the corresponding elements in ∇F and ∇S similarly. We have $\nabla F = \nabla R + \nabla S$ as $F = R + S$.

Definition 7 (Scaled Lipschitz Condition (SLC))**.** *A function $G : D \mapsto \mathbb{R}^n$, $D \subseteq \mathbb{R}^n$*

is said to satisfy Scaled Lipschitz Condition (SLC) *if there exists a scalar* $\lambda > 0$ *such that* $\forall\ h \in \mathbb{R}^n, \forall\ x \in D$*, such that* $\|X^{-1}h\| \leq 1$*, we have:*

$$\|X[G(x+h) - G(x) - \nabla G(x)h]\|_\infty \leq \lambda |h^T \nabla G(x)h|, \tag{4.8}$$

where X *is a diagonal matrix with diagonal entries equal to elements of the vector* x *in the same order, i.e.,* $X_{ii} = x_i$ *for all* $i \in \mathcal{M}$*.*

Satisfying SLC and monotonicity are essential for marginal profit function in Theorem 27. In Lemma 26 we discuss the assumptions for cost and revenue function under which these conditions hold for our marginal profit function. We use Lemmas 20 to 26 to show F satisfies SLC. More specifically, we demonstrate in Lemma 20, if we can derive an upperbound for LHS of SLC for R and S, then we can derive an upperbound for LHS of SLC for $F = R + S$ too. Then in Lemma 21 and Lemma 22 we show LHS of S and R in SLC definition can be upperbounded. Afterwards, we show monotonicity of S in Lemma 25. In Lemma 26 we aim to prove F satifies SLC under some assumptions for cost and revenue functions. We use the fact that LHS of SLC for F can be upperbounded using Lemma 22 and Lemma 21 combined with Lemma 20. Then we use the fact that RHS of SLC can be upperbounded using strong monotonicity of R and Lemma 25. Using these two facts, we conclude F satisfies SLC in Lemma 26.

Lemma 20. *Let* F, R, S *be three* $\mathbb{R}^n \to \mathbb{R}^n$ *functions such that* $F(q) = R(q) + S(q), \quad \forall q \in \mathbb{R}^n$*. Let* R *and* S *satisfy the following inequalities for some* $C > 0$ *and* $\forall\ h$ *such that* $\|X^{-1}h\| \leq 1$*:*

$$\|X[R(q+h) - R(q) - \nabla R(q)h]\|_\infty \leq C\|h\|^2,$$

$$\|X[S(q+h) - S(q) - \nabla S(q)h]\|_\infty \leq C\|h\|^2,$$

where X is the diagonal matrix with $X_{ii} = q_i$. Then we have:

$$\|X[F(q+h) - F(q) - \nabla F(q)h]\|_\infty \leq 2C\|h\|^2.$$

Proof. Definition of function F implies

$$\begin{aligned}\|X[F(q+h) - F(q) - \nabla F(q)h]\|_\infty =& \|X[R(q+h) - R(q) - \nabla R(q)h] \\ &+ X[S(q+h) - S(q) - \nabla S(q)h]\|_\infty\end{aligned}$$

applying triangle inequality gives

$$\begin{aligned}\|X[F(q+h) - F(q) - \nabla F(q)h]\|_\infty \leq& \|X[R(q+h) - R(q) - \nabla R(q)h]\|_\infty \\ &+ \|X[S(q+h) - S(q) - \nabla S(q)h]\|_\infty\end{aligned}$$

Combining with assumptions of the lemma, we have the required inequality. □

The following lemmas give upper bounds for LHS of the SLC for S and R respectively.

Lemma 21. *Assume X is the diagonal matrix with $X_{ii} = q_i$. $\forall\, h$ such that $\|X^{-1}h\| \leq 1$, there exists a constant $C > 0$ satisfying: $\|X[S(q+h) - S(q) - \nabla S(q)h]\|_\infty \leq C\|h\|^2$.*

Proof. Let $m_{ij} = \frac{\partial c_i}{\partial q_{ij}}$. The element of vector $X(S(q+h) - S(q) - h\nabla S)$ corresponding to edge (i,j) is given by:

$$q_{ij}(m_{ij}(q+h) + m_{ij}(q) - h\nabla c_i(q))$$

Let $2L_3$ be an upper bound of Lipschitz constants for derivates of c_i's. Then, from

Theorem 23 and upper bound Q on production quantities, we have:

$$|q_{ij}(m_{ij}(q+h) + m_{ij}(q) - h\nabla c_i(q))| \leq QL_3\|h\|^2$$

$\square$

Lemma 22. *Assume X is the diagonal matrix with $X_{ii} = q_i$. $\forall\, h$ such that $\|X^{-1}h\| \leq 1$, $\exists C > 0$ such that $\|X[R(q+h) - R(q) - \nabla R(q)h]\|_\infty \leq C\|h\|^2$.*

Proof. Before we proceed, we state the following theorem from analysis and Lemma 24.

Theorem 23. *[20] Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a continuously differentiable function with Lipschitz gradient, i.e., for some scaler $c > 0$,*

$$\|\nabla f(x) - \nabla f(y)\| \leq c\|x - y\| \;\; \forall\, x, y \in \mathbb{R}^n.$$

Then, we have $\forall\, x, y \in \mathbb{R}^n$,

$$f(y) \leq f(x) + \nabla f(x)^T(x - y) + \frac{c}{2}\|y - x\|^2 \tag{4.9}$$

Lemma 24. *For any vector $x \in \mathbb{R}^n$ and an arbitrary $S \subseteq [n]$, let $X = \sum_{i \in S} x_i$. Then we have $\sqrt{n}\|x\| \geq X$*

Proof. Let $Y = \sum_{i \in [n]} |x_i|$. Clearly, $|Y| \geq |X|$.

$$Y^2 = \sum_{i,j \in [n]} |x_i x_j| = \sum_{i<j} 2|x_i x_j| + \|x\|^2$$

Since, $s^2 + t^2 \geq 2st \;\forall\, s, t \in \mathbb{R}$, we have

$$X^2 \leq Y^2 \leq \sum_{i<j}(x_i^2 + x_j^2) + \|x\|^2 = n\|x\|^2$$

□

Now we are ready to prove Lemma 22. First note that $R(q+h)-R(q)-\nabla R(q)h$ is an $E\times 1$ vector. Let $H_i=\sum_{j\in N_{\mathcal{M}}(i)} h_{ij}$. The element corresponding to edge $(i,j)\in\mathcal{E}$ in vector $R(q+h)$ is $P_i(D_i+H_i)+P_i'(D_i+H_i)(q_{ij}+h_{ij})$. Similarly, the element corresponding to edge $(i,j)\in\mathcal{E}$ in $R(q)$ is $P_i(D_i)+P_i'(D_i)q_{ij}$ whereas the corresponding element in $\nabla R(q)h$ is $\sum_{k\in N_{\mathcal{M}}} h_{ik}\frac{\partial r_{ij}}{\partial q_{ik}}=-\sum_{k\in N_{\mathcal{M}}(i)} h_{ik}(P_i'(D_i)+P_i''(D_i)q_{ij})+h_{ij}P_i'(D_i)$. Therefore, the element corresponding to edge $(i,j)\in\mathcal{E}$ in vector $R(q+h)-R(q)-\nabla R(q)h$ is:

$$
\begin{aligned}
-P_i(D_i+H_i)-&P_i'(D_i+H_i)(q_{ij}-h_{ij})-P_i(D_i)-P_i'(D_i)q_{ij}\\
&+\sum_{k\in N_{\mathcal{M}}(i)} h_{ik}(P_i'(D_i)+P_i''(D_i)q_{ij})+h_{ij}P_i'(D_i).
\end{aligned}
$$

Besides, X is the diagonal matrix of size $E\times E$ with diagonal entries equal to elements of q in the same order. Therefore, $X[R(q+h)-R(q)-\nabla R(q)h]$ is an $E\times 1$ vector where the element corresponding to edge $(i,j)\in\mathcal{E}$ is q_{ij} multiplied by the element corresponding to edge (i,j) in vector $R(q+h)-R(q)-\nabla R(q)h$:

$$
\begin{aligned}
&-q_{ij}\Bigg(P_i(D_i+H_i)+P_i'(D_i+H_i)(q_{ij}+h_{ij})-P_i(D_i)-P_i'(D_i)q_{ij}\\
&\qquad-\sum_{k\in N_{\mathcal{M}}(i)} h_{ik}(P_i'(D_i)+P_i''(D_i)q_{ij})-h_{ij}P_i'(D_i)\Bigg)\\
=&-q_{ij}\Bigg(\left[P_i(D_i+H_i)-P_i(D_i)-H_iP_i'(D_i)\right]\\
&\qquad+\left[P_i'(D_i+H_i)-P_i'(D_i)-H_iP_i''(D_i)\right](q_{ij}+h_{ij})+h_{ij}H_iP_i''(D_i)\Bigg)\\
\le& q_{ij}\Bigg(|P_i(D_i+H_i)-P_i(D_i)-H_iP_i'(D_i)|\\
&\qquad+|P_i'(D_i+H_i)-P_i'(D_i)-H_iP_i''(D_i)||(q_{ij}+h_{ij})|+|h_{ij}H_iP_i''(D_i)|\Bigg).
\end{aligned}
$$

Let P' and P'' be Lipschitz continuous functions with Lipschitz constants $2L_1$ and $2L_2$ respectively. To bound the last expression, we use Theorem 23 and Lemma 24

$$|P_i(D_i + H_i) - P_i(D_i) - H_i P_i'(D_i)| \leq L_1 H_i^2 \leq L_1 E\|h\|^2$$
$$|P_i'(D_i + H_i) - P_i'(D_i) - H_i P_i''(D_i)| \leq L_2 H_i^2 \leq L_2 E\|h\|^2$$
$$|h_{ij} H_i P_i''(D_i)| \leq E\|h\|^2 P_i''(D_i)$$

Then, from finiteness of derivatives, we have:

$$|h_{ij} H_i P_i''(D_i)| \leq E M_2 \|h\|^2$$

Thus, the LHS is bound from above by:

$$q_{ij} E\|h\|^2 (L_1 + L_2(q_{ij} + h_{ij}) + M_2)$$

Let Q be an upper bound on maximum profitable quantity for any producer in any market. Then the LHS is bound above by $C\|h\|^2$, where:

$$C = QE(L_1 + 2QL_2 + M_2) \tag{4.10}$$

□

If R is assumed to be strongly monotone, we immediately have a lower bound on RHS of the SLC for R. The following lemma gives a lower bound on RHS of the SLC for S.

Lemma 25. *If cost functions are (strongly) convex S is (strongly) monotone*[4,5,6,7].

[4]A matrix $M \in \mathbb{R}^{n \times n}$ is *strongly positive definite* iff $\forall\, x \in \mathbb{R}^n$ and some $\alpha > 0$ $x^T M x \geq \alpha \|x\|^2$.

[5]A differentiable function $f : D \mapsto \mathbb{R}^n$ is *monotone* iff its Jacobian ∇f is positive semidefinite over its domain D.

[6]A differentiable function $f : D \mapsto \mathbb{R}^n$ is *strongly monotone* iff its Jacobian ∇f is strongly positive definite over its domain, D.

[7]A twice differentiable function $f : D \mapsto \mathbb{R}$ is *strongly convex* iff its Hessian $\nabla^2 f$ is strongly positive

The following lemma combines the results of Lemma 21 and Lemma 22 using Lemma 20 to derive an upper bound for LHS of the SLC for F. We bound RHS of the SLC from below by using strong montonicity of R and Lemma 25.

Lemma 26. *F satisfies SLC and is monotone if:*

1. *Cost functions are convex.*
2. *Marginal revenue function is monotone.*
3. *Cost functions are strongly convex or marginal revenue function is strongly monotone.*

Proof. From lemmas 22, 21 and 20, RHS of SLC for F is $O(E\|h\|^2)$. If cost functions are strongly convex or marginal revenue function is strongly monotone, then from Lemma 25 and definition of strong monotonicity, the LHS of SLC for F is $\Omega(\|h\|^2)$. Thus, F satisfies SLC. We note that F is a sum of two monotone functions and hence is monotone. □

We wrap up with the following theorem, which summarizes the main result of this section. Lemma 26 guarantees that our problem satisfies the two conditions mentioned in Zhao and Han 1999. Therefore, we can prove the following theorem.

Theorem 27. *Algorithm 3 converges to an equilibrium of Network Cournot Competition in time $O\big(E^2 \log(\mu_0/\epsilon)\big)$ under the following assumptions:*

1. *The cost functions are strongly convex.*
2. *The marginal revenue function is strongly monotone.*
3. *The first derivative of cost functions and price functions and the second derivative of price functions are Lipschitz continuous.*

---

definite over its domain, D.

This algorithm outputs an approximate solution $(F(q^*), q^*)$ satisfying $(q^*)^T F(q^*)/n \leq \epsilon$ where $\mu_0 = (q_0)^T F(q_0)/n$, and $(F(q_0), q_0)$ is the initial feasible point [8].

Algorithm 3 Compute quantities at equilibrium for the Cournot game.

---

procedure NETWORK-COURNOT(P_i, c_j, ϵ) ▷ The price function P_i for each market $i \in \mathcal{M}$, the cost function c_j for each firm $j \in \mathcal{F}$, and ϵ as the desired tolerance

  Calculate vector F of length E as defined in (4.4).

  Find the initial feasible[9]solution $(F(x_0), x_0)$ for the complementarity problem. This solution should satisfy $x_0 \geq 0$ and $F(x_0) \geq 0$.

  Run Algorithm 3.1 from [142] to find the solution $(F(x^*), x^*)$ to the CP characterized by F.

  return x^* ▷ The vector q of quantities produced by firms at equilibrium

end procedure

4.4.4 Price Functions for Monotone Marginal Revenue Function

This section will be incomplete without a discussion of price functions that satisfy the convergence conditions for Algorithm 3. We will prove that a wide variety of price functions preserve monotonicity of the marginal revenue function. To this end, we prove the following lemma.

Lemma 28. *$\nabla R(q)$ is a positive semidefinite matrix $\forall\ q \geq 0$, i.e., R is monotone, provided that for all markets $|P_i'(D_i)| \geq \frac{|P_i''(D_i)|D_i}{2}$.*

Proof. Let e_1 be the index of the edge (i, j) and e_2 be the index of edge (l, k). The elements of ∇R are as follows.

$$\nabla R_{e_1 e_1} = \begin{cases} \frac{\partial r_{ij}}{\partial q_{ij}} & = -2P_i'(D_i) - P_i''(D_i)q_{ij} \text{ if } e_1 = e_2 \\ \frac{\partial r_{ij}}{\partial q_{ik}} & = -P_i'(D_i) - P_i''(D_i)q_{ij} \text{ if } i = l, j \neq k \\ \frac{\partial r_{ij}}{\partial q_{lk}} & = 0 \text{ if } i \neq l, j \neq k. \end{cases}$$

[8] Initial feasible solution can be trivially found. E.g., it can be the same production quantity along each edge, large enough to ensure losses for all firms. Such quantity can easily be found by binary search between [0, Q].

We note that since price functions are functions only of the total production in their corresponding markets and not the individual quantities produced by firms, $\frac{\partial P_i'(D_i)}{\partial q_{ij}} = \frac{\partial P_i'(D_i)}{\partial q_{ik}}$. Therefore, we have replaced the partial derivatives by $P_i''(D_i)$.

We must show $x^T \nabla R(D_i) x$ is nonnegative $\forall x \in \mathbb{R}^E$ and $\forall D_i \geq 0$.

$$
\begin{aligned}
x^T(\nabla R(D_i))x &= \sum_{(i,j)\in\mathcal{E}} \sum_{(k,l)\in\mathcal{E}} x_{ij}x_{lk}\frac{\partial r_{ij}}{\partial x_{lk}} = \sum_{i\in\mathcal{M}} \sum_{j,k\in N_{\mathcal{M}}(i)} x_{ij}x_{ik}\frac{\partial r_{ij}}{\partial x_{ik}} \\
&= \sum_{i\in\mathcal{M}} \left(\sum_{j\in N_{\mathcal{M}}(i)} x_{ij}^2 \left[-2P_i'(D_i) - P_i''(D_i)x_{ij}\right] \right. \\
&\qquad \left. + \sum_{j,k\in N_{\mathcal{M}}(i), j\neq k} x_{ij}x_{ik}\left[-P_i'(D_i) - P_i''(D_i)q_{ij}\right] \right) \\
&= -\sum_{i\in\mathcal{M}} \left(\sum_{j\in N_{\mathcal{M}}(i)} x_{ij}^2 P_i'(D_i) + \sum_{j,k\in N_{\mathcal{M}}(i)} x_{ij}x_{ik}(P_i'(D_i) + P_i''(D_i)q_{ij}) \right) \\
&= -\sum_{i\in\mathcal{M}} \left(P_i'(D_i) \sum_{j\in N_{\mathcal{M}}(i)} x_{ij}^2 + P_i'(D_i) \sum_{j,k\in N_{\mathcal{M}}(i)} x_{ij}x_{ik} + P_i''(D_i) \sum_{j,k\in N_{\mathcal{M}}(i)} x_{ij}q_{ij}x_{ik} \right) \\
&\geq -\sum_{i\in\mathcal{M}} \left(P_i'(D_i) \sum_{j\in N_{\mathcal{M}}(i)} x_{ij}^2 + P_i'(D_i) \sum_{j,k\in N_{\mathcal{M}}(i)} x_{ij}x_{ik} - |P_i''(D_i)||q||x|| \sum_{j\in N_{\mathcal{M}}(i)} x_{ij}| \right) \\
&\geq \sum_{i\in\mathcal{M}} \left(-P_i'(D_i)|x|^2 - P_i'(D_i) \left(\sum_{j\in N_{\mathcal{M}}(i)} x_{ij} \right)^2 + |P_i''(D_i)|D_i|x|| \sum_{j\in N_{\mathcal{M}}(i)} x_{ij}| \right)
\end{aligned}
$$

Since P_i's are decreasing functions, we have , $P_i'(D_i) \leq 0, \quad \forall i \in \mathcal{M}$. Thus, over domain of P_i's ($D_i \geq 0$), the above expression is non-negative if $|P_i''(D_i)|D_i \leq 2|P_i'(D_i)|$ Hence, $x^T(\nabla R(D_i))x \geq 0$ equivalently $\nabla R(D_i)$ is positive semidefinite.

□

While the above condition may seem somewhat restrictive, they allow the problem to be solved on a wide range of price functions. Intuitively, the condition implies that linear and quadratic terms dominate higher order terms. We present the following corollaries as examples of classes of functions that satisfy the above condition.

Corollary 29. *All decreasing concave quadratic price functions satisfy Lemma 28.*

Corollary 30. *All decreasing concave cubic price functions satisfy Lemma 28.*

Corollary 31. *Let $a_i \in \mathbb{R}^n_{\geq 0}$ for $i \in \{1 \dots k\}$ be arbitrary positive vectors. Let $f : \mathbb{R}^n_{\geq 0} \mapsto R$ be the following function: $f(x) = \sum_{i \in \{1 \dots k\}} (a_i^T x) \log(a_i^T x)$. Then f (and $-f$) satisfies Lemma 28.*

4.5 Algorithm for Cournot Oligopoly

In this section we present a new algorithm for computing equilibrium quantities produced by firms in a Cournot oligopoly, i.e., when the firms compete over a single market. Cournot Oligopoly is a standard model in Economics and computation of Cournot Equilibrium is an important problem in its own right. A considerable body of literature has been dedicated to this problem [21, 78, 91, 103, 127]. All of the earlier works that compute Cournot equilibrium for a general class of price and cost functions rely on solving a Linear Complementarity Problem or a Variational Inequality which in turn are set up as convex optimization problems of size $O(n)$ where n is the number of firms in oligopoly. Thus, the runtime guarantee of the earlier works is $O(n^3)$ at best. We give a novel combinatorial algorithm for this important problem when the quantities produced are integral. Our algorithm runs in time $n \log^2(Q_{max})$ where Q_{max} is an upper bound on total quantity produced at equilibrium. We note that, for two reasons, the restriction to integral quantities is practically no restriction at all. Firstly, in real-world all commodities and products are traded in integral units. Secondly, this algorithm can easily be adapted to compute approximate Cournot-Nash equilibrium for the continuous case and since the quantities at equilibrium may not be rational numbers, this is the best we can hope for.

As we have only a single market rather than a set of markets, we make a few changes to the notation. Let $[n] = \{1, \ldots, n\}$ be the set of firms competing over the single market. Let $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ be the set of all quantities produced by the firms. Note that in this case, each firm is associated with only one quantity. Let $Q = \sum_{i \in [n]} q_i$ be the sum of the total quantity of good produced in the market. In this case, there is only a single inverse demand function $P : \mathbb{Z} \mapsto \mathbb{R}_{\geq 0}$, which maps total supply, Q, to market price. We assume that price decreases as the total quantity produced by the firms increases, i.e., P is a decreasing function of Q. For each firm $i \in [n]$, the function $c_i : \mathbb{Z} \mapsto \mathbb{R}_{\geq 0}$ denotes the cost to this firm when it produces quantity q_i of the good in the market. The profit of firm $i \in [n]$ as a function of q_i and Q, denoted $\pi_i(q_i, Q)$, is $P(Q)q_i - c_i(q_i)$. Also let $f_i(q_i, Q) = \pi_i(q_i+1, Q+1) - \pi_i(q_i, Q)$ be the marginal profit for firm $i \in [n]$ of producing one extra unit of product. Although the quantities are nonnegative integers, for simplicity we assume the functions c_i, P, π_i and f_i are zero whenever any of their inputs are negative. Also, we refer to the forward difference $P(Q + 1) - P(Q)$ by $P'(Q)$.

4.5.1 Polynomial time algorithm

We leverage the supermodularity of price functions and Topkis' Monotonicity Theorem [128] (Theorem 35) to design a nested binary search algorithm which finds the equilibrium quantity vector $\mathbf{q}$ when the price function is a decreasing function of Q and the cost functions of the firms are convex. Intuitively the algorithm works as follows. At each point we guess Q' to be the total quantity of good produced by all the firms. Then we check how good this guess is by computing for each firm the set of quantites that it can produce at equilibrium if we assume the total quantity is the fixed integer Q'. We prove that the set of possible quantities for each firm at equilibrium, assuming a fixed total production, is a consecutive set of integers. Let $I_i = \{q_i^l, q_i^l + 1, \ldots, q_i^u - 1, q_i^u\}$ be the range of all possible quantities for firm $i \in [n]$ assuming Q' is the total quantity produced in the market. We can conclude Q' was too low a guess if $\sum_{i \in [n]} q_i^l > Q'$.

This implies our search should continue among total quantities above Q'. Similarly, if $\sum_{i\in[n]} q_i^u < Q'$, we can conclude our guess was too high, and the search should continues among total quantities below Q'. If neither case happens, then for each firm $i \in [n]$, there exists a $q_i' \in I_i$ such that $Q' = \sum_{i\in[n]} q_i'$ and firm i has no incentive to change this quantity if the total quantity is Q'. This means that the set $\mathbf{q}' = \{q_1', \ldots, q_n'\}$ forms an equilibrium of the game and the search is over.

Algorithm 4 Compute quantities produced by firms in a Cournot oligopoly.

procedure COURNOT-OLIGOPOLY(P, c_i) ▷ The market price function P, the cost functions c_i for each firm $i \in [n]$
    Let $Q_{\min} := 1$
    Let Q_i^* be the optimal quantity that is produced by a firm when it is the only firm in the market
    Let $Q_{\max} := \sum_{i\in[n]} Q_i^*$
    while $Q_{\min} \leq Q_{\max}$ **do**
        $Q' := \lfloor \frac{Q_{\min}+Q_{\max}}{2} \rfloor$
        for all $i \in [n]$ **do**
            Binary search to find the minimum nonnegative integer q_i^l satisfying
            $f_i(q_i^l, Q') = \pi_i(q_i^l + 1, Q' + 1) - \pi_i(q_i^l, Q') \leq 0$
            Binary search to find the maximum integer $q_i^u \leq Q' + 1$ satisfying
            $f_i(q_i^u - 1, Q' - 1) = \pi_i(q_i^u, Q') - \pi_i(q_i^u - 1, Q' - 1) \geq 0$
            Let $I_i = \{q_i^l, \ldots, q_i^u\}$ be the set of all integers between q_i^l and q_i^u.
        end for
        if $\Sigma_{i\in[n]} q_i^l > Q'$ **then**
            $Q_{\min} := Q' + 1$
        else if $\Sigma_{i\in[n]} q_i^u < Q'$ **then**
            $Q_{\max} := Q' - 1$
        else
            Find a vector of quantities $\mathbf{q} = (q_1, \ldots, q_n)$ such that $q_i \in I_i$ and $\sum_{i\in[n]} q_i = Q'$
            return q
        end if
    end while
end procedure

The pseudocode for the algorithm is provided in Algorithm 4, whose correctness we prove next. The rest of this section is dedicated to proving Theorem 39. Here, we present a brief outline of the proof. To help with the proof we define the functions F_i and G_i as follows. Let $F_i(q_i, Q) = P(Q+1)q_i + \frac{P'(Q)}{2}(q_i - \frac{1}{2})^2 - c(q_i)$. We note that the first difference of $F(q_i, Q)$ is the marginal profit for firm i for producing one more quantity given that the total production quantity is Q and firm i is producing q_i. Let $G_i(q_i, Q) = F_i(q_i, Q-1)$. The first difference of $G_i(q_i, Q)$ is the marginal loss for firm i for producing one less quantity given that the total production quantity is Q and firm i is producing q_i. Maximizers of these functions are closely related to equilibrium quantities a firm can produce given that the total quantity in market is Q. We make this connection precise and prove the validity of binary search in Lines 8-12 of Algorithm 4 in Lemma 32. In Lemma 33, we prove that F_i and G_i are supermodular functions of q_i and $-Q$. In lemmas 34 and 36, we use Topkis' Monotonicity Theorem to prove the monotonicity of maximizers of F_i and G_i. This, along with lemmas 37 and 38 leads to the conclusion that the outer loop for finding total quatity at equilibrium is valid as well and hence the algorithm is correct.

4.5.2 Proof of correctness

Throughout this section *we assume that the price function is decreasing and concave and the cost functions are convex.*

Lemma 32. *Let $q_i^*(Q) = \{q_i^l \dots q_i^u\}$, where $q_i^l = \min argmax_{q_i \in \{0 \dots Q_{max}\}} F_i(q_i, Q)$ and $q_i^u = \max argmax_{q_i \in \{0 \dots Q_{max}\}} G_i(q_i, Q)$. Then $q_i^*(Q)$ is the set of consecutive integers I_i given by binary search in lines 8-12 of Algorithm 4. This is the set of quantities firm i can produce at equilibrium given that the total quantity produced is Q.*

Proof. Again let $P'(Q) = P(Q+1) - P(Q)$ be the forward difference of the price function, and let $c_i'(q_i) = c_i(q_i+1) - c_i(q_i)$. From definition of profit function π_i and f_i, we have $f_i(q_i, Q) = P(Q+1) + P'(Q)q_i - c_i'(q_i)$. Assume Q is fixed. Suppose we have $q_i < \tilde{q}_i$. The marginal profit of firm at production quantity q_i is $P(Q+1) + P'(Q)q_i - c_i'(q_i)$ whereas the marginal profit at production quantity $\tilde{q}_i$ is $P(Q+1) + P'(Q)\tilde{q}_i - c_i'(\tilde{q}_i)$. Thus, $P(Q+1) + P'(Q)q_i > P(Q+1) + P'(Q)\tilde{q}_i$ since

$P'(Q)$ is negative (from concavity of P) and $q_i < \tilde{q}_i$. As the discrete cost functions are convex, we have $c'_i(q_i) < c'_i(\tilde{q}_i)$. This implies $f_i(q_i, Q) > f_i(\tilde{q}_i, Q)$ when $q_i < \tilde{q}_i$. Thus, for a fixed Q, $f_i(q_i, Q)$ is a non-increasing function of q_i. Similarly, we can see that $f_i(q_i, Q)$ is a non-increasing function of Q. From definitions of F_i and G_i, we have:

$$F_i(q_i + 1, Q) - F_i(q_i, Q) = f_i(q_i, Q) \tag{4.11}$$

$$G_i(q_i + 1, Q) - G_i(q_i, Q) = F_i(q_i + 1, Q - 1) - F_i(q_i, Q - 1) = f_i(q_i, Q - 1) \tag{4.12}$$

For a fixed Q, Let q_l be the minimum maximizer of $F_i(q_i, Q)$. Then $f_i(q_l - 1, Q) > 0$. Let q_u be the maximum maximizer of $G_i(q_i, Q)$. Because f_i is non-increasing, we have $f_i(q_l - 1, Q - 1) \geq f_i(q_l - 1, Q) > 0$. Thus, any number smaller than q_l cannot be a maximizer of G_i and we have $q_l \leq q_u$. Let $q \in \{q_l \dots q_u\}$. Then, because $q \geq q_l$ we have $f_i(q, Q) \leq 0$ and from $q \leq q_u$, we have $f_i(q - 1, Q - 1) \geq 0$. Thus, q is an equilibrium quantity when total production quantity is Q. If $q < q_l$, then $f_i(q, Q) > 0$ and if $q > q_u$ then $f_i(q - 1, Q - 1) > 0$. Thus $\{q_l \dots q_u\}$ is the set of equilibrium quantities. In Line 9 of Algorithm 4, we are searching for the minimum maximizer of F_i and in Line 11 we are searching for maximum maximizer of G_i. Binary search for these quantities is valid because first differences for both functions (equations 4.11 and 4.12) are decreasing. $\square$

Lemma 33. *Let $F_i^-(q_i, -Q) = F_i(q_i, Q)$ and $G_i^- = G_i(q_i, -Q)$. Then F_i^- and G_i^- are supermodular functions.*

Proof. We use the following definition from submodular optimization in the lemma.

Definition 8. *Given lattices $(X_1, \geq)$ and $(X_2, \geq)$, $f : X_1 \times X_2 \mapsto \mathbb{R}$ is supermodular iff for any $x_1, y_1 \in X_1; x_2, y_2 \in X_2$ such that $x_1 \geq y_1$ and $x_2 \geq y_2$, the following* *holds:*

$$f(x_1, y_2) - f(x_1, x_2) \geq f(y_1, y_2) - f(y_1, x_2)$$

We have, $F_i(q_i, Q) = P(Q+1)q_i + \frac{P'(Q)}{2}(q_i - \frac{1}{2})^2 - c_i(q_i)$. Let $-Q_1 > -Q_2$. Let $q_i' > q_i$. Then, we have:

$$F_i(q_i, Q_1) - F_i(q_i, Q_2) = (P(Q_1+1) - P(Q_2+1))q_i + \frac{P'(Q_1) - P'(Q_2)}{2}(q_i - \frac{1}{2})^2$$
$$F_i(q_i', Q_1) - F_i(q_i', Q_2) = (P(Q_1+1) - P(Q_2+1))q_i' + \frac{P'(Q_1) - P'(Q_2)}{2}(q_i' - \frac{1}{2})^2$$

Since P and P' are a decreasing functions, we have $P(Q_1) \geq P(Q_2)$ and $P'(Q_1) \geq P'(Q_2)$. From this and the fact that $q_i' > q_i$, we have:

$$F_i(q_i', Q_1) - F_i(q_i', Q_2) \geq F_i(q_i, Q_1) - F_i(q_i, Q_2)$$

Therefore F_i^- is a supermodular function. Since $G_i(q_i, Q) = F_i(q_i, Q-1)$, may a similar argument we can conclude that G_i^- is supermodular. □

Lemma 34. *Let* $I = \{q_i^l, \ldots, q_i^u\} = q_i^F(Q) = argmax_{q_i \in \{1 \ldots Q_{max}\}} F_i(q_i, Q)$ *and* $I' = \{q_i^{'l}, \ldots, q_i^{'u}\} = q_i^F(Q') = argmax_{q_i \in \{1 \ldots Q_{max}\}} F_i(q_i, Q')$. *Let* $Q > Q'$. *Then* $q_i^{'l} \geq q_i^l$ *and* $q_i^{'u} \geq q_i^u$.

Proof. We need the following definition and Topkis' Monotonicity Theorem for proving the lemma.

Definition 9. *Given a lattice* $(X, \geq)$, *we define* Strong Set Ordering *over* $A, B \subseteq X$. *We say* $A \geq_s B$ *iff* $\forall a \in A, \forall b \in B, \max\{a, b\} \in A \wedge \min\{a, b\} \in B$.

We note that the strong set ordering induces a natural ordering on sets of consecutive integers. Let $I_1 = \{l_1, \ldots, u_1\}$. Let $I_2 = \{l_2, \ldots, u_2\}$. Then $I_1 \geq_s I_2$ iff $l_1 \geq l_2$ and $u_1 \geq u_2$.

Theorem 35 (Topkis' Monotonicity Theorem[128]). *For any lattices* $(X, \geq)$ *and* $(T, \geq)$*, let* $f : X \times T \mapsto \mathbb{R}$ *be a supermodular function and let* $x^*(t) = argmax_{x \in X} f(x, t)$*. If* $t \geq t'$*, then* $x^*(t) \geq_s x^*(t')$*, i.e.,* $x^*(t)$ *is non-decreasing in* t*.*

We note that in the theorem above, strong set ordering is used over x^* because $argmax$ returns a set of values from lattice X.

Now we are ready to prove Lemma 34. From Lemma 33, $F_i^-(q_i, -Q)$ is a supermodular function. Thus, from Theorem 35, $q_i^F(Q) = argmax F_i(q_i, Q)$ is a non-decreasing function of $-Q$, i.e., q_i^F is a non-increasing function of Q. Thus $Q > Q' \implies I' \geq_s I$. As noted above, strong set ordering on a set of consecutive integers implies that $q_i^{'l} \geq q_i^l$ and $q_i^{'u} \geq q_i^u$. □

Lemma 36. *Let* $I = \{q_i^l, \ldots, q_i^u\} = q_i^G(Q) = argmax_{q_i \in \{1 \ldots Q_{max}\}} G_i(q_i, Q)$ *and* $I' = \{q_i^{'l}, \ldots, q_i^{'u}\} = q_i^G(Q') = argmax_{q_i \in \{1 \ldots Q_{max}\}} F_i(q_i, Q')$*. Let* $Q > Q'$*. Then* $q_i^{'l} \geq q_i^l$ *and* $q_i^{'u} \geq q_i^u$*.*

Proof. From Lemma 33, $G_i^-(q_i, -Q)$ is a supermodular function. Thus, from Theorem 35, $q_i^G(Q) = argmax F_i(q_i, Q)$ is a non-decreasing function of $-Q$, i.e., q_i^G is a non-increasing function of Q. Thus $Q > Q' \implies I' \geq_s I$. As noted above, strong set ordering on a set of consecutive integers implies that $q_i^{'l} \geq q_i^l$ and $q_i^{'u} \geq q_i^u$. □

Lemma 37. *Let* Q *be total production quantity guessed by Algorithm 4 at a step of outer binary search. Let* $I = (I_1, \ldots, I_n)$*, where* $I_i = \{q_i^l, \ldots, q_i^u\}$*, be the set of best reponse ranges of all firms if the total quantity is a fixed integer* Q*. Then, if* $\sum_{i=1}^n q_i^u < Q$*, there does not exist any equilibrium for which the total produced quantity is greater than or equal to* Q*.*

Proof. Assume for contradiction that such an equilibrium exists for total quantity $Q' > Q$. From Lemma 36, we have $q_i^G(Q) \geq_s q_i^G(Q') = \{q_i^{'l} \ldots q_i^{'u}\}$. Thus, we have

$q_i^u \geq q_i^{'u}$. Since Q' is an equilibrium quantity, $\sum_{i=1}^n q_i^{'u} \geq Q'$. Thus, we have $Q' < Q$ and this is a contradiction. □

Lemma 38. *Let Q be total production quantity guessed by Algorithm 4 at a step of outer binary search. Let $I = (I_1, \ldots, I_n)$, where $I_i = \{q_i^l, \ldots, q_i^u\}$, be the set of best reponse ranges of all firms if the total quantity is a fixed integer Q. Then, if $\sum_{i=1}^n q_i^l > Q$, there does not exist any equilibrium for which the total produced quantity is less than or equal to Q.*

Proof. Assume for contradiction that such an equilibrium exists for total quantity $Q' < Q$. From Lemma 34, we have $q_i^F(Q) \leq_s q_i^F(Q') = \{q_i^{'l} \ldots q_i^{'u}\}$. Thus, we have $q_i^l \leq q_i^{'l}$. Since Q' is an equilibrium quantity, $\sum_{i=1}^n q_i^{'l} leq Q'$. Thus, we have $Q' > Q$ and this is a contradiction. □

Finally, the results of this section culiminate in the following theorem.

Theorem 39. *Algorithm 4 successfully computes the vector $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ of quantities at one equilibrium of the Cournot oligopoly if the price function is decreasing and concave and the cost function is convex. In addition, the algorithm runs in time $O(n \log^2(Q_{\max}))$ where $Q_{\max}$ is the maximum possible total quantity in the oligopoly network at any equilibrium.*

Proof. Lemma 32 guarantees that the inner binary search successfully finds the best response range for all firms. Lemmas 37 and 38 ensure that the algorithm always continues its search for the total quantity at equilibrium in the segment where all the equilibria are. Thus, when the search is over, it must be at an equilibrium of the game if one exists. If an equilibrium does not exist, then the algorithm will stop when it has eliminated all quantities in $\{1 \ldots Q_{max}\}$ as possible total equilibrium production quantities. Let $Q_{\max}$ be the maximum total quantity possible at any equilibrium of the oligopoly network. Our algorithm performs a binary search over all possible quantities in $[1, Q_{\max}]$, and at each step finds a range of quantities for each firm $i \in [n]$

using another binary search. This means the algorithm runs in time $O(n \log^2(Q_{\max}))$. We can find an upper bound for $Q_{\max}$, noting that $Q_{\max}$ is at most the sum of the production quantites of the firms when they are the only producer in the market; i.e, $Q_{\max} \leq \sum_{i \in [n]} Q_i^*$ where $Q_i^* = q_i^*(q_i)$ is the optimal quantity to be produced by firm i when there is no other firms to compete with. □

Chapter 5

Couter-Terrorism Policies with Vector Equilibria

5.1 Introduction

The research reported in this chapter was motivated by a concrete application: how can countries trying to rein in the terrorist group Lashkar-e-Taiba[1] (LeT for short) come up with policies against them, especially if these policies need to be coordinated? In the case of a five-player game we formulated for LeT (presented later), there were wide variations of opinion amongst experts on what to do about LeT with respect to, for instance, whether India should carry out covert action, carry out coercive diplomacy, propose peace talks, or just keep the status quo. Likewise, the US has historically had multiple opposing viewpoints on whether to continue financial (development and military aid) to Pakistan, whether to carry out covert action against LeT, or do nothing.

[0]This is a joint work with J. P. Dickerson, M. T. Hajiaghayi and V. S. Subrahmanian. A version of this work appeared in ACM-TIST [116].

[1]Lashkar-e-Taiba (LeT), translated variously from Urdu into "Army of the Pure" or "Army of the Pious", is a prominent south Asian terrorist organization responsible for attacks in India, Kashmir, Pakistan, and Afghanistan, including the three days of attacks in 2008 in Mumbai, India, that resulted in the deaths of 166 innocent people [122, 124].

Analyzing the benefits of these actions even in the case of a single actor (e.g., only India or only the US) has proven challenging. The main contribution of this chapter is a multi-player, game-theoretic framework in which this specific problem can be solved.

However, we wanted to come up with a general solution, one that is applicable to many different settings. For instance, there are many applications where the "payoff matrices", usually one of the very first things needed in any game-theoretic framework, cannot be specified with accuracy. When asked about payoffs, multiple experts might express substantial disagreements. This is what happened with our LeT application. Here are some applications where multiple payoff matrices have been considered in a wide variety of settings.

1. *Socio-Cultural Behavior Modeling.* Woodley et al. 2008 propose a "Culturally Aware Response" (or CAR) framework in conjunction with the well-known World Values Survey to assess the results of different types of interactions between culturally different groups. They use multiple payoff matrices in their framework which vary based on the historical behaviors of different groups, e.g., one payoff matrix may indicate situations where a player responds in kind to responses of other players, while another payoff matrix may reflect situations where the player is largely non-violent.

2. *Open Source Software Releases.* Asundi et al. 2012 analyze the circumstances that are optimal for companies to release software. They argue that by open-sourcing a "crimped" version of their product, a company can hurt competitors, while enabling sales of a more sophisticated pay version of their product. To build their model, the authors utilize four different payoff matrices, corresponding to different regions of the parameter space that defines their model.

3. *International Climate Change Negotiations.* Pittel and Rübbelke 2012 develop a game-theoretic model of climate change negotiations building upon the well-known chicken game and the iterated prisoner's dilemma. The two games are

combined into a 3×3 game and studied under different payoff scenarios.

4. *Telecommunications.* Karami and Glisic 2010 define *asymmetric matrix games* (AMG) with which they model routing and network coding using conflict-free scheduling mechanisms. In their framework, multiple payoff matrices are defined, with one payoff matrix corresponding to each of a set of different partial possible network topologies.

Other applications include international negotiations [79] where the precise payoffs for the nations involved are viewed through different lenses by different experts. They can also include applications where there are different views on the payoffs different corporations get for taking different types of actions (e.g., raise wages for striking workers vs. shut down a factory vs. take legal action). Even a seemingly simple action such as "take legal action" can lead to a diversity of views about costs/payoffs as different views may exist on, e.g., how long the litigation will take (and hence how much it will cost). This chapter has two parts:

- *Approximate Equilibria for Multi-Player Games with Vector Payoffs.* Games with multiple payoffs were introduced by Shapley 1959. Shapley called them vector valued games and they have been extensively studied under various other names such as multicriteria games and multi-objective games. Unfortunately, for real-world applications such as the LeT application motivating this research, the computational cost of these past methods is too high. In order to address this, we introduce a novel combination of vector valued games and approximate equilibria and define new types of approximate equilibria for games with multiple players and multiple payoff matrices. We design algorithms for computing such equilibria for zero-sum games and games of low rank. For the case of rank 1 games, we give a structural result and use it to design a simple algorithm for such games. For general games we give an extension of Althöfer's Approximation Lemma 1994 for simultaneous games with multiple payoff functions (SGMs)

Table 5.1: The actions that different players can take.

Player	Action	Abbrv.
Lashkar-e-Taiba (LeT)	Launch major attacks	`attack`
	Eliminate armed wing	`eaw`
	Hold attacks	`hold`
	Do nothing	`none`
Pakistan's Government (PakG)	Prosecute LeT	`pros`
	Endorse LeT	`endorse`
	Do nothing	`none`
Pakistan's Military (PakM)	Crackdown on LeT	`crack`
	Cut support to LeT	`cut`
	Increase support to LeT	`support`
	Do nothing	`none`
India	Covert action against LeT	`covert`
	Coercive diplomacy against PakG	`coerce`
	Propose peace initiative to PakG	`peace`
	Do nothing	`none`
U.S.	Covert action against LeT	`covert`
	Cut aid to PakG	`cut`
	Expand aid to PakG	`expand`
	Do nothing	`none`

and use it to design a quasi-polynomial time approximation scheme (QPTAS) when the number of players in a game is constant (which is the case for our LeT game).

- *Application of PREVE to Generate Policies to Reduce Terror Acts by LeT.* Building on work by Dickerson et al. 2011, 2013, we then present a real-world application in which there are five parties including four governmental entities and the terrorist group Lashkar-e-Taiba (LeT). The goal was to understand whether there were any pure (or mixed) equilibria in which the group's terrorist acts could be significantly reduced. The five players considered are: the US, India, the Pakistani military, the Pakistani civilian government, and the terrorist group LeT. Table 5.1 shows the actions the players were allowed to take.

  When it comes to the application of game-theoretic reasoning to international strategic elements [117] with both state and non-state actors, the situation becomes much more complex because identifying the payoffs for different players is an enormous challenge and experts vary widely on what these payoffs are. To address this application, we asked three internationally acknowledged world experts to give us payoff matrices—and we received three payoff matrices with substantial differences between them. Leveraging the theoretical constructs and results described above, we built the PREVE (Policy Recommendation Engine based on Vector Equilibria) software suite, and used it to identify approximate equilibria in the multiple payoff game induced by the three expert payoff matrices. We present key results produced by PREVE, and analyze their strengths and weaknesses from a policy perspective.

This chapter begins with Section 5.2 introducing our LeT example briefly. As the LeT example is quite complex, a small toy example is also introduced. This toy example is used throughout the chapter in order to illustrate the various definitions and algorithms we introduce. Section 5.3, our first formal section, consists of preliminaries which cover basic game-theoretic concepts. Section 5.5 formally defines our equilibrium concepts and presents bounds on computing them under various assumptions. Section 5.6 presents a QPTAS for the general case, when the number of players is constant. Building on this QPTAS, it gives efficient algorithms for computing such equilibria and experimentally validates them on simulated data. Section 5.7 gives a brief description of the computational system we built, called PREVE, and applies it to a real-world experimental five-player game used to model LeT. Section 5.7 also summarizes results from computing equilibria from three payoff matrices (created by area experts using open source data) and presents key policy results. Section 5.8 describes related work on game-theoretic models of terrorist group behavior as well as past policy recommendations on how the US and India should deal with LeT.

5.2 Motivating Examples

In this section, we briefly describe the LeT application motivating this research. We also introduce a toy example that will be used throughout the chapter to illustrate definitions, as the full LeT example can be too complex for that.

5.2.1 Reducing Terror Attacks by LeT

Lashkar-e-Taiba (LeT) is a terrorist group primarily funded by the Pakistani intelligence agency, the Inter-Services Intelligence [137]. Created in 1990, the group has carried out numerous terrorist attacks, the most spectacular of which was the November 2008 terrorist attack in Mumbai that targeted several sites including the iconic Taj Mahal hotel, killing 166 innocent civilians (as well as nine terrorists, while a tenth terrorist was captured). LeT has strong links to various other terrorist groups including Al-Qaeda, Indian Mujahideen, Jaish-e-Mohammed, Jabhat-al-Nusra in Syria, groups in Chechnya, Jemaa Islamiyah, as well as organized crime groups such as Dawood Ibrahim's D-Company. For instance, Al-Qaeda leader Khalid Sheikh Mohammed was captured in an LeT safehouse in Pakistan. Given its technical sophistication and the support of a sophisticated intelligence agency, LeT is viewed as a major threat by both the US and India—both in terms of operations they might carry out themselves and in terms of training and logistics support they might provide to other groups that carry out such attacks.

In order to reduce terror attacks by LeT, we developed a five-player game. The players considered are the United States (US), India, Pakistan's government, Pakistan's military, and Lashkar-e-Taiba (LeT). We recall that Table 5.1, presented earlier, gives actions each player can take, and that—in addition to the actions below—each player can take the action `none`, which corresponds to doing nothing. We describe the other actions in depth here.

US Actions. The US can take three actions (and `none`).

1. The first is `covert` action against LeT. While we do not suggest specific operations, this action could be implemented in many ways including covert actions to undermine LeT's leaders or covert actions to target LeT training camps. It is clear that the US is capable of such covert action as evidenced by recent events involving a CIA contractor called Raymond Davis who was arrested by the Pakistanis after a shootout in Lahore.

2. The US could also `cut` military and/or development support currently being given to Pakistan. According to the Congressional Research Service, the US provided $1.727 billion in economic aid to Pakistan in FY2010.[2] In 2012, the US asked Congress for permission to ship almost $3 billion to Pakistan with over half being military aid.[3] Cutting some of this aid is an option the US has long considered, especially in view of US Admiral Mike Mullen's assertions in 2011 about Pakistan's ISI controlling the Haqqani terrorist network which in turn attacked the US embassy in Kabul.[4]

3. The US could also `expand` financial support for Pakistan. Pakistan's educational system and economy are both in shambles and some have argued that additional development assistance would wean young people away from radical elements.

India's Actions. As with the US, we study three actions (and `none`) that India might take. Similarly, there are many ways in which India could tactically implement these actions.

1. Like the US, India can also take different forms of `covert` action against LeT using methods similar to those listed above for the US.

[2]See "Pakistan-U.S. Relations: A Summary," by K. Alan Kronstadt of the Congressional Research Service, May 16, 2011.

[3]http://www.foxnews.com/topics/us-aid-to-pakistan-fy2012-request.htm

[4]http://www.nytimes.com/2011/09/23/world/asia/mullen-asserts-pakistani-role-in-attack-on-us-embassy.html?pagewanted=all&_r=0

2. India can also use `coercive diplomacy` in which diplomatic moves are used to coerce Pakistan. For instance, a credible threat can be used to warn Pakistan of the consequences of carrying out certain actions. For coercive diplomacy to be effective, the threat must be made publicly and must be credible [117]. Credible threats could include withholding water by diverting the headwaters of the Indus or by troop movements or simply by ramping up military spending which would place pressure on other parts of the Pakistani economy.

3. A third option we consider is one where India proposes some kind of `peace` initiative to Pakistan, e.g., granting some additional rights for back and forth movement between India and Pakistan, unifying families in Kashmir who were split up by the partition of Kashmir, and so forth.

Pakistan Military Actions. We study three possible actions for the Pakistani military, all related to their support for LeT.

1. The Pakistani military could implement a `crackdown` on LeT by arresting LeT members and/or closing down LeT's training camps, shutting down the logistical support for LeT operations in Jammu and Kashmir, and taking steps to interdict LeT-allied organizations like Jamaat-ud-Dawa. Pakistani security has, at times, cracked down on LeT, e.g., after the December 2001 parliament attack and the November 2008 attacks in Mumbai.

2. The Pakistani military could `cut` support to LeT by, e.g., arresting military officers who are illicitly supporting LeT and stopping military training of LeT personnel.

3. The Pakistani military could also `expand` support for LeT, e.g., by increasing its logistical and materiel support as well as financial support.

Pakistan Government Actions. We consider just two possible actions (in addition to `none`) by the civilian side of the Pakistani government (excluding the military side).

1. The Pakistani government could `prosecute` and arrest LeT personnel, as they have done periodically (though the leaders are usually released shortly thereafter).

2. The Pakistani government could choose to `endorse` LeT's social services program by routing government services through them. LeT runs many social services in Pakistan ranging from ambulances to hospitals, schools, and disaster relief programs.

Lashkar-e-Taiba's Actions. In the case of LeT, we considered three actions (in addition to the `none` action).

1. LeT could launch a major `attack`. We already know from the November 2008 Mumbai siege that they have the capability and logistical support to execute such attacks.

2. LeT could `hold` attacks (but not major ones), similar to those periodically carried out by them in Kashmir where military and civilian personnel are frequently targeted.

3. LeT could do something dramatic like eliminate its armed wing, give up its weapons, and publicly renounce violence. Though extremely unlikely, this is still worth listing as a possible action.

5.2.2 A Toy Example

We now introduce a small example that will be used to illustrate formal concepts and definitions as they are introduced later in the chapter. Consider a very simple game consisting of two players, a terrorist group T and a government G. Suppose the terrorist group can carry out two actions (`terror-attack` and `peace`) and the government can carry out two actions (`CT-ops` and `peace`). Here, `CT-actions` denotes some traditional counter-terror operations such as killing and arresting group members. Experts are divided on the values of these actions to each player and thus provide two payoff matrices, PM_1 and PM_2.

	`CT-ops`	`peace`
`terror-attack`	$(-5, -5)$	$(3, -10)$
`peace`	$(-8, 6)$	$(0, 0)$

PM_1

	`CT-ops`	`peace`
`terror-attack`	$(-5, 6)$	$(3, 3)$
`peace`	$(-5, -5)$	$(-5, 2)$

PM_2

Much of our analysis is for games with payoffs in $[0, 1]$. Note that a scaled and translated version of the above matrices that does not alter equilibria of the game can be constructed. The modified payoff matrices (rounded to hundredths) are given below.

	`CT-ops`	`peace`
`terror-attack`	$(0.31, 0.31)$	$(0.81, 0)$
`peace`	$(0.12, 1)$	$(0.62, 0.62)$

Scaled PM_1

	`CT-ops`	`peace`
`terror-attack`	$(0, 1)$	$(0.73, 0.73)$
`peace`	$(0, 0)$	$(0, 0.64)$

Scaled PM_2

In each of these tables, the rows show terror group T's actions and the columns show the government G's actions. For example, the entry $(3, -10)$ in PM_1 says that the payoff to the terror group is 3 and the payoff to the government is -10 when the terror group performs `terror-attack` and the government proposes `peace`.

We will use this simple motivating example to illustrate various concepts in this chapter.

5.3 Technical Preliminaries: Approximate Equilibria

In this section, we first review common game-theoretic models and equilibrium concepts (§5.3.1), then build on them to define approximate equilibria in games with multiple payoff functions (§5.4).

5.3.1 Approximate Equilibria in Games with a Single Payoff Function

We consider simultaneous multiplayer games. Let $[n] = \{1, 2, \ldots, n\}$ be the set of players and $[m] = \{1, 2, \ldots, m\}$ be the set of actions for each player. Let Δ_m be the simplex $\{(x_1, x_2, \ldots, x_m) | \sum_{i \in [m]} x_i = 1, x_i \geq 0, \forall i \in [m]\}$.

For any player j, any $\sigma^j \in \Delta_m$ is a probability distribution over the set of actions $[m]$; thus, σ^j is called a *strategy* for player j. If $\sigma^j = (x_1, x_2, \ldots, x_m)$, then x_i is the probability that player j will perform action i. When all but one of the x_i's in σ^j are 0, σ^j is called a *pure strategy*; otherwise, it is called a *mixed strategy*. In mixed strategies, a player probabilistically chooses which action to take. Note that we will calculate these mixed strategies from the multiple payoff matrices provided by experts. They are not inputs to our algorithms (and so experts do not have to provide them); they are outputs generated by our system.

We use Δ to denote the set $\Pi_{j=1}^{n} \Delta_m$. Any $\sigma \in \Delta$ is called a *strategy profile* for a game a. If $\sigma = (\sigma^1, \ldots, \sigma^n) \in \Delta$, then σ^j denotes the strategy of the player j. For convenience, we can represent a strategy profile σ as (σ^j, σ^{-j}), where σ^j represents the strategy of player j and σ^{-j} represents strategies for the rest of the players.

Example 6. *Consider the toy example given in Section 5.2.2. An example* pure strategy *for the government G is to play action* `CT-ops`. *Similarly, a pure strategy for the terror group T is to play action* `terror-attack`. *An example of a* mixed strategy *for G is to play action* `CT-ops` *and action* `peace` *with probabilities of $\frac{1}{3}$ and $\frac{2}{3}$, respectively. Similarly, T could play action* `terror-attack` *and action* `peace` *with probabilities $\frac{1}{2}$ and $\frac{1}{2}$, respectively. The above mixed stategies for G and T together form a* stategy profile *for the game.*

The *payoff* for a player j is a function $u_j : \Delta \mapsto [0, 1]$. In this section, we assume (without loss of generality) that all payoffs are in the unit interval $[0, 1]$. We now define a basic building block of game theory, the Nash equilibrium.

Definition 10. *A strategy profile* σ *is a* Nash equilibrium *iff:*

$$u_j(\sigma^{j\prime}, \sigma^{-j}) \leq u_j(\sigma) \quad \forall \sigma^{j\prime} \in \Delta_m, j \in [n]$$

Thus, a strategy profile is a Nash equilibrium if no player has incentive to deviate from his strategy, assuming all other players play their respective strategies. Classical game theory assumes that players are rational. Hence, players can reason about one another and identify the Nash equilibria that are possible and then typically play actions consistent with one such Nash equilibrium. As Schelling 1980 observes, a good amount of work may also be invested by players in "prepping" the game so that certain strategy profiles are excluded from being equilibria.

Example 7. *Consider the mixed strategy given in Example 6. G plays action* `CT-ops` *and action* `peace` *with probabilities* $\frac{1}{3}$ *and* $\frac{2}{3}$*, respectively. Similarly,* T *plays action* `terror-attack` *and action* `peace` *with probabilities* $\frac{1}{2}$ *and* $\frac{1}{2}$*, respectively. In this case, as per the payoff matrix* PM_1 *(defined in Section 5.2.2), the payoff for G is* $-5 * \frac{1}{3} * \frac{1}{2} + -10 * \frac{2}{3} * \frac{1}{2} + 6 * \frac{1}{3} * \frac{1}{2} + 0 * \frac{2}{3} * \frac{1}{2} = -\frac{19}{6}$*. We note that the payoff for a given strategy profile is the expected payoff given players draw actions independenty at random according to their respective strategies. A Nash equilibrium for the same game is for G to play* `CT-ops` *with probability* 1 *and for* T *to play* `terror-attack` *with probability* 1*, resulting in a payoff of* -5 *for both players.*

Since Nash equilibria are notoriously difficult to compute [27, 34], recent work has focused on finding *approximate* Nash equilibria. We use a well-known notion of an approximate Nash equilibria.

Definition 11. *A strategy profile* σ *is an* ϵ*-approximate Nash equilibrium for some* $0 \leq \epsilon \leq 1$ *iff:*

$$u_j(\sigma^{j\prime}, \sigma^{-j}) \leq u_j(\sigma) + \epsilon, \forall \sigma^{j\prime} \in \Delta_m, j \in [n]. \tag{5.1}$$

A stricter notion of an approximate Nash equilibrium is the well-supported approximate Nash equilibrium. Let $S(\sigma)$, the *support of* a strategy $\sigma \in \Delta_m$, be the set $S(\sigma) = \{i \mid \sigma_i > 0\}$. Intuitively, the support of σ is the set of actions that are executed with nonzero probability. Daskalakis, Goldberg, and Papadimitriou 2006 define a well-supported approximate Nash equilibrium as follows.

Definition 12. *Suppose $0 \leq \epsilon \leq 1$ is a real number. A strategy profile σ is a well-supported ϵ-approximate Nash equilibrium iff:*

$$u_j(e_i, \sigma^{-j}) \leq u_j(e_l, \sigma^{-j}) + \epsilon \quad \forall \sigma^{j\prime} \in \Delta_m, i \in [m], \\ l \in S(\sigma^j), j \in [n]$$

In other words, for a strategy to be a well-supported ϵ-approximate Nash equilibrium, every player's incentive to deviate from his equilibrium strategy is very small (less than a utility of ϵ).

Definition 11 and Definition 12 both define approximate Nash equilibria that are additive in nature (due to the $+\epsilon$ term in the right side of the definition). A multiplicative (relative) approximation can be defined as follows, due to [34].

Definition 13. *A strategy profile σ is a* well-supported relative ϵ-approximate Nash equilibrium *for $0 \leq \epsilon \leq 1$ iff $\forall j \in [n], i \in [m]$:*

$$(1-\epsilon)u_j(e_i, \sigma^{-j}) \leq u_j(e_l, \sigma^{-j}), \forall l \in S(\sigma^j) \tag{5.2}$$

The following example illustrates these different notions of approximate equilibria.

Example 8. *Consider the strategy profile from Example 6. G plays action* `CT-ops` *and action* `peace` *with probabilities $\frac{1}{3}$ and $\frac{2}{3}$, respectively. Similarly, T plays action* `terror-attack` *and action* `peace` *with probabilities $\frac{1}{2}$ and $\frac{1}{2}$, respectively. For T, this is an ϵ-approximate Nash equilibrium with $\epsilon = 1.5$ since the expected payoff for T*

is -1.17 *and deviation to action* `terror-attack` *leads to a payoff of* 0.33*. This is a well-supported* ϵ*-approximate Nash equilibrium for* T *with* $\epsilon = 3$ *because the payoff for action* `peace`*, which is in the support of* T*'s strategy, is* -2.67 *and deviation to* `terror-attack` *leads to a payoff of* 0.33*—a gain of* 3.00*.*

5.4 Approximate Equilibria in Games with Multiple Payoff Functions

In this section, we merge together the ideas of (well-supported) approximate Nash equilibria and Shapley's vector payoffs in an effort to combine multiple conflicting experts' knowledge of payoffs.

Definition 14. *A* simultaneous game with multiple payoff functions *(SGM) is a triple* $G = (n, m, U)$ *where* $[n]$ *is a set of players,* $[m]$ *is the set of actions for each player in* $[n]$*, and* $U = (U_1, U_2, \ldots, U_f)$ *consists of* f *ordered sets of payoff functions* $U_k = (u_1^k, u_2^k, \ldots, u_n^k), \forall k \in [f]$.

Intuitively, an SGM G can be viewed as f different games specified over a set of players, over the same strategy space, with payoff functions for players given by $U_k, k \in [f]$. We refer to these f individual simultaneous games as *constituent games* of G. *Throughout this chapter, we use the variable* f *to denote the number of payoff matrices considered—which is also equal to the number of constituent games in a SGM or a ZSGM (a zero-sum version of an SGM defined later in Section 5.5.1).*

For instance, in our toy example, the game G consists of two different constituent games, one corresponding to each of the two payoff matrices.

We now merge the idea of an approximate Nash equilibrium (Definition 11) with that of Shapley's vector payoffs.

Definition 15. *A strategy profile* σ *is a* multiple ϵ-approximate Nash equilibrium *of an SGM* (n, m, U)*, iff it is an* ϵ*-approximate Nash equilibrium for each of its constituent* *games. Specifically, for all* $k \in [f]$*:*

$$u_j^k(\sigma^{j\prime}, \sigma^{-j}) \leq u_j^k(\sigma) + \epsilon,\ \forall \sigma^{j\prime} \in \Delta_m, j \in [n] \tag{5.3}$$

Building on Definition 15, we also combine well-supported approximate Nash equilibria (Definition 12) with vector payoffs.

Definition 16. *A strategy profile* σ *is a* well-supported multiple ϵ-approximate Nash equilibrium *of an SGM* (n, m, U)*, iff it is a well-supported* ϵ*-approximate Nash equilibrium for each of its constituent games. That is, for all* $k \in [f]$*:*

$$\begin{aligned} u_j^k(e_i, \sigma^{-j}) \leq u_j^k(e_l, \sigma^{-j}) + \epsilon \quad & \forall \sigma^{j\prime} \in \Delta_m, i \in [m], \\ & l \in S(\sigma^j), j \in [n] \end{aligned}$$

Finally, we can define the multiplicative version of Definition 16 as well.

Definition 17. *A strategy profile* σ *is a* well-supported multiple relative ϵ-approximate Nash equilibrium *of an SGM* (n, m, U) *iff it is a well-supported relative* ϵ*-approximate Nash equilibrium for each of its constituent games. Thus, for all* $k \in [f], j \in [n], i \in [m]$*:*

$$(1 - \epsilon) u_j^k(e_i, \sigma^{-j}) \leq u_j^k(e_l, \sigma^{-j}),\ \forall l \in S(\sigma^j) \tag{5.4}$$

We now provide an example of a well-supported multiple ϵ-approximate Nash equilibrium in the context of our toy game.

Example 9. *For the game defined in Section 5.2.2, consider the strategy profile where* T *plays action* `terror-attack` *with probability* 1 *and* G *plays action* `CT-ops` *with probability* 1*. For both the payoff matrices,* PM_1 *and* PM_2*, this is a Nash equilibrium. Therefore, the given staregy profile is a well-supported multiple* ϵ*-approximate Nash equilibrium with* $\epsilon = 0$*. Consider another strategy profile, where* G *plays action*

`CT-ops` and action `peace` with probabilities $\frac{1}{3}$ and $\frac{2}{3}$, respectively. Similarly, T plays action `terror-attack` and action `peace` with probabilities $\frac{1}{2}$ and $\frac{1}{2}$, respectively. This is a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = 5.5$. This is because in one of the payoff matrices, PM_1, the payoff G receives from action `peace` in support is -5 and deviating to action `CT-ops` leads to a payoff of 0.5 leading to a gain of $\epsilon = 5.5$.

A well-supported multiple ϵ-approximate Nash equilibrium is "close" in payoff for each player to a (Nash or approximate Nash) equilibrium in the constituent game corresponding to each payoff matrix in the SGM. A well-supported multiple ϵ-approximate Nash equilibrium closely approximates equilibrium situations irrespective of which of the several experts' payoff matrices is used—it is a robust.

For notational convenience, in the experimental section of this chapter, we will refer to well-supported multiple ϵ-approximate Nash equilibria computed using only $U' \subseteq U$ payoff functions as (ϵ, k)*-equilibria*, where $|U'| = k$. Such equilibria computed with the full set U are simply written as ϵ*-equilibria*.

5.5 Approximate Equilibria in Simultaneous Games with Multiple Payoff Functions

We begin by analyzing two fairly constrained cases, zero-sum games (§5.5.1) and rank 1 games (§5.5.2). We then relax these assumptions, providing results for low-rank games (§5.5.3), which will later lead into results on general games where the number of players is constant (§5.6.2).

5.5.1 Zero-sum Games with Multiple Payoffs

We begin by extending the well-known linear program (LP) for computation of an exact Nash equilibrium in zero-sum games to the computation of an *approximate* Nash equilibrium, and subsequently use it to design an algorithm to compute multiple payoff equilibria in such games. We will focus on the zero-sum equivalent of a simultaneous game with multiple payoff functions, as defined by Definition 18.

Definition 18. *A* zero-sum simultaneous game with multiple payoff functions (for two players) *(ZSGM) is an SGM $(2, m, U)$, with ordered set of payoff functions $U = (u^1, u^2, \dots, u^f)$ such that u^k ($-u^k$) is the payoff function for player 1 (player 2) $\forall k \in [f]$. For convenience, we denote such games $G = (m, U)$.*

Note that ZSGMs are limited to just two players.

Let (m, U), where $U = (u^1, u^2, .., u^f)$, be a ZSGM. Let $P = (r_1, r_2, \dots, r_f)$, $r_i \in [0, 1], \forall i \in [f]$. Consider the following LP:

$$LP^f(U, P, \epsilon)$$

$$\sum_{i \in [m]} \sigma_i^1 = 1$$

$$\sigma_i^1 \geq 0, \forall i \in [m]$$

$$\sum_{i \in [m]} \sigma_i^2 = 1$$

$$\sigma_i^2 \geq 0, \forall i \in [m]$$

$$\sum_{i \in [m]} \sigma_i^1 u^k(e_i, e_j) \geq r_k - \epsilon, \forall j \in [m], k \in [f] \tag{5.5}$$

$$\sum_{i \in [m]} \sigma_i^2 u^k(e_i, e_j) \leq r_k + \epsilon, \forall j \in [m], k \in [f] \tag{5.6}$$

Here, the first four equations are required because σ^1 and σ^2 are distributions over actions of players 1 and 2 respectively. Equations (5.5) and (5.6) are required because we want players to play strategies that are approximate best responses to each of the constituent games of the ZSGM.

For $f = 1$, this LP applies to a zero-sum game with scalar payoff function $u = u^1$. For this special case, we call this LP, $LP^1(u, r, \epsilon)$. Lemma 40 and Lemma 41 show that for the single payoff function case, the linear program LP^1 computes approximate Nash equilibria for zero-sum games. Thus, our framework neatly extends approximate Nash equilibria to the case when there are vector-valued payoffs. The next result states that any solution to the linear program given above yields a Nash equilibrium.

Lemma 40. *Any feasible solution to $LP^1(u, r, \epsilon)$ is a 2ϵ-approximate Nash Equilibrium for a zero-sum game with u as payoff function for player 1.*

The following result says that every ϵ-approximate Nash equilibrium is a solution of the linear program LP given above.

Lemma 41. *Any ϵ-approximate Nash equilibrium strategy profile (σ^1, σ^2) for zero-sum game with payoff function u for player 1 such that payoff for player 1 is in $[r - \epsilon, r + \epsilon], \epsilon \geq 0$ is a feasible solution to $LP^1(u, r, 2\epsilon)$.*

Lemma 42 and Lemma 43 below extend the above results (which apply when $f = 1$, i.e., when there is only one payoff function) to the case of zero-sum games with multiple payoff functions. The first result, analogous to Lemma 40, states that solutions of the above LP are multiple payoff ϵ-approximate equilibria.

Lemma 42. *Any feasible solution to $LP^f(U, P, \frac{\epsilon}{2})$ is a multiple payoff ϵ-approximate equilibrium for the ZSGM (m, U).*

The next result, analogous to Lemma 41 states that for every multiple payoff ϵ-approximate equilibrium, there is a corresponding solution of the above LP.

Lemma 43. *Let $\sigma = (\sigma^1, \sigma^2)$ be a strategy profile that is a multiple payoff ϵ-approximate equilibrium for the ZSGM (m, U). Let $P = (r_1, r_2, \ldots, r_f)$ be the vector of payoffs for player 1 for each of the constituent games of the ZSGM. Let $P' = (r'_1, r'_2, \ldots, r'_f)$ be a vector such that $|r_i - r'_i| \leq \epsilon, \forall i \in [f]$. Then σ is a feasible solution to $LP^f(U, P, 2\epsilon)$.*

Thus, $LP^f(U, P, \epsilon)$ precisely captures the entire set of multiple payoff ϵ-approximate equilibria of our zero sum game.

Algorithm 5 presents a method to compute the set of all approximate ϵ-equilibria in the multiple payoff case. The algorithm uses an input k in order to regulate the approximation error factor, ϵ.

Algorithm 5 Approximate Multiple Payoff ϵ-Equilibrium in Zero-sum Games

Input: k, set of payoff functions $U = (u^1, u^2, \ldots, u^f)$
Output: A set of LPs (see Theorem 44 for details)

```
S ← {0/k, 1/k, ..., 1}
Payoffs ← X_{i=1}^{f} S                    ▷ Cardinality: (k+1)^f
LP_Set ← ∅
for P ∈ Payoffs do
    LP_Set ← LP_Set ∪ LP^f(U, P, 1/k)
end for
return LP_Set
```

The result below shows that Algorithm 5 computes certain types of multiple-payoff approximate equilibria.

Theorem 44. *Algorithm 5 runs in time $O((k+1)^f(2mf+2m+2))$ and outputs a set of LPs. Let S be the union of feasible regions of all LPs in the set returned by the algorithm. Then S satisfies the following conditions:*

1. *All strategy profiles in S are approximate multiple payoff ϵ-equilibria with $\epsilon = \frac{2}{k}$.*

2. *All multiple payoff ϵ-equilibria with $\epsilon = \frac{1}{2k}$ are in S.*

5.5.2 Multiplayer Games of Rank 1

We now deal with the problem of finding equilibria in low-rank multiplayer games with multiple payoffs. Our real-world LeT application is one example of a low rank multiplayer game.

The definition of rank that we use is equivalent to one given by Kalyanaraman and Umans 2007. As is evident from recent papers (e.g., [2, 71, 73, 87, 126]), games of low rank have generated considerable interest.

In this section, we first define *multiplayer games of rank K*. We then give a complete characterization of Nash equilibria for these games when $K = 1$ and use this characterization to compute *well-supported relative ϵ-approximate Nash equilibria*.

Definition 19. *A* multiplayer game of rank K *is a game where the payoff function for each player is specified by K n-tuples of vectors, each of length m. Let $\alpha^{j,k} = (\alpha^{1,j,k}, \alpha^{2,j,k}, \ldots, \alpha^{n,j,k})$ be the tuple specifying the payoff function for player j. Let $\rho = (e_{a_1}, e_{a_2}, .., e_{a_n})$ be a strategy profile with only pure strategies for each player, where a_i is the action for player i. Then, the payoff for player j is defined as:*

$$u_j(e_{a_1}, e_{a_2}, .., e_{a_n}) = \sum_{k \in [K]} \prod_{i \in [n]} \alpha^{i,j,k}_{a_i} \tag{5.7}$$

For a strategy profile, $\sigma = (\sigma^1, \sigma^2, .., \sigma^n)$, payoffs are defined as usual. Let $A = \underbrace{[m] \times [m] \times .. \times [m]}_{n \text{ times}}$ be the set of all possible combinations of actions of all players. Then, payoffs are given by:

$$u_j(\sigma) = \sum_{a \in A} \prod_{i \in [n]} \sigma^i_{a_i} u_j(e_{a_1}, e_{a_2}, .., e_{a_n}) \tag{5.8}$$

$$= \sum_{a \in A} \prod_{i \in [n]} \sigma^i_{a_i} \sum_{k \in [K]} \prod_{l \in [n]} \alpha^{l,j,k}_{a_l} \tag{5.9}$$

We note that each payoff matrix is of rank at most K and it is input as a rank-K decomposition. As can be seen from Equation 5.7, the payoff matrix for player j is a sum of K terms and the k^{th} term is the tensor product of vectors in tuple $\alpha^{j,k}$. This is a complicated definition, so let us consider the case when $K = 2$. In this case, the payoff matrix is implicitly specified by two vectors, each of length m (the number of actions). Consider the strategy profile $\rho = (e_{a_1}, e_{a_2}, .., e_{a_n})$. This strategy profile tells us what actions each of the n players is taking. From Equation 5.9, in a rank $K = 2$

game, we compute the payoff for player 1 (i.e. $j = 1$) as

$$\Pi_{i=1}^{n} \alpha_{a_i}^{i,1,1} + \Pi_{i=1}^{n} \alpha_{a_i}^{i,1,2}.$$

The following example uses our toy example to illustrate rank K games.

Example 10. *As an example, we give a rank-2 decomposition of payoff matrix* PM_1 *of player T given in Section 5.2.2. A rank-2 decomposition of the matrix is* $\alpha^{1,1,1} = \{-1.76, -2.54\}, \alpha^{2,1,1} = \{3.05, -0.55\}, \alpha^{1,1,2} = \{-1.30, 0.90\}, \alpha^{2,1,2} = \{-0.28, -1.56\}$. *It can be easily verified (up to rounding error) that for T,* $\mathrm{PM}_1 = \alpha^{1,1,1}(\alpha^{2,1,1})^T + \alpha^{1,1,2}(\alpha^{2,1,2})^T$. *A similar decomposition of* PM_1 *for G is given by* $\alpha^{1,2,1} = \{-3.25, 0.99\}, \alpha^{2,2,1} = \{1.92, 2.81\}, \alpha^{1,2,2} = \{0.67, 2.18\}, \alpha^{2,2,2} = \{1.88, -1.28\}$. *Therefore, the game specified by* PM_1 *is a rank-2 game.*

For the special case of rank 1 games, we drop the superscript k from vectors $\alpha^{i,j,k}$.

Nash Equilibria for Rank 1 Games

The following result presents a complete characterization of Nash equilibria for multi-player games of rank 1.

Lemma 45. *Let σ be a mixed strategy profile. Let $u'_{-j}(\sigma) = \prod_{i \in [n]-\{j\}} (\sum_{l \in [m]} \sigma_l^i \alpha_l^{i,j})$. Let the support of player j's strategy be $S_j = \{l | \alpha_l^{j,j} = \max(\alpha^{j,j})\}$. σ is a Nash equilibrium iff:*

$$u'_{-j}(\sigma) > 0 \implies support(\sigma^j) \subseteq S_j$$

In the rest of this section, we assume that all players have a non-zero payoff at equilibrium, since, for any general multiplayer game, if a player has a zero payoff at equilibrium, which is the minimum possible payoff for the game, then her maximum and minimum possible payoffs are both zero for any choice of action and the player is free to take any action in her equilibrium strategy.

ϵ-Approximate Multiple Payoff Equilibria for Rank 1 Games

We now give a characterization of *well-supported relative ϵ-approximate Nash equilibria*. Here we solve the multiplicative approximation problem which is harder than the additive approximation for normal form games [33].

Lemma 46. *A strategy profile $\sigma = \langle \sigma^1, \sigma^2, \ldots, \sigma^n \rangle$ is a well-supported relative ϵ-approximate Nash equilibrium (with a non-zero payoff for all players) for a multiplayer game of rank 1 with payoffs as specified in Section 5.5.2 iff:*

$$\alpha_i^{j,j} \geq (1-\epsilon)(\max \alpha^{j,j}), \forall j \in [n], i \in S(\sigma^j) \tag{5.10}$$

We now prove the main result for this section and give an algorithm for the computation of well-supported multple relative ϵ-approximate Nash equilibria in multiplayer games of rank 1. The following theorem is the main result for this section.

Theorem 47. *Consider an SGM of rank 1 with f different payoff functions for each player. For $t \in [f]$, let payoff function k for player j be specified by tuple $(\alpha^{1,j,k}, \alpha^{2,j,k}, \ldots, \alpha^{n,j,k})$. Then, a strategy profile $\sigma = (\sigma^1, \sigma^2, \ldots, \sigma^n)$ is a well-supported multiple relative ϵ-approximate Nash equilibrium, iff $\forall j \in [n]$:*

$$\alpha_i^{j,j,k} \geq (\max \alpha^{j,j,k})(1-\epsilon), i \in S(\sigma^j), k \in [f] \tag{5.11}$$

Algorithm 6 leverages this result to compute well-supported multiple relative ϵ-approximate Nash equilibria for rank-1 games.

We use a variant of our toy example to explain Algorithm 6.

Example 11. *As an example, consider a rank-1 game where Player 1 is T and Player 2 is G. Let a rank-1* PM_1 *for T be $\alpha^{1,1,1} = \{1, 0.5\}, \alpha^{2,1,1} = \{0.5, 1\}$. Let, a similar* PM_1 *for G be given by $\alpha^{1,2,1} = \{1, 0.25\}, \alpha^{2,2,1} = \{1, 0.25\}$. Let* PM_2 *for T be given by $\alpha^{1,1,2} = \{0.75, 1\}, \alpha^{2,1,1} = \{0.75, 1\}$. Let* PM_2 *for G be given by*

Algorithm 6 well-supported multple relative ϵ-approximate Nash equilibria in rank-1 games

Input:ϵ, payoff vectors $\alpha^{i,j,t}$
Output: Allowed actions in supports of a feasible strategy profile

for $i \in [m], j \in [n]$ **do**
  if $\forall t \in [f], \alpha_i^{j,j,t} \geq (\max \alpha^{j,j,t})(1-\epsilon)$ **then** Add i to actions in support of player j's strategies.
  end if
end for
if there exists any player with empty support set **then** ▷ If any support is empty the profile is infeasible
  return NULL
else
  return The support sets constructed in the for loop
end if

$\alpha^{1,2,2} = \{0.75, 1\}, \alpha^{2,2,2} = \{1, 0.75\}$. *Let* $\epsilon = 0.5$. *Then for* PM_1, *all mixed strategies for* T *are well-supported relative* ϵ*-approximate Nash equilibria. For* G *only action* `CT-ops` *can be in the support. For* PM_2, *all mixed strategies for both the players are well-supported relative* ϵ*-approximate Nash equilibria. Therefore, a strategy profile where* T *plays some mixed strategy and* G *plays action* `CT-ops` *is a well-supported multiple relative* ϵ*-approximate Nash equilibrium with* $\epsilon = 0.5$.

5.5.3 Multiple Payoff Games of Low Rank

In this section we consider the general case of *multiplayer games of low rank* (Definition 19). We prove that a class of strategies called "uniform strategies" (which we will define shortly) can be used to compute approximate Nash equilibria for these games when the number of actions is small. We then leverage this result to design an algorithm that computes the set of all multiple payoff equilibria for such games.

In this and the next section, we focus only on uniform strategies. Uniform strategies provide a tradeoff between simplicity and optimality that may be valuable to the end user. For example, in the LeT game we study later in the chapter, a policy prescription like "India should take covert action against LeT with probability 0.0071" may not be very useful to the end user. A simpler policy prescription that is *almost* as good may be a much better option. We now define a uniform strategy profile.

Definition 20. *A strategy profile* $\sigma = (\sigma^1, \sigma^2, \ldots, \sigma^n)$ *is a* t*-uniform strategy profile if,* $\forall i \in [n], \forall j \in [m]$*:*

$$\sigma^i_j \in \{\frac{\ell}{t} \mid \ell \in \{0, 1, \ldots, t\}\}$$

Intuitively, a t-uniform strategy discretizes the $[0, 1]$ real-valued interval into t segments and considers two probabilities within the same segment to effectively be the same. Thus, as t gets bigger, we get finer granularity. Thus, our selection of t controls the *granularity* of the probability distribution on actions in a strategy. A smaller t leads to a coarse-grained, simple strategy whereas a larger t allows a more fine grained strategy that may be closer to an optimal strategy.

Approximate Nash Equilibria in Games of Low Rank

In this subsection, we constructively prove that a uniform strategy profile can be used to approximate a Nash equilibrium for multiplayer games of low rank. First, we state the following lemmas to help with the main result.

Lemma 48. *Let* α *be a vector of length* m *such that each element of* α *is in* $[0, 1]$*. Let* σ *be vector of length* m*. Let* σ' *be a vector such that* $|\sigma_i - \sigma'_i| \leq \epsilon, \forall i \in [m]$*. Then* $|\alpha^T \sigma - \alpha^T \sigma'| \leq m\epsilon$*.*

Lemma 49. *Let* $x_1, \ldots, x_n$ *be n reals such that* $0 \leq x_i \leq 1, \forall i \in [n]$*. Let* $x'_1, \ldots, x'_n$ *be n reals such that* $0 \leq x'_i \leq 1, |x_i - x'_i| \leq \epsilon, \forall i \in [n]$*. Then* $|\prod_{i \in [n]} x_i - \prod_{i \in [n]} x'_i| \leq n\epsilon$*.*

The main technical result of this subsection is that if a strategy profile is a well-supported multiple ϵ-approximate Nash equilibrium, then there exists a t-uniform strategy profile that is also a well-supported multiple ϵ-approximate Nash equilibrium with a slightly higher value of ϵ. However, the simpler lemma below—dealing with the single payoff case—provides the basis for the more complex theorem to follow.

Lemma 50. *Let the strategy profile $\sigma = (\sigma^1, \sigma^2, .., \sigma^n)$ be a well-supported ϵ-approximate Nash equilibrium for the given game of rank k. Then there exists a t-uniform strategy profile σ' that is a well-supported $\epsilon + \frac{2(n-1)mk}{t}$-approximate Nash equilibrium.*

We now extend Lemma 50 to the multiple payoff case pertaining to well-supported multiple ϵ-approximate Nash equilibria.

Theorem 51. *Let the strategy profile σ be a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau$ for the given SGM, all of whose constituent games are rank k games. Then, there exists a t-uniform strategy profile σ' that is a well-supported multiple ϵ-approximate Nash equilibrium, with $\epsilon = \tau + \frac{2(n-1)mk}{t}$.*

5.6 Computing Multiple Payoff Approximate Equilibria

Building on the theoretical results of the last section, we now provide an efficient algorithm for computing well-supported multiple ϵ-approximate Nash equilibria in games where the number of players is constant. First, we present a grid search algorithm for computing equilibria (§5.6.1), and show that is efficient through a quasi-polynomial time approximation scheme (QPTAS) (§5.6.2). This algorithm is validated on simulated data in Section 5.10, and on real data in the next section (§5.7).

5.6.1 Algorithm for Computation of Equilibria

We leverage Theorem 51 to present Algorithm 7 which searches over the space of all uniform strategy profiles and outputs those that are well-supported multiple ϵ-approximate Nash equilibria. Input parameters to the algorithm are t and payoff functions for the constituent games. We assume that each payoff function is given as an oracle, which, given the strategy profiles, returns a vector with payoffs for all players. The output of the algorithm is the set of all t-uniform strategy profiles which are well-supported multiple ϵ-approximate Nash equilibria for the given SGM.

The algorithm first chooses a strategy profile and checks if it is an equilibrium (e.g., by solving the linear programs presented earlier in the chapter). For each payoff function in the list of f payoff functions, it then iteratively looks at pairs of players, trying to set payoffs that are sufficiently close to each other in an attempt to find an equilibrium. It iteratively adds any valid solutions found to the solution and returns the solution at the end. Via Theorem 51, this coarse grid search is guaranteed to find a "reasonable" overall equilibrium (with respect to the parameter t).

The following example illustrates Algorithm 7 on our running toy example.

Example 12. *We note that a well-supported multiple ϵ-approximate Nash equilibrium exists for the game with $\epsilon = 0$. This is the common equilibrium for both payoff matrices when T plays action `terror-attack` and G plays action `CT-ops`. Thus, to guarantee $\epsilon = 0.2$, we require $t = 40$ for this game. For illustrative purposes, to avoid enumeration of all strategies required for $\epsilon = 0.2$, we use the algorithm as follows. We enumerate all t-uniform strategies and report only those strategies that are well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = 0.2$.*

We note that strategy profiles given in rows 1 through 4 of the above table are common Nash equilibria with $\epsilon = 0$. All the other 3-uniform profiles have $\epsilon > 0.2$. Thus strategy profiles in the first 4 rows are all the 3-uniform well-supported multiple ϵ-approximate Nash equilibria of the given game.

	T strategy		G strategy		ϵ for T		ϵ for G		Max ϵ
	`terror-attack`	`peace`	`CT-ops`	`peace`	PM_1	PM_2	PM_1	PM_2	
1	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
2	1.00	0.00	0.67	0.33	0.00	0.00	0.00	0.00	0.00
3	1.00	0.00	0.33	1.00	0.00	0.00	0.00	0.00	0.00
4	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
5	0.67	0.33	1.00	0.00	0.19	0.33	0.00	0.03	0.33
6	0.67	0.33	0.67	0.33	0.19	0.33	0.24	0.03	0.33
7	0.67	0.33	0.33	1.00	0.25	0.33	0.73	0.03	0.73
8	0.67	0.33	0.00	1.00	0.19	0.33	0.73	0.03	0.73
9	0.33	1.00	1.00	0.00	0.19	0.48	0.00	0.55	0.55
10	0.33	1.00	0.67	0.33	0.19	0.48	0.24	0.55	0.55
11	0.33	1.00	0.33	1.00	0.25	0.48	0.73	0.55	0.73
12	0.33	1.00	0.00	1.00	0.19	0.48	0.73	0.55	0.73
13	0.00	1.00	1.00	0.00	0.19	0.38	0.00	0.00	0.38
14	0.00	1.00	0.67	0.33	0.19	0.38	0.24	0.00	0.38
15	0.00	1.00	0.33	1.00	0.25	0.38	0.73	0.00	0.73
16	0.00	1.00	0.00	1.00	0.19	0.38	0.73	0.00	0.73

Though this algorithm can be expected to yield reasonable running times for games of any rank, the guarantees shown in Theorem 51 only apply to low rank games. This algorithm has the added advantage that we do not need to compute the tensor decomposition of the game matrix. As we will show in Section 5.6.2, uniform strategies are expected to provide good results on general games too.

5.6.2 A General Approximation Lemma for SGMs

We prove the existence of a QPTAS for SGMs when the number of players is constant. For this we first state and prove an approximation lemma (an extension to SGMs of Althöfer's Approximation Lemma 1994). Our approximation lemma states that if a well-supported multiple ϵ-approximate Nash equilibrium exists for an n-player game, then there is a well-supported multiple ϵ-approximate Nash equilibrium for the game using a t-uniform strategy with a slightly larger ϵ.

We note that the original version of Althöfer's Approximation Lemma applies only to two-player bimatrix games and its straightforward application leads to a QPTAS for computing well-supported ϵ-approximate Nash equilibria for two-player games.

However, our extension of the lemma to multiplayer games with multiple payoffs is not straightforward. In the multiplayer setting, the variables we consider are mutually overlapping products of independent random variables. Hence, to apply Hoeffding's bound 1963, we iteratively discretize strategies of players.

Lemma 52. *Consider a game with players in set $[n]$ and each player with actions in set $[m]$. Let $\sigma = (\sigma^1, \ldots, \sigma^n)$ be any well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau$ for the game. Then, for any $j \in [n]$, there exists a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau + \delta$ in which the strategy of player j is t-uniform for $t = \frac{2\log(2fmn)}{\delta^2}$.*

We now prove the main result of this section, Theorem 53. This result is stronger than Theorem 51 for the general case (applied directly to the low rank case) due to its logarithmic dependendence on m and f.

Theorem 53. *Let the strategy profile σ be a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau$ for the given SGM. Then, there exists a t-uniform strategy profile that is a well-supported multiple ϵ-approximate Nash equilibrium, with $\epsilon = \tau + \delta$ where $t = \frac{2n^2 \log(2fm(n-1))}{\delta^2}$.*

Thus, in Algorithm 7, if we set $t = \frac{2n^2 \log(2fm(n-1))}{\delta^2}$, we are sure to find a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau + \delta$ given that at least one well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau$ exists. The runtime is then $O(m^{nt})$. Thus, the same algorithm works for both general and low rank case, albeit with different performance guarantees.

5.7 Policy Analysis Results

We developed the Policy Recommendation Engine based on Vector Equilibria (PREVE) using the equilibrium concepts and algorithms described earlier. Using PREVE, we were able to analyze the Lashkar-e-Taiba (LeT) application described in Section 5.2.

Table 5.2: Statistics on number of $(\epsilon, 2)$ equilibria found.

k	ϵ	#Eq. found	#Eq. without LeT attacks
2	0	252	6
2	0.1	357	6
2	0.2	1696	9
2	0.3	13925	42

We first obtained payoff matrices from three experts in the politics of South Asia and LeT in particular; to avoid bias, none had any background in game theory and none had ethnic origins in the Indian subcontinent. Two were retired US State Department employees with over 30 years of knowledge of negotiations in the region. The third was the author of two well-known books on terrorism. The payoff matrices were created completely independently using *open source* information as well as expertise of these experts by following a set of instructions on what payoff values meant.

As described earlier, for notational convenience, we will refer to well-supported multiple ϵ-approximate Nash equilibria computed using only $U' \subseteq U$ payoff functions as (ϵ, k)-*equilibria*, where $|U'| = k$. Such equilibria computed with the full set U are simply written as ϵ-*equilibria*.

Before presenting the policy implications of the results generated by PREVE, we present a summary of the (ϵ, k)-equilibria we found in Table 5.2. We limit the equilibria presented to those where LeT does not attack. *No such* $(\epsilon, 3)$-*equilibria were found for* $\epsilon \leq 0.5$, *so we focus on the case when* $k = 2$. In the case of mixed equilibria, we list an equilibrium as having no LeT attacks when the probability of LeT attacking (action `attack`) or holding its current set of attacks (action `hold`) is 25% or less.

We found no $(0, 3)$-equilibria where LeT did not perform violent actions, but we did find the following:

1. There were 20 $(0, 2)$-equilibria in which experts #1 and #3 agreed, 218 $(0, 2)$-equilibria with experts #2 and #3 agreeing, and 14 $(0, 2)$-equilibria in which experts #1 and #2 agreed.

Table 5.3: All (ϵ, k) equilibria with $\epsilon = 0, k = 2$ in which LeT does not attack.

Equil.	LeT	PakG	PakM	India	US
$E^1_{0,1,3}$	`eaw`	`pros`	`crack`	`covert`	`cut`
$E^2_{0,1,3}$	`eaw`	`pros`	`crack`	0.75: `covert` 0.25: `coerce`	`cut`
$E^3_{0,1,3}$	`eaw`	`none`	`crack`	`coerce`	`cut`
$E^4_{0,1,3}$	`none`	`pros`	`support`	`covert`	`cut`
$E^5_{0,2,3}$	`eaw`	`pros`	`crack`	`coerce`	`cut`
$E^6_{0,2,3}$	`none`	`none`	`crack`	`covert`	`cut`

2. Of these 252 $(0, 2)$-equilibria, there were just six in which LeT did not carry out attacks. There were no $(0, 2)$-equilibria involving experts #1 and #2 in which LeT did not carry out attacks. Table 5.3 below summarizes the actions present in these six situations. An equilibrium named $E_{\epsilon,j,j'}$ is used to denote an $(\epsilon, 2)$ equilibrium in which the two experts who "agree" are j and j'.

   In all six $(0, 2)$-equilibria listed above where LeT stands down, the US cuts aid (development and military) to Pakistan, and India either carries out covert action against LeT or engages in coercive diplomacy. Moreover, in most $(0, 2)$-equilibria, the Pakistani military must crack down on LeT (though there is one case where they may expand support) and additionally, the Pakistani government must mostly prosecute LeT leaders (though there are two cases where they could do absolutely nothing). When we look at experts #2 and #3, we see that there are only two $(0, 2)$-equilibria in which LeT does not attack—in one India takes covert action and the US cuts aid. In both scenarios, the Pakistani military cracks down on LeT—in one the Pakistani government prosecutes LeT personnel and does nothing in the other.

   When we do the same with experts #1 and #3, we see that there are four $(0, 2)$-equilibria in which LeT does not attack. In all four, India takes either covert action or applies coercive diplomacy and the US cuts aid. In three cases, LeT eliminates its armed wing, while in another it does nothing. In the other two, LeT

Table 5.4: All (ϵ, k) equilibria with $\epsilon = 0.1, k = 2$ which are not $(0, 2)$-equilibria, and in which LeT does not attack.

Equil.	LeT	PakG	PakM	India	US
$E^7_{0.1,1,3}$	0.5: `attack` 0.5: `none`	`pros`	`expand`	`covert`	`cut`
$E^8_{0.1,1,3}$	0.25: `attack` 0.75: `none`	`pros`	`expand`	`covert`	`cut`
$E^9_{0.1,1,3}$	`none`	`pros`	`expand`	`covert`	`cut`

has a 50% (resp. 75%) chance of doing nothing and a 50% (resp. 25%) chance of attacking. In three cases, the Pakistani government prosecutes LeT personnel and does nothing in the fourth. In three of the cases, the Pakistani military cracks down on LeT, and in the one remaining case, it actually expands support for LeT. What these results may suggest is that *India should expand covert action against LeT with the US cutting financial aid to Pakistan at the same time* if the goal is to reduce violence by LeT.

3. We also looked at $(0.1, 2)$-equilibria (i.e., where $\epsilon = 0.1$), which means that each player may lose up to 10% of their best utility while being near an equilibrium with 2 of the 3 experts in agreement. In this case, we see no $(0.1, 2)$-equilibria involving experts #1 and #2 where LeT does not attack. But with experts #1 and #3, and experts #2 and #3, we do see such equilibria. As all $(0, 2)$-equilibria continue to be $(0.1, 2)$-equilibria, we only show new $(0.1, 2)$-equilibria in Table 5.4.

   With $\epsilon = 0.1$, we only get three new equilibria as compared to Table 5.3. In all of these, the US needs to cut aid to Pakistan and India needs to carry out covert action against LeT. As in the previous table, this requires that the Pakistani government prosecute LeT. Even with an expansion in Pakistani military support for LeT, this provides hope that covert action on India's part and cuts in US aid to Pakistan will lead to reduced terrorist attacks by LeT.

We now consider $(\epsilon, 2)$-equilibria, for $\epsilon \in \{0.0, 0.1, 0.2\}$. Though we computed

$(\epsilon, 2)$-equilibria for $\epsilon = \{0.3, 0.4, 0.5, \ldots\}$, all of these equilibria involve players giving up 30% or more of their payoffs—something that we think is unlikely.

Of the 252 $(0, 2)$-equilibria, there were five equilibria in which the US cut aid, India carried out either covert operations against LeT or coercive diplomacy against Pakistan, and the Pakistani military cracked down on LeT. In every one of these situations, LeT either eliminated its armed wing or did nothing, and the Pakistani government either prosecuted LeT or did nothing. Moreover, there are 24 $(0, 2)$-equilibria in which the US cuts aid and India carries out either covert action or coercive diplomacy—and in 5 of these 24 equilibria, LeT either eliminated its armed wing or did nothing. However, the situation is more complex. In our data, we noticed that one expert's payoffs were significantly different from those of the other two. In fact, there were vastly more equilibria between experts #2 and #3 than between experts #1 and #2 or between #1 and #3, suggesting expert #1 was a bit of an outlier. If we only consider experts #2 and #3, then the proportion of "good" equilibria where LeT stands down with the US cutting aid to Pakistan and India either engages in covert action or coercive diplomacy against Pakistan rises to 5 out of only 14. Of course, other inducements not considered in this study can be used to get the Pakistani military to crack down on LeT.

We continued the same analysis of the 357 $(0.1, 2)$-equilibria. There were a 23 equilibria where the US cut aid and India acted covertly. Of these, 6 equilibria led to LeT either disbanding its armed wing or doing nothing—good outcomes for peace. If we ignored expert #1 (who continued to be an outlier when we considered $(0.1, 2)$-equilibria), the number of "good" equilibria remained the same, with fewer (20) overall equilibria. Again, when the Pakistani military cracked down on LeT, there was a 100% chance of LeT either eliminating its armed wing or getting rid of terrorism altogether.

When we look at the 1696 $(0.2, 2)$-equilibria, we see a similar pattern. We had a total of 51 $(0.2, 2)$-equilibria, of which LeT cut attacks in 9. There were only 51 of these $(0.2, 2)$-equilibria in which the US cut aid and India took either covert action or engaged in coercive diplomacy. However, we note that when the Pakistani military also cracks down on LeT (in addition to the US and Indian actions just described), the majority (8 out of 9) of the remaining equilibria involve LeT eliminating its attacks.

5.8 Related Work

We group our survey of related work into three sections: first, the purely theoretical aspects of computational game theory; second, the application of (computational and traditional) game theory to counter-terrorism and modeling conflict; and third, dealing with the purely social science study of Lashkar-e-Taiba.

5.8.1 Computational Game Theory

Games where each player has multiple payoffs have been studied before under many names such as vector-valued games [119], multi-criteria games [89], games with multiple payoffs [140], and multiple objective games [141]. However, past work mainly focuses on multiple payoffs as a way to model the situation where each player is trying to optimize many non-tradable and non-monetizable criteria simultaneously. For such games, Pareto equilibria [19] and its variants [89] have been the solutions of choice. However, the situation we consider compares alternate realities subscribed to by each expert. Hence, we are not interested in Pareto optimality.

As explained earlier, our motivation for this work is to analyze a simultaneous game when experts disagree on payoffs for players. Different experts (with unknown accuracy of prior knowledge) provide payoff functions for each player—and we expect experts to differ on such payoff functions because of subjective judgment in such applications. Since our work requires computation of approximate Nash equilibria that are common to all given payoffs, the problem is related to enumeration of Nash equilibria in the multiplayer setting.

The computation of even a single Nash equilibrium for two-player games is a hard problem [27, 34]. Enumeration of all Nash equilibria for a multiplayer game is likely to be an even harder problem [12]. There are also some hardness results known for computation of approximate Nash equilibria. It has been proven that it is unlikely that an FPTAS exists for the problem of finding Nash equilibria in a two-player game [27]. Multiplicative approximation of Nash equilibria is also PPAD-complete for a constant approximation factor—even for two-player games [33].

Recently, there has been considerable progress in computation of approximate Nash equilibria for two-player games. The best known approximation factor for a polynomial time algorithm is 0.3393 [129]. However, most recent work focuses on computation of a single Nash equilibrium for two-player games.

Theobald 2009 studies enumeration of Nash equilibria for two-player games of rank 1; however, that algorithm is not known to run in polynomial time. Lipton, Markakis and Mehta 2003 give the first QPTAS for computation of Nash equilibria in two-player and multiplayer games. However, the exponent depends on the inverse square of the approximation factor and on the square of the number of players and hence the algorithm is not feasible in practice. In fact, it has been proven by Feder, Nazarzaded and Saberi 2007 that, as far as brute force search over uniform strategy profiles is concerned, the runtime for these algorithms is tight. However, the above two results do indicate that, in general, a uniform grid search over the strategy space is a good heuristic for finding approximate Nash equilibria.

A particularly pertinent paper is that of Kalyanaraman and Umans 2007, which defines constant rank multiplayer games and gives a PTAS for finding approximate Nash equilibria for such games. We prove a structural theorem and also give a polynomial time algorithm for computation of Nash equilibria and well-supported multiple ϵ-approximate Nash equilibria for the rank 1 case. We also prove that when players have a small number of strategies to choose from, an assumption which holds for many real-world games, then a uniform strategy does well in the constant rank case.

Games we consider cannot be modeled as Bayesian games because Bayesian games require knowledge about priors. We study the situation where experts differ in their perception of payoffs and *where we have no prior over the accuracy of each expert*. Our definitions of equilibria for multiple payoffs are closer in spirit to definition of *minimax-regret equilibrium* given by Hyafil and Boutilier 2004.

5.8.2 Game Theory and the Study of Conflict

Though there has been extensive work on the use of game theory for political analysis, almost none of it involves large multiplayer games, and almost none of it involves the use of formal computational methods. The use of game theory to study conflict was pioneered by Schelling 1980, who developed a social scientist's view of how two-player conflicts—including terrorism—could be studied via game theory. Later, Bueno de Mesquita 2010 recounts how he used two-person games to predict various actions including one of interest in this project, namely that current US President Obama would not be able to stop Pakistani-based terrorism. Both these and similar efforts focus on two-player games; in contrast, the theory of equilibria in multiplayer games with multiple payoff matrices was not described by either of them. Lastly, this work uses the LeT game proposed by Dickerson et al. 2011. In contrast to this work, which used only one payoff matrix corresponding to the views of a single expert, we use a multiple payoff matrix model in this work for which the relevant game theory and the resulting implications for dealing with LeT had to be completely reconsidered.

Ozgul et al. 2007 have studied the problem of detecting terror cells in terror networks and proposed a variety of algorithms such as the GDM and OGDM methods. Similarly, Lindelauf et al. 2009 have studied the structure of terrorist networks and how they need to maintain sufficient connectivity in order to communicate while simultaneously maintaining sufficient disconnectivity in order to stay hidden. They model this tension between communication and covertness via a game-theoretic model. This same intuition led to the concept of covertness centrality [106] in social networks where a statistical (rather than game-theoretic) method is used to predict covert vertices in a network.

Sandler and Enders 2004 use the ITERATE data set of terrorist events to discuss how economic methods including both game theory and time series analysis can be used to propose policies for counter-terrorism. In an earlier survey [48], the same authors specify how game theory might be used to model target selection by terrorists. Major 2002 uses a mix of game theory, search, and statistical methods to model terrorism risk. None of these works provide a formal game-theoretic model involving both multiple players and multiple payoff matrices.

5.8.3 Research on and Analysis of Lashkar-e-Taiba

On the social science side, Clark 2010 was the first to study LeT from a military perspective. He argues that LeT has grown beyond the control of Pakistan and the Directorate for Inter-Services Intelligence (ISI), and that it will continue to grow with help from fringe elements in the Pakistani military establishment. He argues that India can only insulate itself from LeT-backed attacks by diminishing the internal threat posed by the Indian Mujahideen, an Indian group closely affiliated with LeT.

Tankel 2011 wrote a detailed analysis of LeT based on years of field work and multiple visits to Pakistan to interview both LeT operatives as well as members of Pakistan's ISI. He provides a wonderful insight into LeT's origins, ideology, and operational structure, but does not include a policy analytics section specifically saying how to deal with the menace posed by LeT. John's excellent volume [70] on the same topic provides another in-depth study of LeT but does not propose policies on how the US and/or India can collectively help reduce LeT attacks.

Virtually all past work on counter-terrorism policy is qualitative (see work by Mannes 2013 for an overview). A group of experts gather around a table, hypothesize about the impacts of different possible policies, and then decide which one to use. It is only recently that quantitative methods for generating policies against terror groups have started playing a role. Data mining approaches have been used to study the Pakistani terror group Lashkar-e-Taiba [122] with considerable impact in the strategic policy community in both the US and India, both of whom have attended talks on the results. Subrahmanian et al. 2012 performs a data mining study of LeT involving 770 variables that are analyzed via data mining algorithms to learn the conditions under which LeT executes various types of attacks. It goes on to consider the problem of shaping the behavior of LeT by using abductive inference models. Another excellent recent book on Pakistan in general by Bruce Riedel 2012, a former top CIA official who advised the last five US presidents on relations with India and Pakistan, lays much of the blame for terrorism out of Pakistan (including LeT terrorism) squarely at the doorstep of the Pakistani intelligence agency but does not address LeT attacks in particular.

5.9 Conclusions

In this chapter, we showed how to merge vector payoffs [119] and well-supported ϵ-approximate equilibria [34, 35] so as to handle the problem of efficient computation of equilibria in multiplayer games where multiple experts provide different payoff matrices, each capturing their own perception of reality. We present efficient algorithms to find such equilibria—as well as a QPTAS—and experimental results showing they work. The work is motivated by a real-world game we have built to formulate policies against the terror group Lashkar-e-Taiba (LeT) which carried out the 2008 Mumbai attacks. We then presented PREVE, a set of algorithms based on multiplayer game theory that extends a game developed earlier [43] to the case where there are multiple payoff matrices that reflect differing opinions of different experts. As a consequence, the resulting equilibria are much more robust to variations than the equilibria developed in [43] that are very sensitive to minor changes in the payoff matrix.

Pakistan is widely recognized as being one of the biggest threats to global security today because of several factors: (i) its nuclear arsenal, (ii) the large milieu of violent terrorist and extremist groups in the area with close ties to Pakistani intelligence, (iii) tensions with India, and (iv) a collapsing economy. In this chapter, we have focused primarily on Pakistan-India relations, which India views primarily through the lens of terrorist acts in India that are backed by the Pakistani military and are usually operationally executed by LeT and/or its allies, like the Indian Mujahideen.

The PREVE theory, framework, and code have been developed in order to help policymakers with an interest in peace in South Asia determine the best ways for the parties involved to move forward in order to reduce the threat of Lashkar-e-Taiba. Though we applied PREVE only to LeT in this chapter, the theory is general and can be applied to any set of actors with any set of actions as long as one or more payoff matrices are available. In this work, area experts used *open source* data to create payoff matrices for our five-player game.

From a public policy perspective, the results of this chapter may support three ideas.

1. The US must cut aid to Pakistan. There are no equilibria where LeT behaves well where the US is providing aid to Pakistan. However, we do not have a recommendation for exactly how much this cut should be—only that cuts need to be made.

2. India must engage in additional covert action against LeT and its allies and/or coercive diplomacy towards Pakistan. By cutting aid, the US would intuitively increase political and economic pressure on the Pakistani establishment, leading to a potential loss of support for the Pakistani military leadership amongst the Pakistani people. By engaging in covert action, India would put operational constraints on LeT, making attacks harder by "taking the fight to them" as the US has done against Al-Qaeda. By taking steps towards coercive diplomacy, India

would concurrently increase pressure on the Pakistani government and military, complementing the US aid cuts proposed.

3. The key policy element is getting the Pakistani military to crack down on LeT, in conjunction with US cuts on aid to Pakistan and covert action/coercive diplomacy by India. The key question is how to induce the Pakistani military to crack down on LeT. An examination of the deep social, political, economic, and jihadist links that the Pakistani military has could lead to better understanding of the pressures that might induce them to crack down on extremist elements, many of whom they currently support.

PREVE is a codebase, not an operational system. Top politicians and policymakers are busy and are often more interested in white papers addressing their problem than learning how to use software systems. In our case, PREVE has been used to generate these results and then generate a report interpreting the results for policymakers. The results of this study have been disclosed to top government officials in both the US and Indian government. There is significant interest in continuing these studies.

5.10 Additional Experiments

In this section, we present experimental results for Algorithm 7 on simulated data. First, we present results showing the algorithm's running time and output on generated games. Second, we explore the relationship between various traits of the game and the percentage of strategies that are equilibria. The framework was implemented in about 700 lines of C++, and the experiments were run on a 4-CPU, 4-core Intel Xeon 3.4GHz machine with 64GB of RAM running Ubuntu 12.04.

To test the scaling properties of Algorithm 7, we built a game generator and varied the number of experts (each giving one set of payoff matrices), players, and actions per player. We also varied the granularity factor t when generating t-uniform strategies.

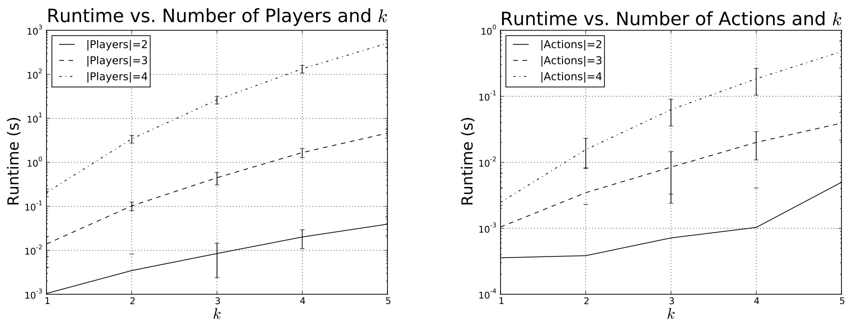


Figure 5.1: Runtime as the number of players increases (left) and number of actions increases (right) for t-uniform factor $t \in \{1, \ldots, 5\}$.

Figure 5.1 shows the runtime of Algorithm 7 on generated data as both the number of players and number of actions increase, for varying granularity factors. As expected, increasing the number of players (while holding the number of actions constant) hurts runtime significantly more than increasing the number of actions (while holding the number of players constant). Similarly, increasing the granularity factor t (shown on the x-axis) exponentially increases the number of possible strategy profiles over which the algorithm must iterate, resulting in large runtime increases. Future research would increase the algorithm's equilibrium-generation capabilities to games with many players and many actions.

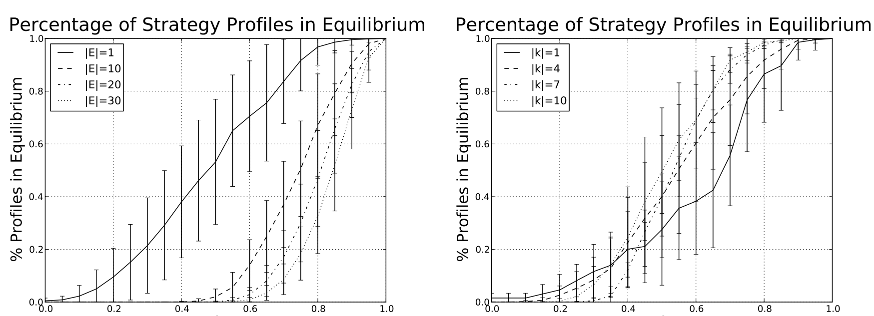


Figure 5.2: Percentage of all sets of strategy profiles that are well-supported multiple ϵ-approximate Nash equilibria as the number of experts increases (left) and t-uniform factor increases (right), for $\epsilon \in \{0.0, 0.05, \ldots, 1.0\}$.

Figure 5.2 quantifies the relationship between the ϵ-approximation threshold and the percentage of strategy profiles that are well-supported multiple ϵ-approximate Nash equilibria. Intuitively, increasing the slack in the approximation factor ϵ yields a higher percentage of strategy profiles being equilibria, while increasing the number of potential payoff matrices decreases this percentage of strategy profiles. The rate of increase of this line is highly dependent on the distribution of payoffs to each individual player. With random generation of payoffs, the increase is fairly steady; however, a more structured (e.g., real-world) payoff function would affect this trend. In Section 5.7, we considered such a real-world game.

5.11 Proofs

In this section, we provide complete proofs for various theorems and lemmas in the chapter.

5.11.1 Proofs for Section 5.5.1

Lemma 40

Proof. Let (σ^1, σ^2) be a feasible solution to the given LP. Let $p = \sum_{i \in [m]} \sum_{j \in [m]} \sigma_i^1 \sigma_j^2 u(e_i, e_j)$ be the payoff for player 1. The payoff for player 2 will be $-p$. Then, we have:

$$\begin{aligned} p &= \sum_{i \in [m]} \sum_{j \in [m]} \sigma_i^1 \sigma_j^2 u(e_i, e_j) = \sum_{i \in [m]} \sigma_i^1 \sum_{j \in [m]} \sigma_j^2 u(e_i, e_j) \\ &\leq \sum_{i \in [m]} \sigma_i^1 (r + \epsilon) \quad (\textbf{from } (5.6)) \\ &= r + \epsilon \because \sum_{i \in [m]} \sigma_i^1 = 1 \end{aligned} \tag{5.12}$$

Similarly,

$$p = \sum_{j\in[m]} \sigma_j^2 \sum_{i\in[m]} \sigma_i^1 u(e_i, e_j) \geq \sum_{j\in[m]} \sigma_j^2 (r - \epsilon) \quad (\textbf{from } (5.5))$$
$$= r - \epsilon \because \sum_{j\in[m]} \sigma_j^2 = 1 \tag{5.13}$$

From (5.12) and (5.5):

$$\sum_{i\in[m]} \sigma_i^1 (-u(e_i, e_j)) \leq -p + 2\epsilon, \forall j \in [m] \tag{5.14}$$

Similarly, from (5.13) and (5.6):

$$\sum_{j\in[m]} \sigma_j^2 u(e_i, e_j) \leq p + 2\epsilon, \forall i \in [m] \tag{5.15}$$

Since p and $-p$ are payoffs for given strategies and u and $-u$ are the payoff functions for players 1 and 2 respectively, the claim follows from (5.14), (5.15) and the definition of approximate Nash Equilibrium (Definition 11). □

Lemma 41

Proof. Let $p = \sum_{i\in[m]} \sum_{j\in[m]} \sigma_i^1 \sigma_j^2 u(e_i, e_j)$ be the payoff for player 1. Because its a zero-sum game, the payoff for player 2 will be $-p$. Then, from Definition 11 of approximate Nash Equilibrium:

$$\sum_{i\in[m]} \sigma_i^1(-u(e_i,e_j)) \leq -p+\epsilon, \forall j \in [m]$$
$$\implies \sum_{i\in[m]} \sigma_i^1 u(e_i,e_j) \geq p-\epsilon, \forall j \in [m] \quad \text{(multiplying by -1)}$$
$$\implies \sum_{i\in[m]} \sigma_i^1 u(e_i,e_j) \geq r-\tau, \forall j \in [m] \because p \geq r-\epsilon, 2\epsilon \leq \tau \tag{5.16}$$

Similarly,

$$\sum_{j\in[m]} \sigma_j^2 u(e_i,e_j) \leq p+\epsilon, \forall i \in [m]$$
$$\implies \sum_{j\in[m]} \sigma_j^2 u(e_i,e_j) \leq r+\tau, \forall i \in [m] \because p \geq r-\epsilon, 2\epsilon \leq \tau \tag{5.17}$$

The other constraints in the LP are satisfied by any valid strategy profile. Thus, the claim follows from (5.16), (5.17) and the fact that (σ^1, σ^2) is a strategy profile. □

Lemma 42

Proof. Any feasible solution to $LP\_MEAE(U, P, \frac{\epsilon}{2})$ is a feasible solution to $LP\_EAE(u^i, r_i, \frac{\epsilon}{2}), \forall i \in [f]$ because constraints for $LP\_MEAE(U, P, \frac{\epsilon}{2})$ are a superset of constraints for $LP\_EAE(u^i, r_i, \frac{\epsilon}{2}), \forall i \in [f]$. Thus, from Lemma 40, the feasible solution is a strategy profile that is an ϵ-approximate Nash equilibrium for all constituent games of the ZSGM. The result then follows from Definition 15 of multiple ϵ-approximate Nash equilibrium. □

Lemma 43

Proof. From Lemma 41, any ϵ-approximate Nash equilibrium for zero-sum game with payoff matrix u^i such that payoff for player 1 is between $r_i - \epsilon$ and $r_i + \epsilon, \forall i \in [f]$ satisfies all constraints of $LP\_EAE(u^i, r_i, 2\epsilon)$. Thus, the given equilibrium is feasible for $LP\_EAE(u^i, r_i, 2\epsilon), \forall i \in [f]$. Hence the given equilibrium satisfies $LP\_MEAE(U, P, 2\epsilon)$. □

Theorem 44

Proof. All LPs in the returned set are of the form $LP\_MEAE(U, P, \frac{1}{k})$. From Lemma 43, all strategy profiles that are feasible for these LPs are multiple ϵ-approximate Nash equilibrium with $\epsilon = \frac{2}{k}$ and hence the first condition is satisfied. From Lemma 42, all feasible solutions are ϵ-approximate Nash equilibria with $\epsilon = \frac{1}{2k}$ for the given game and hence the second condition is satisfied. For computing the runtime of the algorithm, we observe that the **for** loop in the algorithm runs $(k+1)^f$ times and each time it outputs an LP of size $2mf + 2m + 2$ which can be appended to a data structure such as a list in constant time. Thus, the algorithm can be implemented in time $O((k+1)^f(2mf + 2m + 2))$. □

5.11.2 Proofs for Section 5.5.2

Lemma 45

For arbitrary real numbers, we have, by matching coefficients on LHS and RHS,

$$\sum_{a\in A}\prod_{i\in[n]} x^i_{a_i} \equiv \prod_{i\in[n]}\sum_{l\in[m]} x^i_l \tag{5.18}$$

From (5.9), by combining the two product terms into one, we get: $u_j(\sigma) = \sum_{a\in A}\prod_{i\in[n]}\sigma^i_{a_i}\alpha^{i,j}_{a_i}$. From (5.18), with $x^i_{a_i} = \sigma^i_{a_i}\alpha^{i,j}_{a_i}$, we get

$$u_j(\sigma) = \prod_{i\in[n]}\sum_{l\in[m]}\sigma^i_l\alpha^{i,j}_l \tag{5.19}$$

This function is not a function of j's strategy. When player j's strategy is pure and its support is just an action, say $l \in [m]$, substituting value of σ_j as e_l in (5.19), we have:

$$u_j(e_l, \sigma_{-j}) = \alpha_l^{j,j} \prod_{i \in [n]-\{j\}} \sum_{l \in [m]} \sigma_l^i \alpha_l^{i,j} = \alpha_l^{j,j} u'_{-j}(\sigma) \tag{5.20}$$

A necessary and sufficient condition for Nash equilibrium is that only the best pure responses can be in support of each player's strategy Let $\sigma = (\sigma^1, \sigma^2, .., \sigma^n)$ be a Nash equilibrium for the above game. Let E be the set of all Nash equilibria for the given game. Therefore, for any player, j, with a positive payoff and any action a in support of σ^j, we have:

$$\sigma \in E \iff u_j(e_a, \sigma_{-a}) \geq u_j(e_l, \sigma_{-a}), \forall l \in [m] \tag{5.21}$$

$$\iff \alpha_a^{j,j} u'_{-j}(\sigma) \geq \alpha_l^{j,j} u'_{-j}(\sigma) \text{ (Substituting from (5.20))} \tag{5.22}$$

Assuming that $u'_{-j} > 0$ (otherwise, j can play any action without affecting his payoff, which remains 0), we have:

$$(5.22) \iff \alpha_a^{j,j} \geq \alpha_l^{j,j} \iff a \in S_j \text{ (From defn. of } S_j) \tag{5.23}$$

Since the above is true $\forall a \in support(\sigma^j)$ and $\forall j \in [n]$, the claim follows.

Lemma 46

Proof. A necessary and sufficient condition for well-supported relative ϵ-approximate Nash equilibrium is that only the approximate pure best responses can be in support of each player's strategy. Let W be the set of all well-supported relative ϵ-approximate Nash equilibria (with non-zero payoffs for each player) for the given game. Therefore, for any player, j, with a positive payoff and any strategy i in support of σ^j, we have:

$$\begin{aligned}
\sigma \in W &\iff u_j(i, \sigma_{-i}) \geq (1-\epsilon) u_l(l, \sigma_{-i}), \forall l \in [m] \\
&\iff \alpha_i^{j,j} u_j'(\sigma) \geq (1-\epsilon) \alpha_l^{j,j} u_j'(\sigma), \forall l \in [m] \\
&\iff \alpha_i^{j,j} \geq (1-\epsilon) \alpha_l^{j,j}, \forall l \in [m] \\
&\quad\;\; \alpha_i^{j,j} \geq (1-\epsilon)(\max \alpha^{j,j})
\end{aligned}$$

Thus, equation (5.10) above is the complete characterization of well-supported relative ϵ-approximate Nash equilibria for the given game. □

Theorem 47

Proof. Let σ be a strategy profile for the given game for which equation (5.11) holds. From the definition of well-supported multiple relative ϵ-approximate Nash equilibrium (Definition 17), a strategy profile is well-supported multiple relative ϵ-approximate Nash equilibrium iff it is a well-supported ϵ-approximate Nash equilibriumfor each constituent game of the SGM. Since equation (5.11) holds for σ for all constituent games, from Lemma 46, σ is well-supported multiple relative ϵ-approximate Nash equilibrium for the given game. Thus, from Lemma 46, σ satisfies equation (5.10) for all constituent games. Thus, σ satisfies equation (5.11). □

5.11.3 Proofs for Section 5.5.3

Lemma 50

Proof. From definition of well-supported ϵ-approximate Nash equilibrium, for any action $a \in Support(\sigma^j)$:

$$u_j(e_a, \sigma_{-j}) \geq u_j(e_l, \sigma_{-j}) - \epsilon, \forall l \in [m] \tag{5.24}$$

Now, we construct a t-uniform strategy profile, σ', from σ as follows. Let $\sigma'_i = \frac{\lceil \sigma_i t \rceil}{t}$. The above rounding procedure can make $||\sigma'||_1$ greater than 1. To counter this, select any element in σ' arbitrarily (without replacement) and round it down to $\frac{\lfloor \sigma_i t \rfloor}{t}$. Repeat this until $||\sigma'||_1 \neq 1$. The above procedure is guaranteed to give a t-uniform strategy profile σ' such that:

$$|\sigma'_i - \sigma_i| \leq \frac{1}{t} \tag{5.25}$$

From the definition of multiplayer games of low rank, the payoff is given by:

$$u_j(\sigma) = \sum_{k=1}^{K} \prod_{i \in [n]} \sum_{l \in [m]} \sigma_l^i \alpha_l^{i,j,k} \tag{5.26}$$

From above, when player j plays action a with probability 1 and the rest of the players play their respective strategies in σ, we have:

$$u_j(e_a, \sigma_{-j}) = \sum_{k=1}^{K} \alpha_a^{j,j,k} \prod_{i \in [n]-\{j\}} \sum_{l \in [m]} \sigma_l^i \alpha_l^{i,j,k} \tag{5.27}$$

Thus, we have:

$$\begin{aligned} |u_j(e_a, \sigma_{-j}) - u_j(e_a, \sigma'_{-j})| =& |\sum_{k=1}^{K} \alpha_a^{j,j,k} \prod_{i \in [n]-\{j\}} \sum_{l \in [m]} \sigma_l^i \alpha_l^{i,j,k} \\ & - \sum_{k=1}^{K} \alpha_a^{j,j,k} \prod_{i \in [n]-\{j\}} \sum_{l \in [m]} {\sigma'}_l^i \alpha_l^{i,j,k}| \end{aligned}$$

By taking the outermost summation and the common factor $\alpha_a^{j,j,k}$ out, we get:

$$\begin{aligned}
&|u_j(e_a, \sigma_{-j}) - u_j(e_a, \sigma'_{-j})| \\
=&|\sum_{k=1}^{K} \alpha_a^{j,j,k} (\prod_{i \in [n]-\{j\}} \sum_{l \in [m]} \sigma_l^i \alpha_l^{i,j,k} - \prod_{i \in [n]-\{j\}} \sum_{l \in [m]} \sigma_l'^i \alpha_l^{i,j,k})| \\
=&|\sum_{k=1}^{K} \alpha_a^{j,j,k} (\prod_{i \in [n]-\{j\}} x_{i,k} - \prod_{i \in [n]-\{j\}} x'_{i,k})| \qquad (5.28)
\end{aligned}$$

Where, $x_{i,k} = \sum_{l \in [m]} \sigma_l^i \alpha_l^{i,j,k} = (\alpha^{i,j,k})^T \sigma$ and
$x'_{i,k} = \sum_{l \in [m]} \sigma'^i_l \alpha_l^{i,j,k} = (\alpha^{i,j,k})^T \sigma'$. From Lemma 48 and Equation 5.25, we have $|x_{i,k} - x'_{i,k}| \le \frac{m}{t}$. From Lemma 49 and the above, we have:

$$|\prod_{i \in [n]-\{j\}} x_{i,k} - \prod_{i \in [n]-\{j\}} x'_{i,k}| \le \frac{(n-1)m}{t}$$

From above and Equation 5.28, we have:

$$\begin{aligned}
|u_j(e_a, \sigma_{-j}) - u_j(e_a, \sigma'_{-j})| &\le |\sum_{k=1}^{K} \alpha_a^{j,j,k} \frac{(n-1)m}{t}| \\
&\le \frac{(n-1)mk}{t} \because \alpha_a^{j,j,k} \le 1 \qquad (5.29)
\end{aligned}$$

From above and Equation 5.24, we have, for every action a in support of σ':

$$u_j(e_a, \sigma'_{-j}) \ge u_j(e_l, \sigma'_{-j}) - \frac{2(n-1)mk}{t} - \epsilon, \forall l \in [m]$$

Thus, from the definition, σ' is a well-supported $\epsilon + \frac{2(n-1)mk}{t}$-approximate Nash equilibrium. □

Theorem 51

Proof. From Lemma 50, given a well-supported τ-approximate Nash equilibrium strategy profile, we can always construct a t-uniform strategy profile that is a well-supported $\tau + \frac{(n-1)mk}{t}$-approximate Nash equilibrium. From the definition of well-supported multiple ϵ-approximate Nash equilibrium, σ is a well-supported τ-approximate Nash equilibrium strategy profile for each of the constituent games. The construction of σ' from σ, as described in the proof of Lemma 50, is independent of payoff functions for constituent games and hence the lemma applies simultaneously to all the constituent games of the SGM. Hence σ' is a well-supported $\tau + \frac{(n-1)mk}{t}$-approximate Nash equilibrium for all the constituent games of the SGM. Thus, from the definition of well-supported multiple ϵ-approximate Nash equilibrium, σ' is a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau + \frac{(n-1)mk}{t}$ and the claim follows. □

5.11.4 Proofs for Section 5.6.2

Lemma 52

Proof. Let $\sigma = (\sigma^1, \dots, \sigma^n)$ be a Nash equilibrium. Construct a multiset M^j by sampling independently at random from the set of actions according to distribution σ^j. Construct strategy ρ^j by assigning probability of $\frac{l}{t}$ to an action that appears l times in M_j and $\rho^{-j} = \sigma^{-j}$. We prove that ρ is an $\tau + \delta$-well supported multiple payoff approximate Nash equilibrium **w.p.p**.

Let $M^j_s \in [m]$ be a random variable that denotes the s^{th} element of M^j. Let $Y_s = u^i_l(e_{M^j_s}, e_k, \sigma^{-j,l})$ be a random variable that is equal to payoff for player l, when player l plays action k and player j plays action M^j_s in constituent game i. Let μ be the mean of random variables $\{Y_1, \ldots, Y_t\}$. Then, we have:

$$\begin{aligned}
\mu =& \sum_{s=1}^{t} \frac{u_l^i(e_{M_s^j}, e_k, \sigma^{-j,l})}{t} \\
=& \sum_{s=1}^{t} \frac{u_l^i(e_{M_s^j}, e_k, \rho^{-j,l})}{t} \quad \textbf{(from } (\rho^{-j} = \sigma^{-j})) \\
=& u_l^i(\sum_{s=1}^{t} \frac{e_{M_s^j}}{t}, e_k, \rho^{-j,l}) \quad \textbf{(from } \text{(definition of } u_l^i)) \\
=& u_l^i(e_k, \rho^{-l})
\end{aligned}$$

Where, the last equality follows from the fact that if an action occurs l times in M^j, it gets a weight of $\frac{l}{t}$ is ρ^j. Also, from construction of ρ, it follows that $E(\rho_k^j) = \sigma_k^j$. From linearity of expectation it follows that $E(u_l^i(e_k, \rho^{-l})) = u_l^i(e_k, \sigma^{-l})$. Thus, $\forall i \in [f], l \in [n] - \{j\}, k \in [m], u_l^i(e_k, \rho^{-l})$ is the mean of t i.i.d. random variables, each with expectation $u_l^i(e_k, \sigma^{-l})$. Payoffs are in $[0, 1]$ and the same bound applies to all these random variables. Also, since only player j's strategy is changed, $u_j^i(e_k, \sigma^{-j}) = u_j^i(e_k, \rho^{-j})$.

Let $A(i, k, l)$ be the event $|u_l^i(e_k, \rho^{-l}) - u_l^i(e_k, \sigma^{-l})| \geq \frac{\delta}{2}$. From Hoeffding inequality, we have:

$$Pr[A(i, k, l)] \leq 2 \exp{(\frac{-t\delta^2}{2})}$$

Let $A = \bigcup_{i \in [f], k \in [m], l \in [n] - \{j\}} A(i, k, l)$. From union bound:

$$Pr[A] \leq 2fm(n-1) \exp{(\frac{-t\delta^2}{2})}$$

Thus, $2fmn \exp{(\frac{-t\delta^2}{2})} = 1 \implies Pr[A^c] > 0$. Therefore, $t = \frac{2\log(2fmn)}{\delta^2}$ ensures that $\exists \rho$ for which event A^c occurs. Let π be the strategy profile for which this happens.

From definition of A^c and well-supported multiple ϵ-approximate Nash equilibrium, we have, $\forall i \in [f], \forall k \in [m], \forall l \in [n]$:

$$u_l^i(e_k, \sigma^{-l}) \leq u_l^i(e_h, \sigma^{-l}) + \tau \quad (\textbf{from} \text{ (Definition 16)})$$
$$\Longrightarrow u_l^i(e_k, \pi^{-l}) \leq u_l^i(e_h, \pi^{-l}) + \tau + \delta \quad (\textbf{from} \text{ (definition of } A^c))$$

Thus π is well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau + \delta$. □

Theorem 53

Proof. Let $t = \frac{2n^2 \log(2fm(n-1))}{\delta^2}$. Then by application of Lemma 52 for player 1, we get a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau + \frac{\delta}{n}$ with t-uniform strategy of player 1. To the resulting strategy profile, we can apply Lemma 52 to the strategy for player 2 and get a strategy profile that is a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau + \frac{2\delta}{n}$ with t-uniform strategy of players 1 and 2. We can do this successively for all players and get a t-uniform strategy profile which is a well-supported multiple ϵ-approximate Nash equilibrium with $\epsilon = \tau + \delta$ □

Algorithm 7 well-supported multiple ϵ-approximate Nash equilibrium for constant rank games

Input: t, payoff functions for the SGM

Output: t-uniform well-supported multiple ϵ-approximate Nash equilibrium strategy profiles

```
S ← Set of all possible t-uniform strategies
E ← ∅
Σ ← ⨉_{i=1}^{n} S                                   ▷ Cardinality: O(m^{kn})
for l ∈ [f] do                ▷ f is the number of constituent games of the SGM
    E_l ← ∅
    for σ ∈ Σ do
        isEquilibrium ← TRUE
        for j ∈ [n] do
            if not isEquilibrium then
                break
            end if
            payoff ← u_j^l(σ)
            for i ∈ [m] do
                payoff_i ← u_j^l(e_i, σ_{-j})
                if payoff_i − payoff > ϵ then
                    isEquilibrium ← FALSE
                    break
                end if
            end for
        end for
        if isEquilibrium then
            E_l ← E_l ∪ {σ}
        end if
    end for
end for
return E_1 ∩ E_2 ∩ . . . ∩ E_f
```

Chapter 6

Payoff Inference

6.1 Introduction

Virtually almost all work in game theory starts with a payoff matrix. In his pioneering study of conflict, Schelling [117] starts out with a payoff matrix for virtually every scenario. Unfortunately, getting a payoff matrix poses an enormous challenge in many real-world strategic games. In this chapter, we answer the following question: *Given a body of historical data about the interactions of multiple players, is there a way to learn a payoff matrix?* In order to answer this question, we make certain assumptions:

Time Discounting. We believe that players are more likely to be influenced by "recent" history as opposed to events from a distant past. In order to model this, we developed a notion of time-discounted regret.

No Correlated Equilibria, Short Histories. We do not assume correlated equilibria [52] nor do we assume the existence of a signaling mechanism. When long histories are available and some extra assumptions are made e.g. [52], game play can converge to correlated equilibrium even without a signaling mechanism. However, our real-world applications have short histories for which convergence cannot be assumed.

[0] This is a joint work with E. Serra, M. T. Hajiaghayi, S. Kraus and V. S. Subrahmanian.

Bounded Rationality. Unlike past work on inverse reinforcement learning that assume fully rational agents, we assume bounded rationality, i.e. players take actions whose payoffs are within ϵ percent of the action with best response payoff.
No Knowledge of Outcomes. We assume that we only know the game history but nothing about outcomes.
Best Response. We assume that all players have complete knowledge of the history of past events and that in each time period, players choose an action that is an approximate best response to the history (subject to the bounded rationality assumption).

We define constraints whose variables represent the payoffs for each player under each joint action. Our constraints informally state that at each time point t in the past, each player i chose to perform the action for which he had the maximal expected time-discounted regret prior to time t. We also interpret these constraints as a myopic best response to the state of the world. This leads to a set of constraints with many possible solutions. We define three heuristics to estimate payoffs.

1. **Centroid Solution (CS).** In CS, the (approximate) centroid of the constraint polytope is picked as the solution.

2. **Soft Constraints Approach (SCA).** In SCA, we allow the rationality constraints to be violated but penalize such violations in the objective function used.

3. **SVM-based Method (SVMM).** In SVMM, we propose a heuristic method to map the payoff inference problem onto a support vector machine [32] and build a separator that captures the payoff function we wish to learn.

We implemented CS, SCA, SVMM, as well as the recent ICEL algorithm (Inverse Correlation Equilibrium Learning) for comparison [134]. We compared all 4 algorithms w.r.t. solution quality and run-time. On synthetic data where we knew the ground truth (because we generated player behavior using known payoff functions), we showed that SVMM outperforms both CS and SCA w.r.t. both solution quality and run-time. We also compared CS,SCA, and SVMM on two real-world data sets: (i) the Minorities at Risk Organizational Behavior (MAROB) dataset [10] the contains data on terrorist group behaviors and related government actions and (ii) a much more fine-grained data set [122] about the behavior of the terrorist group Lashkar-e-Taiba (LeT).[1] Again, SVMM outperformed CS and SCA. We then ran experiments comparing SVMM with ICEL. When we compare the ability of SVMM with that of ICEL to predict true behaviors from learned payoffs on the MAROB data, SVMM's ability to predict behavior from the learned payoffs was much better than that of ICEL (median Spearman Correlation Coefficient of 0.7 for SVMM, compared to just 0.114 for ICEL).

6.2 Related Work

Inverse Reinforcement Learning [101] learns payoffs of a single-agent operating in a given (usually Markovian) environment. [101] addresses the problem of learning a reward function by observing behavior of MDPs. However, they and a series of subsequent works [97, 143] assume a single rational agent in a given environment. Some recent works have focused on *Multi-agent Inverse Reinforcement Learning (MIRL)* [84, 96]. [96] examines a cooperative setting with an explicit centralized coordinator. [84] studies the problem in a 2-person zero-sum stochastic games. [136] focuses on learning payoffs for symmetric games. Waugh et al. [134] proposed an approach to predict player behavior when no payoff matrix is available. A convex optimization formulation finds a maximum entropy solution to find the predicted distribution over joint actions. Finally, payoffs are computed by using the dual of the above optimization convex problem.

Economists have studied payoff inference problems for various markets [31, 99,

[1] As no ground truth exists about payoffs for real-world players in the MAROB and LeT data sets, we learned player payoffs from a training data set and then validated them on a separate validation data set by making predictions based on learned payoffs. We emphasize the fact that this work is not about prediction – but about learning payoffs in order to understand group behavior. The goal is to understand the payoff structure for different players for different strategies so diplomats and counter-terrorism agencies can shape policies towards the terrorist groups. We use predictions solely to validate learned payoffs.

Table 6.1

PIE	Waugh	Ng-Russell	Natarajan	Lin	Wiedenbeck
multi-player	yes	no	yes	2-player	yes
non-perfect rationality	yes	no	no	no	yes
non-cooperative	no (assumes corr. eq.)	NA	no	yes (zero-sum)	yes
not symmetric	yes	NA	yes	yes	no
demonstrated w/ real data	yes	no	no	no	no
scalable	no (Non-smooth convex program)	yes (LP)	yes (LP)	no (QP)	yes

108, 123]. However, their focus is on modeling a particular market and then to use various model fitting and regression to learn the best parameters. For instance, [123] studies the effect of land use regulations on the mid-scale hotel market.

The assumption of equilibrium is common to most work on MIRL and other payoff learning methods. However, decision theory bears out the fact that human players don't follow equilibrium strategies, even when the equilibrium is unique (which is rare!) [117]. However, decision theory does highlight the importance of recency [100] and regret in human decision making. Anticipated regret is considered an important determinant of choice-behavior [29, 120, 139]. These aspects form the basis for PIE's time discounted regret minimization and myopic best response with exponentially decaying state. Thus, PIE differs from existing work on payoff inference in that we assume myopic rationality and not global rationality (equilibrium). We also assume simple game play dynamics inspired by relevant work from decision theory. In addition, we develop a fast and practical data analytic approach compared to more theoretical approach taken by most machine learning papers. We show this with experiments on two real-world datasets and show superior performance compared to Waugh et al [134]. Table 6.1 compares our work with related work.

A major driver for our work is counter-terrorism applications. The development of game-theoretic methods to analyze terrorist behavior and organization has been pioneered by Lindelauf [85, 86] and subsequently adopted by others [44, 92, 133].

6.3 Preliminaries

Let $[N] = \{1, \ldots, N\}$ be a set of players. We assume that each player i has an associated set A_i of actions that it can take. Let $\mathcal{A} = A_1 \times \cdots \times A_N$ denote the set of all possible *joint actions.* Given a joint action $a \in \mathcal{A}$, a_i is the action of player i and a_{-i} is the joint action of all other players. Let u_i be an unknown payoff function: $u_i : \mathcal{A} \to [0, 1]$. $u_i(a)$ is the payoff of joint action a for player i. Let $\mathcal{U} = \{u_1, \ldots, u_N\}$ be the set of all (as yet unknown) payoff functions where u_i is the payoff function for player i. Let $[T] = \{1, \ldots, T\}$ be a set of past time points.

Let $m = \sum_{i \in [n]} |A_i|$ be the total number of actions for all players in the game. We encode a joint action as an m-dimensional binary vector. Each action for a player is indexed from $\sum_{j \in \{1 \ldots, i-1\}} |A_j| + 1$ to $\sum_{j \in \{1 \ldots i\}} |A_j|$ in a fixed but arbitrary order. In other words, the first $|A_1|$ entries in the vector describe the actions for the first player, the next $|A_2|$ entries describe the actions for the second player, and so forth. Let v be an encoding for $a \in \mathcal{A}$. If, in the joint action represented by a, player i plays action $a_i \in A_i$ at time t, then, and only then is $v[a_i] = 1$, otherwise $v[a_i] = 0$.

Example 13. *Suppose we have two players* $1, 2$ *and suppose* $A_1 = \{a, b, c\}$ *and* $A_2 = \{a, e\}$ *are the actions they can perform. Then the dimensionality of a joint action is* 5 *and an example of the vector representation of a joint action is:*

pl-1	*pl-1*	*pl-2*	*pl-2*	*pl-2*
a	*b*	*c*	*a*	*e*
1	*0*	*0*	*0*	*1*

The first row is the player's ID and the second row is the action name. Here, the 5-dimensional vector (1,0,0,0,1) tells us that in this joint action, player 1 performed action a *and player 2 performed action* e*.*

A *history* is a sequence $H^\tau = < a^1, \ldots, a^\tau >$ where a^t is the vector of joint actions taken at time $t \in [T]$. We represent the history of a game as a matrix, H, where $H[t, a] = 1$ iff player i plays action $a \in A_i$ at time t. Thus, H_t, the t^{th} row of the history matrix represents the joint action taken by all players at time t. Likewise, the i'th column of H tells us that actions taken by player i at each time point.

A *time-weighed history*, w_t, at time t is an m-dimensional vector defined as follows:

$$w_t = \frac{\sum_{i \in \{1 \ldots t\}} \alpha^{t-i} H_i}{\sum_{i \in \{1 \ldots t\}} \alpha^{t-i}}.$$

Example 14. *Suppose we have two players $[N] = \{1, 2\}$; player 1 is a terror group and player 2 is the government. Assume that the players' actions are* `pe_g` *("political engagement with the government") for player 1 and* `pe_tg` *("political engagement with the terror group") for player 2. Each of these variables has 3 possible levels (low, medium, high) of intensity. Therefore, player 1 has three actions,* $\texttt{pe\_g}(l), \texttt{pe\_g}(m), \texttt{pe\_g}(h)$*, corresponding to the three levels of intensity of this action, and similarly, player 2 has three actions* $\texttt{pe\_tg}(l), \texttt{pe\_tg}(m), \texttt{pe\_tg}(h)$*. Let indices of the actions* $\texttt{pe\_g}(l), \texttt{pe\_g}(m), \texttt{pe\_g}(h)$ *be 1, 2 and 3 respectively for player 1, and 4, 5 and 6 respectively for player 2. Suppose we have three years ($[T] = \{1, 2, 3\}$) history below:*

$[T]$	*year*	*player* 1	*player* 2
1	*2010*	$\texttt{pe\_g}(l)$	$\texttt{pe\_tg}(m)$
2	*2011*	$\texttt{pe\_g}(m)$	$\texttt{pe\_tg}(l)$
3	*2012*	$\texttt{pe\_g}(h)$	$\texttt{pe\_tg}(m)$

Then the history matrix, H representing the above game history is given by:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

6.4 Time Discounted Regret

In this section, we define the concept of time discounted regret. Classical regret is defined with respect to a class Φ of modification functions. Each modification function $f \in \Phi$ is a mapping $f : A \to A$. Intuitively, a modification function suggests an alternative choice $f(a)$ for an action a - instead of taking action a, the player takes action $f(a)$. As there are many ways in which a player could modify his choice, we consider a set Φ of modification functions. In the context of our running counter-terrorism example, the different modification functions might correspond to all feasible actions that could replace a given action a. The *regret* for a player i is defined as:

$$R_{i,\Phi}(t) \quad = \quad \max_{f \in \Phi} \sum_{\hat{t}=1}^{t-1} u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}).$$

Here, $u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}})$ is the difference in utility for player i had he elected to take action $f(a_i^{\hat{t}})$ instead of whatever action he took at time t in the past. The summation $\sum_{\hat{t}=1}^{t-1} u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}})$ reflects the total regret that player i had w.r.t. his past actions, had he chosen to use modification function f instead of whatever method he used to select his past actions. Had player i used the modification function $f \in \Phi$ that maximizes this summation, then he would have gotten the maximal possible benefit, and the fact that he (maybe) did not use it is what leads to this regret.

When determining what action to take, players in the real world are often more influenced by recent actions than by actions in the distant past. Our notion of *time-discounted regret* takes this into account by allowing a player to discount the past at a rate α s.t. $0 < \alpha \leq 1$. After each time point, the "importance" of a past event is reduced by a factor of α. The time-discounted regret is defined as follows:

$$TDR_{i,\Phi}(t) = \max_{f \in \Phi} \frac{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}))}{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}} \qquad (6.1)$$

Because of the α parameter in the definition of TDR, for us, a history is a *timed-stamped* collection of past joint actions. This is very different from [134] which only uses the history to extract the distribution of joint actions and consider it to be a *collection (without timestamps)* of past joint actions. When $\alpha = 1$, the definitions of regret and time-discounted regret coincide.

Suppose Φ_c is the set of all functions from A to A that are *constant* functions, i.e. if f is in Φ_c, there must exist an action $a' \in A$ such that for all $a \in A$, $f(a) = a'$. The *time discounted external regret* w.r.t. Φ_c is then simply given by:

$$TDER_{i,\Phi_c}(t) = \max_{\hat{a} \in A} \frac{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}))}{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}}$$

In other words, $TDER_{i,\Phi_c}$ only considers constant functions when computing time-discounted regret. We define the time-discounted external regret w.r.t. action $\hat{a}$ as:

$$TDER_i(\hat{a}, t) = \frac{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}))}{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}} \tag{6.2}$$

Intuitively, $TDER_i(\hat{a}, t)$ is the regret for player i due to the fact that she/he did not use the strategy to always play the action $\hat{a}$ in the past. We assume that for a rational player, the greater the regret w.r.t an action $\hat{a}$ in the past, the more likely it is that the player will play the action $\hat{a}$ in the future.[2]

Example 15. *Let's reconsider Example 14 with $\alpha = 0.9$. The time-discounted external regret for player 1 w.r.t. action h in the year 2013 ($t = 4$) is:*

$$TDER_1(h, 4) = \frac{0.81(u_1(h, m) - u_1(l, m)) + 0.90(u_1(h, l) - u_1(m, l)) + 1.00(u_1(h, m) - u_1(h, m))}{0.81 + 0.9 + 1.0}$$

Observe that the weights 0.81*,* 0.9 *and* 1.0 *are the weights for years* 2010*,* 2011 *and* 2012*, respectively.* □

A player is *ration* if, for each time t, the player chooses the action that caused the

[2] In simple terms: if the player had great regret about not doing something in the past, especially the recent past, then he is more likely to do it in the future, especially in the near future.

maximum time-discounted external regret in the past. Thus, our rationality constraints require that $\forall t \in [T] \setminus \{1\}$, $\forall i \in [N]$, $\forall \hat{a} \in A \setminus \{a_i^t\}$ the following condition holds:

$$TDER_i(\hat{a}, t) \leq TDER_i(a_i^t, t) \tag{6.3}$$

or, equivalently,

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a_i^t, a_{-i}^{\hat{t}})) \leq 0 \tag{6.4}$$

Bounded Rationality. As only a few players in the real world are completely rational, we introduce a parameter $\epsilon \in [0, 1]$ that is the degree of rationality. The closer ϵ is to 1, the more rational the player is, while the closer ϵ is to 0, the more irrational the player is. We replace Equation 6.3 (which assumes complete rationality) with the equation below, which allows weaker notions of rationality:

$$\epsilon \cdot TDER_i(\hat{a}, t) \leq TDER_i(a_i^t, t)$$

or equivalently

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(\epsilon \cdot u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a_i^t, a_{-i}^{\hat{t}})) \leq 0. \tag{6.5}$$

As ϵ and the α's are constants, this equation is linear. Each $u_i(-)$ term is a variable in this constraint. Let LC be the set of all linear constraints generated by Equation 6.5 above. We demonstrate them in the next example.

Example 16. *By considering only the last two years of the history in Example 14 we observe that H^T is*

$[T]$	*year*	*player* 1	*player* 2
1	*2011*	*m*	*l*
2	*2012*	*h*	*m*

If $\alpha = 0.9$ and $\epsilon = 0.8$, the rationality constraints for both players are:

$$\begin{array}{cc} (0.8\, u_1(l,l) - u_1(h,l)) \leq 0 & (0.8\, u_1(m,l) - u_1(h,l)) \leq 0 \\ (0.8\, u_2(m,l) - u_2(m,m)) \leq 0 & (0.8\, u_2(m,h) - u_2(m,m)) \leq 0 \end{array}$$

The result below states that LC is polynomial in size.

Proposition 1. *The number of variables occurring in LC is polynomial in the number of players N, the number of actions M and in the size of the history T.*

Proof. The number of constraints for each player at time $t \in [T]$ is $M - 1$. Each constraint has at most T variables. Thus, the total number of variables that can occur in LC is at most $(M - 1)NT$. □

One problem with LC is that it may have multiple solutions, some of which may be trivial. An example of a trivial solution is when the utility function returns the same value for each joint action for each player. For instance, the maximal entropy solution of LC (the entropy function is applied to all variables of LC) assigns the same utility to all combinations of players and joint actions.

Proposition 2. *Suppose $\mathcal{U} = \{u_1, \ldots, u_N\}$ is a maximal entropy solution for LC. Then for all joint actions a, a' and all players i, j, $u_i(a) = u_j(a')$.*

Proof. Let the entropy function be defined as follows

$$-\sum_{i \in [N], a \in \mathcal{A}} u_i(a)\, ln(u_i(a))$$

We obtain the maximum value of this function when $\forall i \in [N]$, $\forall a \in \mathcal{A}$ we can deduce that $u_i(a) = e^{-1}$. Since we know that when $\epsilon \in [0, 1]$,

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}} \epsilon \cdot e^{-1} \quad \leq \quad \sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}} e^{-1}$$

our rationality constraints in Equation 6.5 are satisfied. For these reasons it follows that the theorem holds. □ □

Hence, given the assumptions in this chapter, maximal entropy is not a very effective way of choosing a solution.

6.5 Non-Linear Time-Weighted History Payoffs

PIE was motivated by our ongoing counter-terrorism research. We have applied PIE to a multi-player game involving the terrorist group Lashkar-e-Taiba (responsible for the 2008 Mumbai attacks) and the governments of Pakistan and India. In such scenarios, a time-weighted history of players' actions is a representation of state of the world (which is a history of players' actions) at the time the player decides to take a new action. Time-weighted history captures the idea that recent actions may be more relevant than older ones.

The payoffs in Equation 6.5 are linear. Suppose we assume that a payoff function for player i at time t is *any* (linear or non-linear) function $\pi_i : \mathbb{R}^{m+1} \mapsto \mathbb{R}$ of the time-weighted history at time t. That is, $\pi_i(a, w_t)$ takes an action a and a time weighted history w_t (which is a vector of dimensionality m as defined earlier in Section 6.3) as input and returns a payoff value as output, specifying the payoff to player i of playing a at time t, w.r.t. w_t.

We assume that a player chooses an action at time t that has highest payoff with respect to the state of the world at time $t-1$. Let a be the action played by player i at time t. The constraint below encodes the fact that a is player i's best response.

$$\pi_i(a, w_t) \geq \pi_i(a', w_t) \; \forall a' \in A_i, i \in [N], t \in [T] \tag{6.6}$$

One such constraint needs to be written for each player i and each time t. *Note that these constraints may be non-linear as no constraints have been imposed to make*

π_i *linear.* It is easy to see that Equation 6.6 generalizes Equation 6.5. Note that this system of inequalities is feasible as assigning identical payoffs for all actions always satisfies Equation 6.6. However, this is not a good solution. Hence, it is very important to choose a "robust" solution to the above system. For real-world problems, we require that the selected solution satisfies the following properties:

(P1) The family of functions to which our payoff functions π_i belong should not have arbitrary complexity, i.e., our hypothesis space should not allow arbitrary payoff functions to avoid overfitting. On the other hand, we should allow somewhat complicated non-linear payoff functions to avoid over-simplification.

(P2) While it is reasonable to assume that players react to game history and use actions which would generate high payoffs, we cannot assume that each and every player always adheres to this heuristic at all times. Therefore, our algorithm must admit the possibility that some points in the history may violate Equation 6.6. However, Equation 6.6 should hold for most of the game history.

(P3) Last but not the least, there must be a tractable algorithm to select a solution of these constraints so that PIE can apply to real-world strategic games involving many players and dozens of actions. While the current best approach [134] in literature has been applied to games of upto 6 players with 3 actions for each player, they don't discuss runtime of their approach. As per our experiments, our best approach is 1 to 2 orders of magnitude faster.

6.6 Solution Selection

In this section, we present three approaches to select a solution of the system of constraints.

Centroid Based Solution (CS) The Centroid Based Solution uses LC (not Equation 6.6). The classical way to choose one solution of LC is to choose the maximum entropy solution. However, as proved earlier in Proposition 2, this is not useful as the maximum entropy solution assigns the same utility to all combinations of players and joint actions. In order to avoid this, we choose a centroid based approach. The *centroid solution* of LC is the mean position of all points satisfying LC. Unfortunately, computing the centroid of a convex region is computationally very complex — even approximating it is $\#P$-hard [111]. We therefore approximate the centroid by using Hit-and-Run (HAR) sampling [16]. In HAR sampling, we start with a randomly selected solution of the constraints (point in the polytope). We then randomly identify a direction and distance and head in that direction for the selected distance from the last sampled point. If we are still within the polytope, this becomes our next sampled point. If the new point is outside the polytope, we regenerate a distance and direction till a valid point within the polytope is found. This process is iterated till the desired number of sample points is generated. HAR sampling allows us to sample points from a convex polytope uniformly at random in time polynomial in the number of dimensions (number of variables of LC). We approximate the centroid by taking the component-wise mean of the sampled payoffs.

Proposition 3. *The centroid approximation described above is also a solution of LC.*

The proposition above follows as the centroid approximation is a convex combination of solutions of LC.

Soft Constraints Approach (SCA) In the Soft Constraints Approach, we again use only LC (Equation 6.6 is not used) and allow the rationality constraints to be violated by introducing a slack variable in each constraint in LC. These slack variables are denoted $s_{i,a,t}$ in the revised linear program RLP given below. We then find a solution of RLC that minimizes the sum of the slack variables which, in a sense, minimizes the amount of violation of the constraint. The *Revised Linear Program* RLP is shown below.

$$\underset{s,u}{\text{Maximize}} \sum_{i,a,t} s_{i,a,t}$$

Subject to:

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}} (\epsilon u_i(a, a_{-i}^{\hat{t}}) - u_i(a_i^t, a_{-i}^{\hat{t}}) + s_{i,a,t}) \leq 0, \forall i \in [N], a \in A, t \in [T] \quad (6.7)$$

The slack variables in Equation 6.7 are inside the parentheses to normalize for the history length and time decay of payoffs.

Payoff Inference using SVMM In this section, we present a novel approach that uses Support Vector Machines (SVM) to find a "good" candidate solution to the system of inequalities given in Equation 6.6. Here, we use a set $CONS$ of constraints which are generated by Equation 6.6 – we do not use Equation 6.5 in this approach. Each constraint generated by Equation 6.6 has a left hand side and a right hand side. We encode the right hand and the left hand sides of the inequalities in Equation 6.6 as points in a space. If a point encodes the right hand side of an inequality, it is assigned a label 1, otherwise, it is assigned a label 0. We then run the classification algorithm. The decision function for the learned classifier is the desired payoff function. We now describe this method in more detail.

Encoding points in game history. Let the number of actions of player i be $n_i = |A_i|$. Let a be an action of player i, whose index is equal to $\sum_{j \in \{1 \ldots i-1\}} |A_j| + k$. Consider an $m+1$-tuple (a, h), where $a \in A_i$ and h is an m-dimensional point. Let $V : \mathbb{R}^{m+1} \mapsto \mathbb{R}^{(m+1)n_i}$ be a map that takes this $m+1$'th-tuple as input and outputs an $(m+1)*n_i$-dimensional vector. Map V is defined as follows:

$$V(a,h)[(k-1)*(m+1)+1] = 1$$
$$V(a,h)[(k-1)*(m+1)+1+j] = h[j] \ \ \forall j \in \{1 \ldots m\}$$

All other entries of $V(a,h)$ are 0. Suppose player i plays action a at time t. Then $V(a, w_{t,i}, w_{t-1,-i})$ is labeled 1. For all actions $a' \neq a$, $V(a', w_{t,i}, w_{t-1,-i})$ are labeled 0. The following example shows how this works.

Example 17 (SVMM Method)**.** *Consider a 3 player game with players $\{1,2,3\}$ with 2 actions (namely, Action 1 and Action 2) for each player. A point in the history of the game is represented by a 6-dimensional binary vector, e.g. the vector (1,0,0,1,1,0) represents the fact that players 1, 2 and 3 played actions 1, 2 and 1 respectively. Let the current state of the world be given by the vector $w_t = (w_1, w_2, w_3, w_4, w_5, w_6)$. Assume that at any time, player 1 plays a myopic best response to this state of the world. For simplicity, let payoffs be a linear function of state of the world. The payoff for playing action 1 by player 1 is $p_1 = a_1 + \sum_{i\in\{1..6\}} a_{1i}w_i$. Similarly for action 2, $p_2 = a_2 + \sum_{i\in\{1..6\}} a_{2i}w_i$. Thus, the payoff function can be represented as a 14-dimensional vector $p = (a_1, a_{11}, \ldots, a_{16}, a_2, a_{21}, \ldots, a_{26})$. Further, assume that player 1 actually chooses action 1 as her best response. Then:*

$$p_1 >= p_2 \tag{6.8}$$

Now, lets encode RHS as $V(1, w_t) = (1, w_1, w_2, w_3, w_4, w_5, w_6, 0, 0, 0, 0, 0, 0, 0)$ and the LHS as $V(2, w_t) = (0, 0, 0, 0, 0, 0, 0, 1, w_1, w_2, w_3, w_4, w_5, w_6)$. If we use SVM to learn a separating hyperplane W for points $V(1, w_t)$ and $V(2, w_t)$ such that $V(1, w_t)$

is on the positive side and $V(2, w_t)$ is on the negative side, then we have:

$$W^T V(1, w_t) > 0 W^T V(2, w_t) < 0 \tag{6.9}$$

Thus, we have $W^T V(1, s) > W^T V(2, s)$ and thus W is a 14-dimensional vector representing a feasible payoff function satisfying 6.9.

Going back to the general case, let $\mathcal{E}$ be the function that takes a given game history as input and returns as output, the labeling and encoding of points as defined above. We now describe the relationship between SVM classifier applied to points given by mapping V and the system of inequalities given by Equation 6.6 with the help of the following two propositions.

Proposition 4. *The system of inequalities given in Equation 6.6 is feasible for a game history H, i.e., we can find payoff functions such that all the inequalities are satisfied if the SVM algorithm can find a separator for encoding $\mathcal{E}(H)$.*

Proof. We note that for points on one side of decision surface, the value of decision surface is less than 0 and for the other side it is greater than 0. Therefore, if points encoded for the RHS are on the positive side and LHS on the negative side, Equations 6.6 are satisfied. Otherwise, we can flip sign of the decision function and achieve the same result. □

Proposition 5. *If the SVM algorithm can find a separator that misclassifies n_1 points with labels 1 and n_0 points with labels 0, then we can find payoff functions such that at most $n_0 + n_1$ of the inequalities given by Equation 6.6 are not satisfied.*

Proof. Without loss of generality, we assume that points encoding the RHS of Equation 6.6 are assigned positive labels and points encoding the LHS are assigned negative labels. A misclassified point LHS point can be assigned a decision value higher than the RHS point can lead to at most one violated constraint. Similarly, a misclassified

RHS point can lead to at most one violated constraint. Thus in all we can have at most $n_0 + n_1$ violated constraints. □

6.7 Implementation and Experiments

We implemented CBS, SCA, SVMM as well as the ICEL algorithm [134]. Sections 6.7.1 use synthetic data (with known payoff functions to evaluate these algorithms' accuracy). Section 6.8 uses the real-world MAROB data about 10 terrorist groups [10] with actions by both the terrorist groups and the government of the country involved. Section 6.8.1 uses a very fine-grained counter-terrorism data set with three actors: the terror group Lashkar-e-Taiba [122] which carried out the Mumbai attacks and the governments of Pakistan and India. The next two subsections compare the three algorithms presented in this chapter in order to identify which one is best - both from an accuracy and from a run-time perspective. Section 6.9 compares our best algorithm with theICEL algorithm. Because of Proposition 2, we could not apply ICEL to the synthetic data — and because the Lashkar-e-Taiba contained a host of environmental variables, we could not apply ICEL to that either. We used the MAROB data to compare ICEL and our SVMM metric.

6.7.1 Experiments on Synthetic Data

Generation of Synthetic Data

We wrote R code to generate random games with random linear payoffs functions and a random state of the world at each time. A payoff function is represented as a vector of coefficients of a linear function. Each of the payoff functions and the state of the world at each time point is a uniformly randomly directed positive vector of norm 1. After generating the payoffs and the state of the world at different times, an action history for all players is generated assuming best response. We are not simulating a game. Instead, each time point is a “what if” scenario, where each player is presented with a state of the world and they choose the best response as per their payoff functions. The code to generatef random games varies the following inputs:

np	Number of players
na	Number of actions for each player
n	Length of history for each game
noise	Probability that action of a player will be chosen uniformly at random (instead of the best response heuristic)
seed	The seed for random number generation (to ensure reproducibility of experiments)

The state of the world is thus an $(na * np)$-dimensional vector. Payoff for each action is a linear function of the state of the world and hence is represented as an $(na * np)$-dimensional vector of coefficients. Thus, each player has na such vectors. [3] We introduce noise into our experiments by allowing each player, at each time step, to either play a random response independently at random with probability given by parameter “noise”, or a best response to the current state of the world. To evaluate quality of payoffs learned, for each player, we learn the $np * na^2$ length vector of parameters of the player's payoff function. We measure the quality of our three payoff learning algorithms by comparing this vector with the actual payoff function vectors using Pearson Correlation Coefficients (PCCs for short).

Performance of SVM based method

We use a linear soft margin SVM classifier using the R interface to libsvm [22]. The hyperparameter for tuning this SVM is the cost of misclassification C. We tried values of $C \in \{0.01, 0.1, 1, 10, 100\}$ and chose the best-forming SVM model. However, we also report the overall results (encompassing all five values of C). The choice of C turns out to be not critical to the performance of our algorithms. *SVMM performs very well with median PCC above* 0.8 *for games with* 5 *players and* 5 *actions for each player and median PCC between* 0.6 *and* 0.8 *for most of the smaller games.* In addition,

[3]For most experiments on synthetic data, we have $na = np = 3$. Thus, each payoff function is a 9 dimensional vector. As there are 3 payoff functions per player (one per action), we are trying to learn a total of 9 vectors, each of which is 9-dimensional.

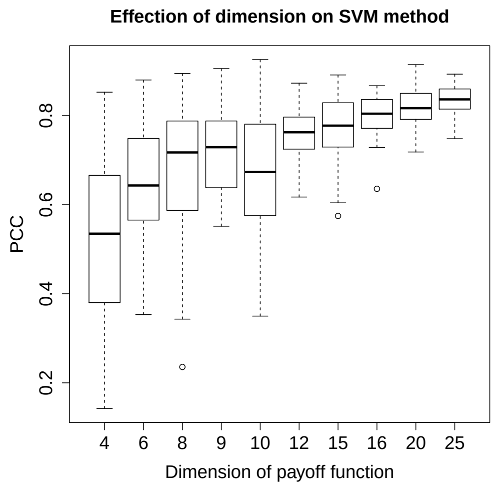


Figure 6.1: Effect of dimension of payoff function on performance of SVM method for synthetic data

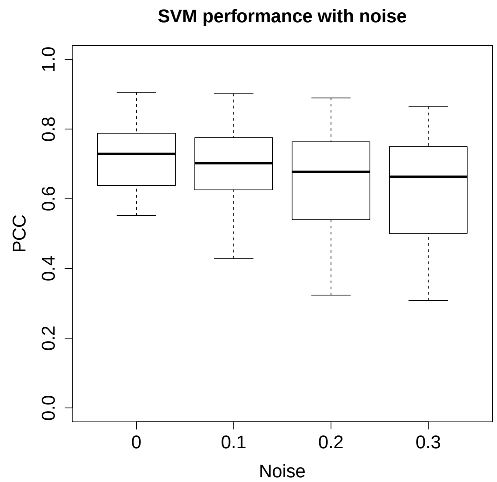


Figure 6.2: Effect of noise on performance of SVM method for synthetic data

performance degrades slowly with noise. We now analyze SVMM's performance in more detail.

Effect of Dimension of Payoff Function (SVM) Figure 6.1 shows the effect of dimension of the payoff function on performance of SVMM.[4] Here, the number of sample history points is 1000 and the noise parameter is set to 0. Surprisingly, the performance improves with dimension of the payoff function.

Effect of Noise (SVM) Figure 6.2 shows the effect of noise on SVMM's performance. The number of samples is 1000 and the dimension of the payoff function is 9. We note that performance degrades gracefully under noise. For zero noise, the median PCC value is 0.73, whereas even with noise as high as 0.3 (i.e., with probability 0.3 a player chooses to play a random action instead of the best response), we still get a

[4]The figure was plotted using the standard "boxplot" function in R (http://www.r-bloggers.com/boxplots-and-beyond-part-i/). The boxes denote the range of 25^{th} and the 75^{th} percentiles. The line in the box is the median. The upper and lower lines outside the box is the "nominal" range of values and the circles are outliers.

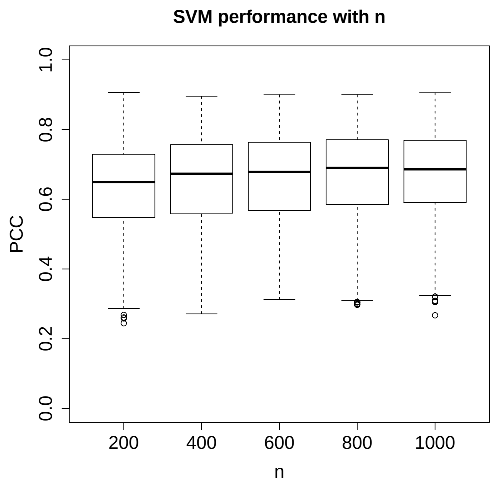


Figure 6.3: Effect of history length on performance of SVM method for synthetic data

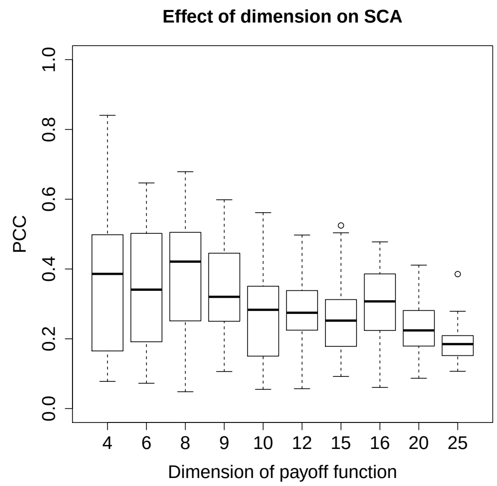


Figure 6.4: Effect of dimension of payoff function on performance of SCA for synthetic data

median PCC of 0.66.

Effect of Length of History (n) (SVM) Figure 6.3 shows the effect of the length of the history on SVMM's performance. There is slight improvement in median PCC as n increases from 200 to 1000.

Performance of SCA

In this section we go into details of performance of the SCA method and show that is is inferior to SVMM.

Effect of Dimension of Payoff Function (SCA) Figure 6.4 shows the effect of dimension of the payoff function on SCA's performance. The number of sample history points is 1000 and the noise parameter is set to 0. Performance degrades with dimension of the payoff function.

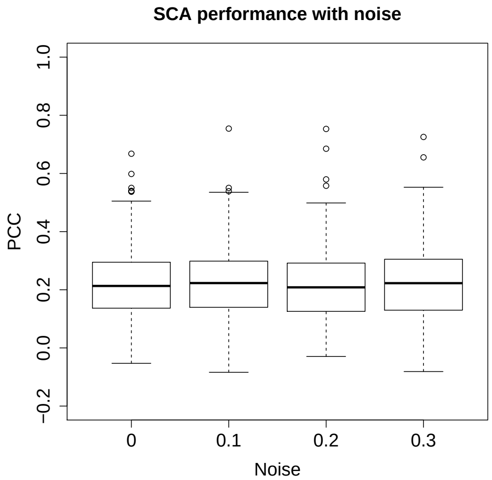


Figure 6.5: Effect of noise on performance of SCA for synthetic data

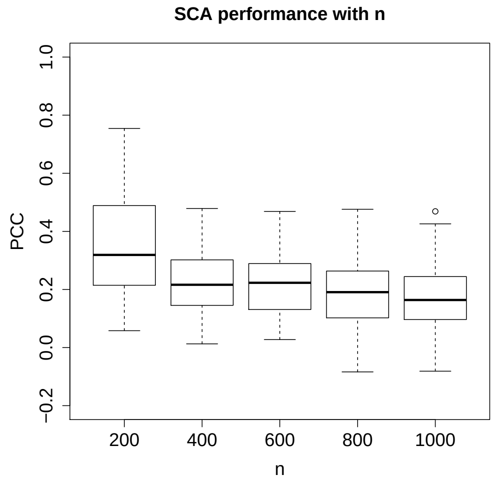


Figure 6.6: Effect of history length on performance of SCA for synthetic data

Effect of Noise (SCA) Figure 6.5 shows the effect of noise on SCA's performance. The number of samples is 200 and dimension of the payoff function is 9. While performance does not degrade significantly with noise, overall performance is poor (Overall PCC median of 0.22).

Effect of History length (SCA) Figure 6.6 shows the effect of history length on SCA's performance. Here noise is 0 and dimension of the payoff function is 9. Somewhat counter-intuitively, performance degrades with history length. This could be because the number of slack variables increases linearly with the length of history. Thus, the degrees of freedom of the model is potentially higher with a longer history and thus a longer history can lead to overfitting.

Performance of Centroid Based Method

In this section we study CBS's performance and show that it is far inferior than SVMM.

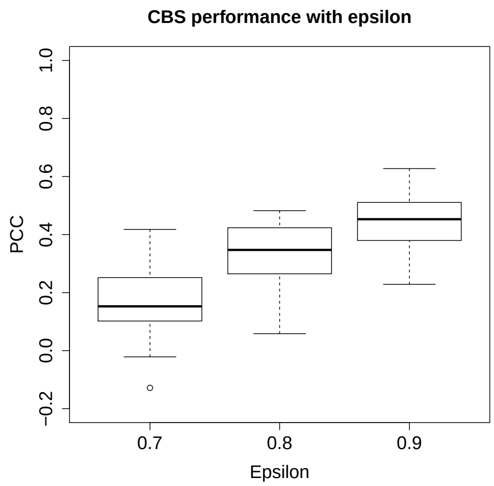


Figure 6.7: Effect of epsilon on performance of CBS for synthetic data

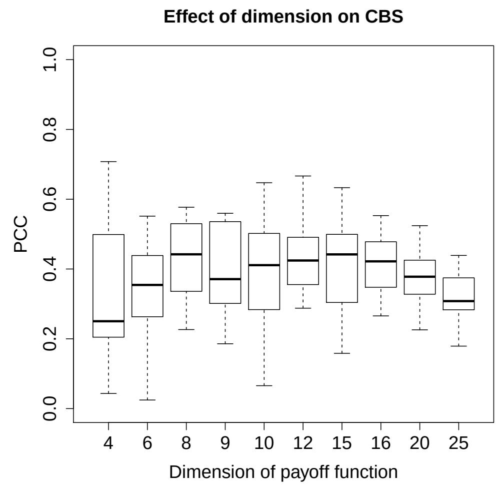


Figure 6.8: Effect of dimension of payoff function on performance of CBS for synthetic data

Effect of ϵ (CBS) Figure 6.7 shows the effect of ϵ on CBS's performance. Here, the length of history is 200 and dimension of the payoff function is 9. The noise is 0. Overall performance is better than SCA but much worse than the SVM based method (PCC median of 0.45 for $\epsilon = 0.9$).

Effect of Dimension of Payoff Function (CBS) Figure 6.8 shows the effect of dimension of the payoff function on performance of CBS. Here, the number of sample history points is 1000 and the noise parameter is set to 0. Performance shows no discernible trend.

Effect of Noise Figure 6.9 shows the effect of noise on performance of CBS. Here, the number of samples is 1000 and dimension of the payoff function is 9. Performance degrades sharply with noise and even with 10% noise, is close to random.

Effect of Length of History (CBS) Figure 6.10 shows the effect of history length on performance of CBS. Here noise is 0 and dimension of the payoff function is 9.

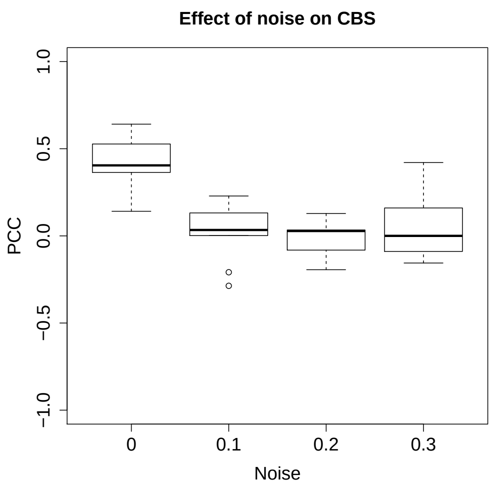


Figure 6.9: Effect of noise on performance of CBS for synthetic data

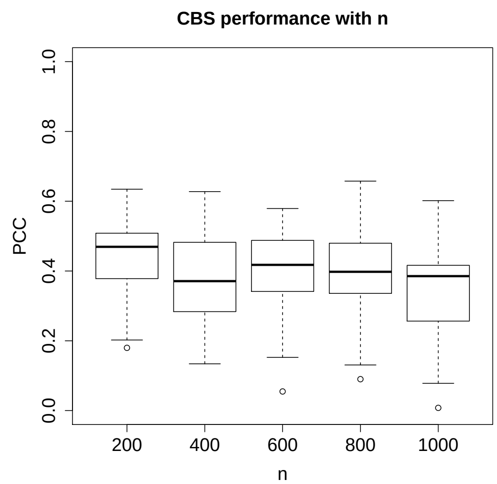


Figure 6.10: Effect of history length on performance of CBS for synthetic data

Performance does not improve with n and shows no discernible trend.

6.7.2 Runtime Comparison of CBS, SCA, SVMM

Figure 6.11 shows the relative runtime performance of the three methods for varying lengths of histories. SVMM is faster than the SCA by an order of magnitude and faster than the CBS by 2-3 orders of magnitude. For this comparison na and np are 3. Actual performance time of SVMM for varying values of n, na and np are given in Figure 6.12.

6.7.3 Discussion of the Results

Effect of Dimension of Payoff Function The effect of the dimension of the payoff functions on the performance of the SVMM, SCA and CBS is depicted in Figures 6.1, 6.4 and 6.8 respectively. We see that SVMM's performance improves with dimension. This is counter-intuitive as performance of most classifiers degrades with dimension. However, for the payoff inference problem, the number of constraints and hence the

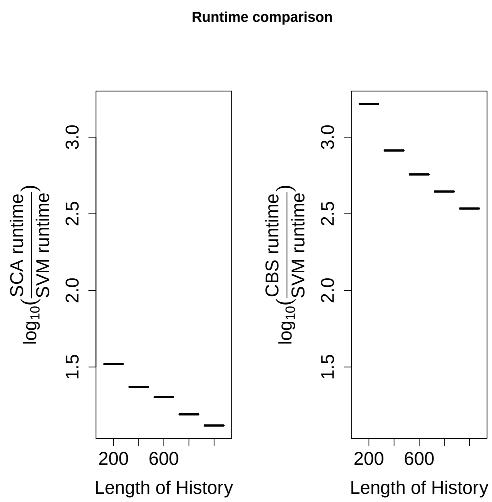


Figure 6.11: Comparison of runtimes of various methods for synthetic data

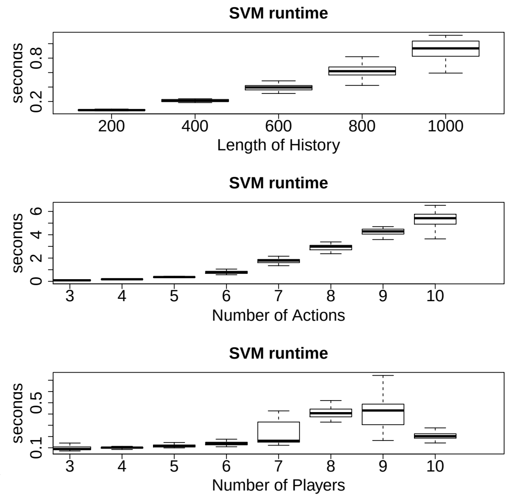


Figure 6.12: Runtime for the SVM method

number of points increases with the payoff function's dimension (for a fixed length of history). Thus, while the complexity of the classifier increases with dimension of data, we have more data to learn from and hence, performance improves with dimension. In case of SCA, we see that performance degrades with dimension. For SCA the number of slack variables is the product of history length and dimension of the payoff function. Thus, the increase in dimension leads to an increase in number of the slack variables. We hypothesize that in SCA this increase in number of slack variables with increase in dimension leads to performance degradation. CBS shows no discernible trend with increasing dimension. First, we are only approximating the centroid. Second, the centroid is very sensitive to individual constraints. Thus, CBS chooses an approximation to a feasible representative solution that is very sensitive to individual constraints. Therefore, it is not surprising that its performance is erratic.

Effect of Noise The effect of noise on the performance of the SVMM, SCA and CBS is depicted in Figures 6.2, 6.5 and 6.9 respectively. SVMM's performance degrades gracefully with noise. As oft margin SVMs evolved from hard margin SVMs to handle misclassification, this graceful degradation is expected. SCA's performance remains more or less constant and very poor with and without noise. SCA accommodates noisy points by allowing constraints to be violated by allowing negative slack variables and hence some robustness to noise is expected. However, a total lack of trend is a bit surprising. As noted earlier, centroid is very sensitive to individual constraints and hence extreme sensitivity to noise, as depicted in Figure 6.9 is expected.

Effect of History Length The effect of history length on the performance of SVMM, SCA and CBS is depicted in Figures 6.3, 6.6 and 6.10 respectively. SVMM's performance improves slightly when n increases. Thus, SVMM learns a better classifier with more data. Surprisingly, for synthetic data, it seems that SVMM is able to learn a very good classifier even with $n = 200$. SCA again shows the trend of degrading performance with the increasing number of constraints. Performance of CBS is again erratic.

Runtime The runtimes of the SVMM, SCA and CBS are compared in Figure 6.11. CBS is easily the worst. SVMM runtime increases with length of the history (Figure 6.12 because the problem size varies linearly with the length of the history. The corresponding increase with the number of actions is faster as the problem size varies quadratically with the number of actions. SVMM's runtime increases with the number of players in general, however, when the number of players is 10 it suddenly drops. We don't have a good explanation for this behavior and it is left for future work.

We conclude by stating that of the three algorithms presented in this chapter, SVMM achieves significantly larger accuracy than SCA and CBS, is more robust to noise, and performs much better and faster than the other two methods. It gives excellent performance on relatively large games (Median PCC above 0.8 *for games with* 5 *players,* 5 *actions per player).*

6.8 Experiments on Marob Data Set

We ran tests on all 10 terror groups in the Minorities at Risk Organizational Behavior (MAROB) [10] data set for which at least 20 rows of data are available. We aggregated low level MAROB actions (both by the group and the government of the nation where the group is based) into high level actions. The high level actions involved two actions each for the group (political engagement with the government, militant activities) and for the government (political engagement with the group and suppression of the group). Each of these actions can be carried out at low, medium, high levels. Hence, each player can take one of 9 actions, leading to 81 total joint actions. We ran experiments with data about 10 group/nation pairs. To test validity of the payoffs learned, we made predictions of actions of terror groups and governments and checked the accuracy of these predictions. The mean number of actions for government player, denoted by G is 4.1 and the mean for terro organization, denoted TO is 5.6. The mean number of joint actions (product of actions of G and TO) is 25.30. The history length of each game is between 20 and 25.

Working on this data is problematic for two reasons: (i) we only have 20 data points for each group/nation pair, (ii) we don't know the ground truth. We evaluate the quality of the learned payoffs as follows. We compute the Spearman Rank Correlation Coefficient (SCC) correlation between predicted payoffs and the binary vector representing actual actions performed during the time period. While the payoffs are reals in $[0, 1]$, we are correlating them with binary variables in $\{0, 1\}$, thus even in the best case, we cannot expect the correlation to be 1, e.g. for 5 actions (about average for the games in MAROB dataset), a point in history may be $(1, 0, 0, 0, 0)$. It will be correctly predicted by a payoff vector such as $p = (1, p_2, p_3, p_4, p_5), p_i < 1 \; \forall i \in \{2..5\}$. Assuming that $p_i \neq p_j \, \forall i, j \in \{1 \dots 5\}$, the expected SCC for this data (assuming p_i's uniformly distributed over $[0, 1)$) is 0.71 which is quite good, given the paucity of data. Figure 6.13 gives the best expected SCC as a function of the number of actions of a player.

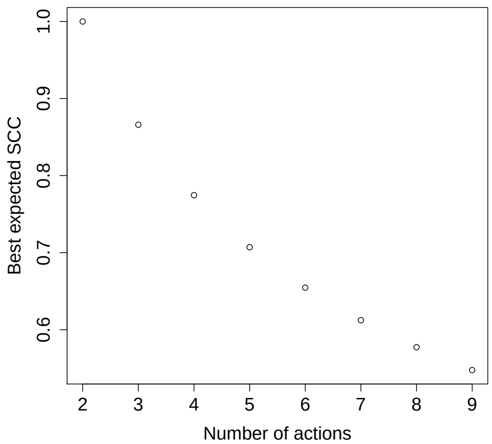


Figure 6.13: Best expected SCC

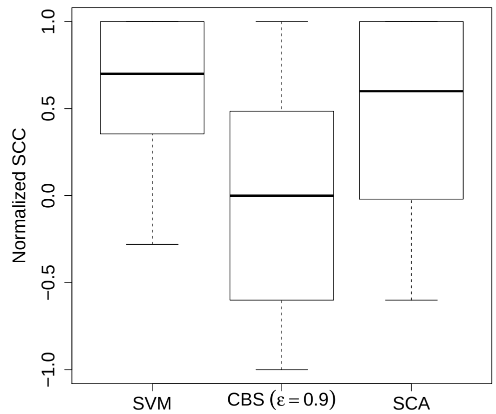


Figure 6.14: Comparitive performance of the three methods on MAROB data set

Depending on the number of actions of a player, we normalize the SCC between predicted payoffs and the actual binary action vector to arrive at normalized SCC (NSCC).

Comparison of performance of the three methods (Marob) Figure 6.14 compares CBS, SCA, and SVMM. On this dataset, SVMM uses the radial basis function as the kernel and the model is selected based on leave-one-out cross validation. Here again, SVMM performs well (median NSCC=0.7) and comfortably outscores SCA (median NSCC=0.6) and CBS (median NSCC=0). While in for some part of the data set CBS does do well with 75^{th} percentile NSCC value close to 0.4, the average CBS performs particularly poorly, with a median NSCC of close to 0 indicating near random performance. SCA performs well but not as well as SVMM.

As in the case of synthetic data (Figure 6.9) CBS is very sensitive to noise. SVMM and SCA perform well becausethey both allow some constraints to be violated. As shown in Figures 6.4 and 6.6, SCA's performance degrades with the number of constraints. However, since all the histories of in the MAROB data set are short (20), SCA performs quite well. SVMM performs even better than SCA. We hypothesis that this is because SVMM accommodates non-linear payoff functions using kernel methods and SCA allows only linear payoffs. We believe that real players' payoffs functions are not necessary linear.

6.8.1 LeT Experiments

We conducted extensive tests on the Lashkar-e-Taiba (LeT) dataset[122] which contains 252 rows (months) of data about $700+$ variables.[5] We choose 24 variables which we considered relevant to our problem. These variables include six variables for various types of attacks carried out by LeT and eight for actions taken by the Pakistani government and military. The other 10 variables such as existence of international ban, existence of conflict within LeT, split in LeT etc. are treated as *environmental variables*.[6]

The LeT dataset models a relatively big game. Players can take many actions simultaneously. If we encode each combination of actions as a separate action, we will end up with 64 actions for LeT and 256 actions for Pakistani government. This leads to a very high dimensional encoding for this dataset. As an illustration, for a the given history, H of this game, $\mathcal{E}(H)$ for Pakistani government would be $256{*}(1{+}336) = 86272$ dimensions (ignoring the environment variables). We now describe how we tackle the dimensionality problem.

[5] It includes details about attacks carried out by LeT, communications campaigns and rallies organized by LeT. It also includes actions by the state (Pakistan) and international actors (US, India, EU etc.) such as arrests, tribunals, killings related to members of LeT.

[6] Environment variables can be seen to be actions of another player (similar to the "nature" player in classical game theory), whose actions we cannot predict (or are not interested in predicting). Nevertheless, these actions do have an effect on payoffs of other players.

Independent Payoffs for Simultaneous Actions

One natural way to reduce dimensions of the embedding is to assume that payoffs for simultaneous actions are independent and additive. However, different actions may require different level of effort for the player and it is reasonable to assume that payoffs are proportional to effort. For example, if attacking a security installation requires double the effort of attacking civilian transport, then the payoffs for the two actions are comparable only if the payoff from attacking the security installation is double the payoff from attacking civilian transport. This is because capability and resources of an organization are limited and thus, to maximize the payoff, effort should be spent on actions that give maximum payoff for each unit of effort. Therefore, for this approach, we need to assign effort-based weights to players' actions. However, there is no reliable way of knowing how much effort was needed for each action of the player. Therefore, we reject this approach. Instead, we relax the constraints in Equation 6.6 by assuming that regret for each action actually played at time t is greater than or equal to regret for actions not played at time t. For example, assume that at time t a player played actions $(0, 1, 0, 1, 0, 0)$ indicating that they took actions 2 and 4 out of possible actions in $\{1, \ldots, 6\}$. Then, we assume that regrets for actions 2 and 4 were higher than regrets for other actions at time t. The other alternative could have been encoding each of the possible combinations of actions as a separate action, which leads to 64 possible distinct actions at each time step.

Evaluation of Quality of Learned Payoffs

We evaluate the quality of prediction in two ways. First, we compute the SCC of events and payoffs for each time period from the test data. We compute correlation between predicted payoffs for actions and the binary vector representing actual occurrence of the events during the time period. We use this method to compare the performance of SVMM, CBS and SCA. Second, for SVMM, we compute the quality of predictions

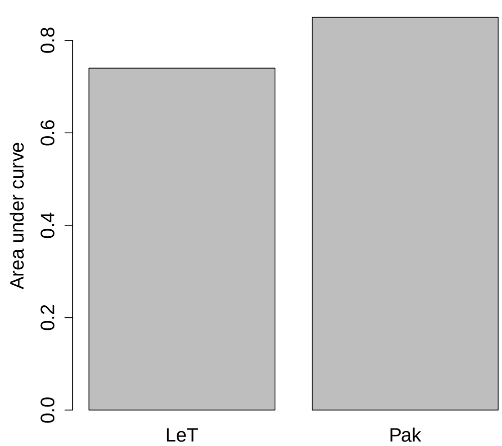


Figure 6.15: Predictive performance of SVMM on LeT dataset

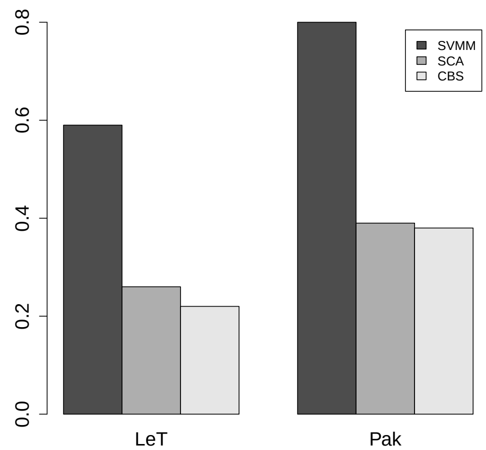


Figure 6.16: Performance of SVMM, SCA and CBS on LeT dataset

using Area Under the RoC curve as our metric. We don't use predictions to evaluate CBS and SCA methods because these methods don't extend naturally to prediction and prediction is not our primary objective.

Comparison of the Methods

Figure 6.16 presents comparative performance of the three methods on the LeT dataset. Again, SVMM performs well and clearly outperforms SCA and CBS. The NSCC for SVMM for Player LeT in the dataset is 0.59. The NSCC for SVMM for Player Pakistan is 0.80. The predictive performance of SVMM is good with area under single point RoC curve of 0.74 and 0.85 for LeT and Pakistan respectively (Figure 6.15).

The performance of SVMM is much better than the other two methods. We think that the reasons are three-fold. First, SVMM is robust to noise. Second, SVMM allows for non-linear payoff functions. Third, SVMM performs better with more constraints and higher dimensional payoff functions.

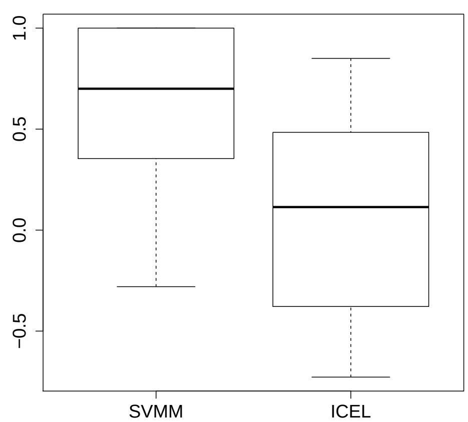


Figure 6.17: Comparative performance of ICEL and SVMM

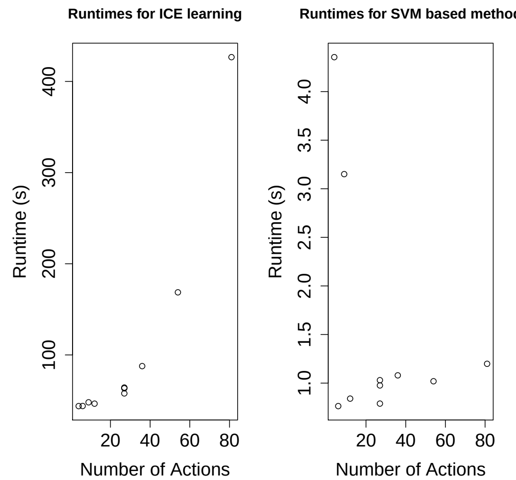


Figure 6.18: Comparative runtimes for ICEL and SVMM

6.9 Comparison with ICEL

We compare the performance of our best algorithm (SVMM) against the *Inverse Correlated Equilibrium Learning (ICEL)* algorithm of [134]. ICEL assumes that players play a correlated equilibrium. Input to the ICEL algorithm is the game history and corresponding outcomes, which depend on joint actions of the players. The output is a joint distribution over the player's actions. The learned distribution is a Correlated Equilibrium for the corresponding inferred payoffs.

We evaluated the performance of ICEL against SVMM using NSCC metric on the MAROB dataset. We could not evaluate ICEL on LeT dataset because it has environment variables in addition to player actions and ICEL is not applicable to games with environmental variables. If the payoffs learned by ICEL correspond to a Correlated Equilibrium actually played by the players, we can expect good rank correlation between chosen actions and payoffs. However, as can be seen from Figure 6.17, this is not the case. The median NSCC is 0.1140 over all games (which suggests that ICEL is only marginally better than random noise). NSCC is above 0.5 in only 3 out of 20 instances (10 games, 2 players per game). In comparison, SVMM performs much better with median NSCC of 0.7.

We believe the poor performance of ICEL stems from two factors. First, we have no knowledge of the outcomes, only of game history. We empirically observed that in the absence of any information about the outcomes, the ICEL convex program simply converges to distribution of actions actually played by the players (mean Kullback-Leibler divergence between actual and learned distributions is 0.043, max 0.088). Thus, in effect, the method corresponds to predicting that whatever happened in past will happen in future with no notion of recency and dynamics. Secondly, in our real-world data, the players may not play a correlated equilibrium and our proposed dynamics, which are based on regret minimization and recency, may be a closer approximation of reality.

The runtime comparison is shown in Figure 6.18. Here again, SVMM is 1-2 orders of magnitude faster than ICEL. However, this is a bit of apples and oranges comparison as the SVMM code is in R with a Libsvm backend and the ICEL code uses the Python code provided by authors publicly at their website. We note that ability to use highly optimized, stable and mature libraries provided by machine learning community for classification tasks is one of the advantages of our approach over other extant approaches.

6.10 Conclusion

In this work, we have developed, for the first-time, a method to infer payoffs for real-world games, under much more reasonable assumptions than past work. Specifically, unlike much past work, PIE is applicable to multi-player games, allows players to not be fully rational, does not assume a coordination mechanism, does not assume a symmetric game and is scalable, while other works are lacking in at least one of these aspects. Moreover, we apply our theory to real-world strategic games with a real world dataset based on a widely influential previous study[122] that was briefed to two national security advisors and heads of multiple security agencies around the world. Such individuals seek *understanding* - and our goal in learning these payoffs were to facilitate explaining the payoffs to such senior decision makers - rather than prediction. Toward this end, we propose three heuristics that may be used to learn payoffs of players in *multi-player* real-world games including one that builds upon Support Vector Machines - a tested technique in data mining that has never been used before for learning payoffs. Though the goal of this work is not prediction, we test our methods in three ways. We use a synthetic data set and a well-known terrorism data set[10] to see how well we can predict known payoff functions (synthetic data) and actions ([10] data). Even though we have small amounts of data in both cases, they are bigger than those in previous studies, and our best algorithm (SVMM) achieves good correlations. Our third test looks at 10 years of data about the terror group Lashkar-e-Taiba (responsible for the 2008 Mumbai attacks). We show that SVMM is both faster and much more accurate than ICEL [134] one of the best prior algorithms in the literature. Much work remains to be done. Even though PIE is more scalable than past work, there is still a long way to go. And explaining learned payoff functions to real world decision makers also has many challenging aspects that deserve much future study.

Chapter 7

Future Work

We conclude this work with a brief overview of potential future research directions related to our work. In absence of symmetry introduced by assumption of uniform thresholds, the problem is notoriously hard to approximate [39] and that too for case of a simple diffusion model (Independent Cascade). To make this research relevant to diffusion processes in real-world social networks, there is a need to extend our algorithms to the case where more information about individual nodes can be incorporated into the model. Ideally, these approaches also need to be extended to various diffusion models. One work in this directions is the work on diffusion centrality [72]. However, this work is without the clean theoretical guarantees provided by our algorithms. Thus, modeling diffusion of influence in social networks using models that balance technical tractability with real-world applications is still somewhat of an open problem.

Another productive direction of research can be unification of various strategic resource allocation problems being studied in a single framework. Network Cournot Competition is a very flexible model for strategic resource allocation. Intuitively, there are important similarities between the linear Arrow-Debreu market model, Fisher market model and Network Cournot Competition market model. Thus, studying all these related problems under a single model is an exciting possibility.

With regards to Payoff Inference problem, we make certain assumptions about dynamics of the game. Incorporation different dynamics into this framework is a direction for future research. Another interesting possibility is incorporation and learning of players belief about each other into the game play dynamics.

Bibliography

[1] Melika Abolhassani, Mohammad Hossein Bateni, MohammadTaghi Hajiaghayi, Hamid Mahini, and Anshul Sawant. Network cournot competition. In *WINE '14*, pages 15–29. Springer, 2014.

[2] Bharat Adsul, Jugal Garg, Ruta Mehta, and Milind Sohoni. Rank-1 bimatrix games: a homeomorphism and a polynomial time algorithm. In *STOC '11*, pages 195–204. ACM, 2011.

[3] N. AhmadiPourAnari, S. Ehsani, M. Ghodsi, N. Haghpanah, N. Immorlica, H. Mahini, and V. S. Mirrokni. Equilibrium pricing with positive externalities. In *WINE '10*, pages 424–431. Springer-Verlag, 2010.

[4] Hessameddin Akhlaghpour, Mohammad Ghodsi, Nima Haghpanah, Vahab S Mirrokni, Hamid Mahini, and Afshin Nikzad. Optimal iterative pricing over social networks. In *WINE '10*, pages 415–423. Springer, 2010.

[5] Ingo Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and its Applications*, 199:339–355, 1994.

[6] Rabah Amir. Cournot oligopoly and the theory of supermodular games. *Games and Economic Behavior*, 1996.

[7] Simon P. Anderson and Régis Renault. Efficiency and surplus bounds in cournot competition. *Journal of Economic Theory*, 2003.

[8] David Arthur, Rajeev Motwani, Aneesh Sharma, and Ying Xu. Pricing strategies for viral marketing on social networks. In *WINE '09*, pages 101–112. Springer, 2009.

[9] W. Brian Arthur. Competing technologies, increasing returns, and lock-in by historical events. *The Economic Journal*, 99(394):pp. 116–131, 1989.

[10] Amy Pate Asal, Victor and Jonathan Wilkenfeld. Minorities at risk organizational behavior data and codebook version 9/2008 online, September 2008. URL `http://www.cidcm.umd.edu/mar/data.asp`.

[11] Jai Asundi, Octavian Carare, and Kutsal Dogan. Competitive implications of software open-sourcing. *Decision Support Systems*, 54(1):153 – 163, 2012.

[12] David Avis, Gabriel D. Rosenberg, Rahul Savani, and Bernhard von Stengel. Enumeration of nash equilibria for two-player games. *Economic Theory*, 42(1): 9–37, 2010.

[13] Moshe Babaioff, Brendan Lucier, and Noam Nisan. Bertrand networks. In *EC '13*, pages 33–34, New York, NY, USA, 2013. ACM.

[14] Abhijit V. Banerjee. A simple model of herd behavior. *The Quarterly Journal of Economics*, 107(3):797–817, August 1992.

[15] J. Bertrand. Book review of théorie mathématique de la richesse sociale and of recherches sur les principes mathématiques de la théorie des richesses. *Journal de Savants*, 67:499–508, 1883.

[16] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.

[17] Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100:992–1026, 1992.

[18] Christian Borgs, Michael Brautbar, Jennifer T. Chayes, and Brendan Lucier. Maximizing social influence in near optimal time. *CoRR*, abs/1212.0884, 2012.

[19] Peter E.M. Borm, S.H. Tijs, and J. van den Aarssen. Pareto equilibria in multiobjective games. Technical report, Tilburg University, 1988.

[20] Stephen Poythress Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[21] F. A. Campos, J. Villar, and J. Barquín. Solving cournot equilibriums with variational inequalities algorithms. *Generation, Transmission & Distribution, IET*, 4(2):268–280, 2010.

[22] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2 (3):27, 2011.

[23] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *KDD '09*, pages 199–208. ACM, 2009.

[24] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD '10*, pages 1029–1038. ACM, 2010.

[25] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM '10*, pages 88–97. IEEE, 2010.

[26] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincon, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In *ICDM '11*, pages 379–390, 2011.

[27] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *FOCS '06*, 2006.

[28] Flavio Chierichetti, Jon Kleinberg, and Alessandro Panconesi. How to schedule a cascade in an arbitrary graph. In *EC '12*, pages 355–368. ACM, 2012.

[29] Caspar G Chorus. A new model of random regret minimization. *EJTIR*, 2(10), 2010.

[30] Ryan Clarke. *Lashkar-i-Taiba: The Fallacy of Subservient Proxies and the Future of Islamist Terrorism in India*. Strategic Studies Institute, US Army War College, 2010.

[31] Michael Conlin and Vrinda Kadiyali. Entry-deterring capacity in the texas lodging industry. *Journal of Economics & Management Strategy*, 15(1):167–185, 2006.

[32] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[33] Constantinos Daskalakis. On the complexity of approximating a nash equilibrium. In *SODA '11*, pages 1498–1517. SIAM, 2011.

[34] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. In *STOC '06*, pages 71–78. ACM, 2006.

[35] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. A note on approximate nash equilibria. In *WINE '06*, pages 297–306. Springer, 2006.

[36] Andrew F. Daughety. *Cournot Oligopoly: Characterization and Applications*. Cambridge University Press, 2005.

[37] Christopher J. Day, Benjamin F. Hobbs, and Jong-Shi Pang. Oligopolistic competition in power networks: a conjectured supply function approach. *Power Systems, IEEE Transactions on*, 17(3):597–607, 2002.

[38] B. Bueno de Mesquita. *The Predictioneer's Game: Using the Logic of Brazen Self-Interest to See and Shape the Future*. Random House, 2010.

[39] Erik D. Demaine, Mohammad T. Hajiaghayi, Hamid Mahini, David L. Malec, S. Raghavan, Anshul Sawant, and Morteza Zadimoghadam. How to influence people with partial incentives. In *WWW '14*, pages 937–948. International World Wide Web Conferences Steering Committee, 2014.

[40] Nikhil R. Devanur and Ravi Kannan. Market equilibria in polynomial time for fixed number of goods or agents. In *FOCS '08*, pages 45–53. IEEE, 2008.

[41] Nikhil R. Devanur and Vijay V. Vazirani. An improved approximation scheme for computing arrow-debreu prices for the linear case. In *FST TCS '03*, pages 149–155. Springer, 2003.

[42] Nikhil R. Devanur, Christos H. Papadimitriou, Amin Saberi, and Vijay V. Vazirani. Market equilibrium via a primal-dual-type algorithm. In *FOCS '02*, pages 389–395. IEEE, 2002.

[43] John P. Dickerson, Aaron Mannes, and V. S. Subrahmanian. Dealing with Lashkar-e-Taiba: A multi-player game-theoretic perspective. In *International Symposium on Open Source Intelligence and Web Mining (OSINT-WM)*, pages 354–359, 2011.

[44] John P. Dickerson, Anshul Sawant, Mohammad T. Hajiaghayi, and V. S. Subrahmanian. Preve: a policy recommendation engine based on vector equilibria applied to reducing let's attacks. In *ASONAM '13*, pages 1084–1091. ACM, 2013.

[45] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *KDD '01*, pages 57–66. ACM, 2001.

[46] Shayan Ehsani, Mohammad Ghodsi, Ahmad Khajenezhad, Hamid Mahini, and Afshin Nikzad. Optimal online pricing with network externalities. *Information Processing Letters*, 112(4):118 – 123, 2012.

[47] Edmund Eisenberg and David Gale. Consensus of subjective probabilities: The pari-mutuel method. *The Annals of Mathematical Statistics*, 30(1):165–168, 1959.

[48] Walter Enders and Todd Sandler. Terrorism: Theory and applications. *Handbook of Defense Economics*, 1:213–249, 1995.

[49] EU Commission Staff Working Document. Annex to the green paper: A European strategy for sustainable, competitive and secure energy. Brussels, 2006.

[50] Tomas Feder, Hamid Nazerzadeh, and Amin Saberi. Approximating nash equilibria using small-support strategies. In *EC '07*, pages 352–354. ACM, 2007.

[51] Amos Fiat, Elias Koutsoupias, Katrina Ligett, Yishay Mansour, and Svetlana Olonetsky. Beyond myopic best response (in cournot competition). In *SODA '12*, pages 993–1005. SIAM, 2012.

[52] Dean P. Foster and Rakesh V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1):40–55, 1997.

[53] Steven A Gabriel, Supat Kiet, and Jifang Zhuang. A mixed complementarity-based equilibrium model of natural gas markets. *Operations Research*, 53(5): 799–818, 2005.

[54] Mehdi Ghiyasvand and James B. Orlin. A simple approximation algorithm for computing arrow-debreu prices. *Operations Research*, 60(5):1245–1248, 2012.

[55] Sanjeev Goyal and Michael Kearns. Competitive contagion in networks. In *STOC '12*, pages 759–774. ACM, 2012.

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

[57] Dilek Gunnec. *Integrating Social Network Effects in Product Design and Diffusion*. Ph.d. thesis, University of Maryland, College Park, MD, USA, 2012.

[58] Dilek Gunnec and S. Raghavan. Integrating social network effects in the share-of-choice problem. Technical report, University of Maryland, College Park, MD 20742, 2012.

[59] Dilek Gunnec, S. Raghavan, and Rui Zhang. The least cost influence problem. Technical report, University of Maryland, College Park, MD 20742, 2013.

[60] Jonas Häckner. A note on price and quantity competition in differentiated oligopolies. *Journal of Economic Theory*, 93(2):233–239, 2000.

[61] Mohammad T. Hajiaghayi, Hamid Mahini, and Anshul Sawant. Scheduling a cascade with opposing influences. In *SAGT '13*, pages 195–206. Springer Berlin Heidelberg, 2013.

[62] Jason Hartline, Vahab Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *WWW '08*, pages 189–198, New York, NY, USA, 2008. ACM.

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

[64] Harold Hotelling. *Stability in Competition*. Springer, 1990.

[65] Nathanael Hyafil and Craig Boutilier. Regret minimizing equilibria and mechanisms for games with strict type uncertainty. In *UAI*, pages 268–277. AUAI Press, 2004.

[66] Rahmi Ilkılıç. Cournot competition on a network of markets and firms. Technical report, Fondazione Eni Enrico Mattei, 2009.

[67] Nicole Immorlica, Evangelos Markakis, and Georgios Piliouras. Coalition formation and price of anarchy in cournot oligopolies. In *WINE '10*, pages 270–281. Springer, 2010.

[68] Kamal Jain. A polynomial time algorithm for computing an arrow-debreu market equilibrium for linear utilities. *SIAM Journal on Computing*, 37(1):303–318, 2007.

[69] Wei Jing-Yuan and Yves Smeers. Spatial oligopolistic electricity models with cournot generators and regulated transmission prices. *Operations Research*, 47 (1):102–112, 1999.

[70] Wilson John. *Caliphate's Soldiers: The Lashkar-e-Tayyeba's Long War*. Amaryllis Press and the Observer Research Foundation, 2011.

[71] Shankar Kalyanaraman and Christopher Umans. Algorithms for playing games with limited randomness. In *ECCC '07*, pages 323–334. Springer, 2007.

[72] Chanhyun Kang, Cristian Molinaro, Sarit Kraus, Yuval Shavitt, and VS Subrahmanian. Diffusion centrality in social networks. In *ASONAM '12*, pages 558–564. IEEE Computer Society, 2012.

[73] Ravi Kannan and Thorsten Theobald. Games of fixed rank: a hierarchy of bimatrix games. In *SODA '07*, pages 1124–1132. SIAM, 2007.

[74] Ebrahim Karami and Savo Glisic. Optimization of wireless multi-source multicast ad-hoc networks using asymmetric matrix games. In *ICC'10*, pages 1–5. IEEE, 2010.

[75] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, pages 137–146. ACM, 2003.

[76] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diusion model for social networks. In *ICALP '05*, pages 1127–1138. Springer Verlag, 2005.

[77] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[78] Charles D. Kolstad and Lars Mathiesen. Computing cournot-nash equilibria. *Operations Research*, 39(5):739–748, 1991.

[79] Sarit Kraus, Jonathan Wilkenfeld, and Gilad Zlotkin. Multiagent negotiation under time constraints. *Artificial intelligence*, 75(2):297–345, 1995.

[80] David M. Kreps and Jose A. Scheinkman. Quantity precommitment and Bertrand competition yield Cournot outcomes. *The Bell Journal of Economics*, pages 326–337, 1983.

[81] Nikolai S. Kukushkin. *Cournot Oligopoly with" almost" Identical Convex Costs*. Instituto Valenciano de Investigaciones Económicas, 1993.

[82] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *KDD '07*, pages 420–429. ACM, 2007.

[83] Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *WSDM '13*, pages 657–666. ACM, 2013.

[84] Xiaomin Lin, Peter A Beling, and Randy Cogill. Multi-agent inverse reinforcement learning for zero-sum games. *arXiv preprint arXiv:1403.6508*, 2014.

[85] RHA Lindelauf, HJM Hamers, and BGM Husslage. Cooperative game theoretic centrality analysis of terrorist networks: The cases of jemaah islamiyah and al qaeda. *European Journal of Operational Research*, 229(1):230–238, 2013.

[86] Roy Lindelauf, Peter Borm, and Herbert Hamers. The influence of secrecy on the communication structure of covert networks. *Social Networks*, 31(2):126–137, 2009.

[87] Richard J. Lipton, Evangeloas Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *EC '03*, pages 36–41. ACM, 2003.

[88] John A. Major. Advanced techniques for modeling terrorism risk. *Journal of Risk Finance, The*, 4(1):15–24, 2002.

[89] Lina Mallozzi, Lucia Pusillo, and Stef Tijs. Approximate equilibria for bayesian games. *Journal of Mathematical Analysis and Applications*, 343(2):1098–1102, 2008.

[90] Aaron Mannes. Qualitative analysis & computational techniques for the counterterror analyst. *Handbook of Computational Approaches to Counterterrorism*, pages 83–98, 2013.

[91] Lars Mathiesen. Computation of economic equilibria by a sequence of linear complementarity problems. In *Economic Equilibrium: Model Formulation and Solution*, pages 144–162. Springer, 1985.

[92] Tomasz P Michalak, Talal Rahwan, Oskar Skibski, and Michael Wooldridge. Defeating terrorist networks with game theory. *IEEE Intelligent Systems*, (1): 53–61, 2015.

[93] Paul Milgrom and John Roberts. Rationalizability, learning, and equilibrium in games with strategic complementarities. *Econometrica: Journal of the Econometric Society*, 1990.

[94] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[95] Elchanan Mossel and Sébastien Roch. Submodularity of influence in social networks: From local to global. *SIAM J. Comput.*, 39(6):2176–2188, 2010.

[96] Sriraam Natarajan, Gautam Kunapuli, Kshitij Judah, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik. Multi-agent inverse reinforcement learning. In *ICMLA '10*, pages 395–400. IEEE, 2010.

[97] Gergely Neu and Csaba Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. *arXiv preprint arXiv:1206.5264*, 2012.

[98] Karsten Neuhoff, Julian Barquin, Maroeska G. Boots, Andreas Ehrenmann, Benjamin F. Hobbs, Fieke A. M. Rijkers, and Miguel Vazquez. Network-constrained cournot models of liberalized electricity markets: the devil is in the details. *Energy Economics*, 27(3):495–525, 2005.

[99] Aviv Nevo. Measuring market power in the ready-to-eat cereal industry. *Econometrica*, 69(2):307–342, 2001.

[100] Iris Nevo and Ido Erev. On surprise, change, and the effect of recent outcomes. *Frontiers in psychology*, 3, 2012.

[101] Andrew Y. Ng, Stuart J. Russell, et al. Algorithms for inverse reinforcement learning. In *ICML '00*, pages 663–670, 2000.

[102] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*, volume 1. Cambridge University Press Cambridge, 2007.

[103] Koji Okuguchi and Ferenc Szidarovszky. On the existence and computation of equilibrium points for an oligopoly game with multi-product firms. *Annales, Univ. Sci. bud. Roi. Fotvos Nom*, 1985.

[104] James B. Orlin. Improved algorithms for computing fisher's market clearing prices: computing fisher's market clearing prices. In *STOC '10*, pages 291–300. ACM, 2010.

[105] Martin J. Osborne and Carolyn Pitchik. Price competition in a capacity-constrained duopoly. *Journal of Economic Theory*, 38(2):238–260, 1986.

[106] Michael Ovelgönne, Chanhyun Kang, Anshul Sawant, and V. S. Subrahmanian. Covertness centrality in networks. In *ASONAM '12*, pages 863–870, 2012.

[107] Fatih Ozgul, Julian Bondy, and Hakan Aksoy. Mining for offender group detection and story of a police operation. In *AusDM '07*. Australian Computer Society, 2007.

[108] Amil Petrin. Quantifying the benefits of new products: The case of the minivan. Technical report, National Bureau of Economic Research, 2001.

[109] Karen Pittel and Dirk T. G. Rübbelke. Transitions in the negotiations on climate change: From prisoner's dilemma to chicken and beyond. *International Environmental Agreements: Politics, Law and Economics*, 12(1):23–39, 2012.

[110] J.S. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM Journal on Computing*, 15(3):694–702, 1986.

[111] Luis A Rademacher. Approximating the centroid is hard. In *SOCG '07*, pages 302–305, 2007.

[112] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02*, pages 61–70. ACM, 2002.

[113] Bruce O. Riedel. *Deadly Embrace: Pakistan, America, and the Future of the Global Jihad*. Brookings Institution Press, 2012.

[114] Everett M. Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[115] Todd Sandler and Walter Enders. An economic perspective on transnational terrorism. *European Journal of Political Economy*, 20(2):301–316, 2004.

[116] Anshul Sawant, John P. Dickerson, Mohammad T. Hajiaghayi, and V. S. Subrahmanian. Automated generation of counterterrorism policies using multiexpert input. *ACM Trans. Intell. Syst. Technol.*, 6(4):44:1–44:27, July 2015. ISSN 2157-6904.

[117] Thomas C. Schelling. *The Strategy of Conflict*. Harvard Univ Press, 1980.

[118] Thomas C. Schelling. *Micromotives and Macrobehavior*. WW Norton & Company, 2006.

[119] Lloyd S. Shapley. Equilibrium points in games with vector payoffs. *Naval Research Logistics Quarterly*, 6(1):57–61, 1959.

[120] Itamar Simonson. The influence of anticipating regret and responsibility on purchase decisions. *Journal of Consumer Research*, pages 105–118, 1992.

[121] Nirvikar Singh and Xavier Vives. Price and quantity competition in a differentiated duopoly. *The RAND Journal of Economics*, pages 546–554, 1984.

[122] V. S. Subrahmanian, Aaron Mannes, Amy Sliva, Jana Shakarian, and John P. Dickerson. *Computational Analysis of Terrorist Groups: Lashkar-e-Taiba*. Springer New York, 2012.

[123] Junichi Suzuki. Land use regulation as a barrier to entry: Evidence from the texas lodging industry*. *International Economic Review*, 54(2):495–523, 2013.

[124] Stephen Tankel. *Lashkar-e-Taiba: Past Operations and Future Prospects*. New America Foundation, 2011.

[125] Stephen Tankel. *Storming the World Stage: The Story of Lashkar-e-Taiba*. Columbia University Press, 2011.

[126] Thorsten Theobald. Enumerating the nash equilibria of rank-1 games. In *CRM Proceedings and Lecture Notes*, volume 48, 2009.

[127] Lars Thorlund-Petersen. Iterative computation of cournot equilibrium. *Games and Economic Behavior*, 2(1):61–75, 1990.

[128] Donald M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 1978.

[129] Haralambos Tsaknakis and Paul G. Spirakis. An optimization approach for approximate nash equilibria. In *WINE '07*, volume 4858, page 42. Springer, 2007.

[130] Thomas W. Valente. *Network Models of the Diffusion of Innovations (Quantitative Methods in Communication Series)*. Hampton Press (NJ)(January 10, 1995), 1995.

[131] Mariano Ventosa, Alvaro Baıllo, Andrés Ramos, and Michel Rivier. Electricity market modeling trends. *Energy Policy*, 33(7):897–913, 2005.

[132] Xavier Vives. *Oligopoly Pricing: Old Ideas and New Tools*. The MIT press, 2001.

[133] Zheng Wang, Yue Yin, and Bo An. Computing optimal monitoring strategy for detecting terrorist plots. In *AAAI'16*, 2016.

[134] Kevin Waugh, Drew Bagnell, and Brian D Ziebart. Computational rationalization: The inverse equilibrium problem. In *ICML '11*, pages 1169–1176, 2011.

[135] Jörgen W. Weibull. Price competition and convex costs. Technical report, SSE/EFI Working Paper Series in Economics and Finance, 2006.

[136] Bryce Wiedenbeck and Michael P Wellman. Learning payoffs in large symmetric games. In *AAMAS '15*, pages 1881–1882, 2015.

[137] Sean P. Winchell. Pakistan's ISI: The invisible government. *International Journal of Intelligence and Counterintelligence*, 16(3):374–388, 2003.

[138] Robert Woodley, Michael Gosnell, and Warren Noll. Adding culture to command decisions. In *ICCRTS '08*. IEEE, 2008.

[139] Marcel Zeelenberg and Rik Pieters. A theory of regret regulation 1.0. *Journal of Consumer psychology*, 17(1):3–18, 2007.

[140] Milan Zeleny. Games with multiple payoffs. *International Journal of Game Theory*, 4(4):179–191, 1975.

[141] Jingang Zhao. The equilibria of a multiple objective game. *International Journal of Game Theory*, 20(2):171–182, 1991.

[142] Y. B. Zhao and J. Y. Han. Two interior-point methods for nonlinear $p_*(\tau)$-complementarity problems. *Journal of Optimization Theory and Applications*, 102(3):659–679, 1999.

[143] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI '08*, pages 1433–1438, 2008.