Protecting Networks Against Diffusive Attacks:
Game-Theoretic Resource Allocation for Contagion Mitigation

by

Jason Tsai

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Computer Science)

May 2013

Acknowledgments

First, I would like to thank my advisor, Professor Milind Tambe. When I first joined, I had no idea what constituted a good advisor and it was only after a few years in the program that I realized how lucky I was to have chosen to work with Milind. I should have realized it early on. Between the first phone call to me before USC even sent out admittance letters and the numerous exchanges thereafter, I should have realized it. Milind, during the course of my doctoral pursuit, you have treated me with a level of courtesy and diligence that never ceases to astound me and I cannot thank you enough for the immeasurable effect your attitude and aptitude have had upon my own development as a researcher and a person. As a conference attendee once so eloquently said to me, ”Wow, that guy is like a rock star, I can’t believe he’s your advisor. You’re so lucky.” I could not agree more.

Next, I would also like to thank my two pseudo advisors whose hours I have taken far more than my fair share of over the course of my doctoral program. Stacy Marsella, you have been a wonderful mentor throughout my years at USC and your lab has been like a second home to me. Given our shared, highly peculiar educational background, I suppose it was only a matter of time before we worked together and I am so glad that it happened earlier rather than later. Emma Bowring, for your ceaseless confidence in me and support through the years, I cannot thank you enough. Even when we did not have regular calls, evidence of your guiding hand was ever-present, and I know that my research would have suffered dearly without your efforts.

Of course I would also like to thank the other members of my dissertation guidance committee for the time they devoted to my development and research. In a line of research at the intersection of many disciplines, my interdisciplinary committee could not have been more perfect for shaping and pushing my research to the heights I have been able to achieve. My sincerest gratitude to you all: Bhaskar Krishnamachari, Mathew McCubbins, and Sha Yang.

During my time at USC I have also had the honor to work with many great researchers: Gal Kaminka, Fernando Ordóñez, Wendy Wood, David Kempe, Eugene Vorobeychik, Natalie Fridman, Inbal Rika, Avishay Zilka, Christopher Kiekintveld, Matthew Taylor, James Pita, Matthew Brown, Thanh Nguyen, Shyamsunder Rathi, Harish Bellamane, Shira Epstein, Andrew Ogden, Ankur Sheel, and Xuezhi Wang.

I would also like to thank the rest of the TEAMCORE community that Milind has built into an international conglomerate, particularly those that I have had the pleasure of spending my PhD career with: Eric Shieh, Fei Fang, Leandro Marcolino, Chao Zhang, Yundi Qian, Paul Scerri, Rajiv Maheswaran, and Bo An. I would particularly like to thank Manish Jain for all of your advice over the years, for your great sense of humor, and for being a fantastic badminton buddy; Jun-young Kwak for being the best fake Korean friend I have ever had, for having an infectiously positive attitude towards everything, and for being a wonderful dinner/drinks companion; Zhengyu Yin for always having the answer to everything, for the many games you have tried to make me play, and for including me in all the brilliant pursuits you undertake in the years to come (please!); and Rong Yang for being the best officemate I could ask for, for being a constant motivator to me with your unbeatable work ethic, and for always having heartfelt discussions with me about everything from career goals to childhood stories.

Finally, I want to thank my family for their support through the years. In particular, thank you to my parents for supporting me as I stayed on my own in the U.S. for so many years and putting up with my highly infrequent calls home. Lastly, I would like to thank my beautiful wife Alice for coming into my life, being my best friend, and for saying ‘Yes’. Thank you for your patience, kindness, and support and for lighting up my life from the moment I wake up in the morning to the moment I close my eyes at night.

Table of Contents

List of Figures

List of Tables

Abstract

Many real-world situations involve attempts to spread influence through a social network. For example, viral marketing is when a marketer selects a few people to receive some initial advertisement in the hopes that these ‘seeds’ will spread the news. Even peacekeeping operations in one area have been shown to have a contagious effect on the neighboring vicinity. Each of these domains also features multiple parties seeking to maximize or mitigate a contagious effect by spreading its own influence among a select few seeds, naturally yielding an adversarial resource allocation problem. My work models the interconnected network of people as a graph and develops algorithms to optimize resource allocation in these networked competitive contagion scenarios.

Game-theoretic resource allocation in the past has not considered domains with both a networked structure and contagion effects, rendering them unusable in critical domains such as rumor control, counterinsurgency, and crowd management. Networked domains without contagion effects already present computational challenges due to the large scale of the action space. To address this issue, my first contribution proposed efficient game-theoretic allocation algorithms for the graph-based urban road network domain. This work still provides the only polynomial-time algorithm for allocating vehicle checkpoints through a city, giving law enforcement officers an efficient tool to combat terrorists making their way to potential points of attack. Second, I have provided the first game-theoretic treatment for contagion mitigation in social networks and given practitioners the first principled techniques for such vital concerns as rumor control and counterinsurgency. Finally, I extended my work on game-theoretic contagion mitigation to address uncertainty about the network structure to find that, contrary to what evidence and intuition suggest, heuristic sampling approaches provide near-optimal solutions across a wide range of generative graph models and uncertainty models. Thus, despite extreme practical challenges in attaining accurate social network information, my techniques remain near-optimal across numerous forms of uncertainty in multiple synthetic and real-world graph structures.

Beyond optimization of resource allocation, I have further studied contagion effects to understand the effectiveness of such resources. First, I created an evacuation simulation, ESCAPES, to explore the interaction of pedestrian fear contagion and authority fear mitigation during an evacuation. Second, using this simulator, I have advanced the frontier in contagion modeling by developing empirical evaluation methods for comparing and calibrating computational contagion models that are critical in crowd simulations and evacuation modeling. Finally, I have also conducted an examination of agent-human emotional contagion to inform the rising use of simulations for personnel training in emotionally-charged situations.

Chapter 1: Introduction

Many real-world situations involve attempts to spread influence through a social network. For example, viral marketing is when a marketer selects a few people to receive some initial advertisement in the hopes that these ‘seeds’ will spread the news. Even peacekeeping operations in one area have been shown to have a contagious effect on the neighboring vicinity [Beardsley, 2011]. Each of these domains also features multiple parties seeking to maximize or mitigate a contagious effect by spreading its own influence among a select few seeds, naturally yielding an adversarial resource allocation problem.

Recently, game-theoretic resource allocation has been used to generate policies for numerous security agencies facing a similar challenge [Tsai et al., 2009; Pita et al., 2011; Shieh et al., 2012; Yin et al., 2012]. Specifically, a Stackelberg game is used to model these situations. Stackelberg games model a leader and a follower [von Stackelberg, 1934], where the leader must commit to a policy which the follower will observe and then act against.

The many deployed applications using game-theoretic resource allocation have only considered isolated targets without contagious effects [Tsai et al., 2009; Pita et al., 2011; Shieh et al., 2012; Yin et al., 2012]. In the U.S. Federal Air Marshal domain, for example, the security forces are responsible for placing undercover Federal Air Marshals aboard flights to and from the United

States. With far fewer marshals than flights, the decision of which flights to take each day can naturally be modeled as a resource allocation problem. Based on the value of each flight, the FAMS can estimate a payoff for each outcome in the game and model the problem as a Stackelberg game. The Stackelberg game is used in all of the above applications to describe the asymmetric information between the two players. The security forces can then solve for the Stackelberg equilibrium strategy and implement this as their allocation policy. Game-theoretic resource allocation in the past, however, has not considered domains with both a networked structure and contagion effects.

In domains such as counterinsurgency and viral-marketing, actions (e.g., selecting a subset of people to target) carry with them a probabilistic contagion effect across a social network. These domains admit six primary challenges beyond that of past work in game-theoretic resource allocation:

- *Large Action Space*: Actions in this domain translate into subsets of nodes of a given size (e.g., number of resources), which increases combinatorially with the size of the network. Standard computing equipment today cannot store the payoff matrix for a game with networks of larger than 20 nodes with only 3 resources.

- *Intractable Payoff Calculation*: Payoff determination becomes computationally intractable. Specifically, given a pair of player actions, calculating the expectation of the propagation process has been shown to be #P-Hard under commonly assumed propagation dynamics [Chen et al., 2010b]. Since any payoff in this domain is a function of this expectation, the payoff associated with a given pair of player actions is also computationally intractable.

- *Network Uncertainty*: Real-world social networks are difficult to learn with high accuracy. In particular, we can easily be uncertain about the existence or strength of a set of links or nodes in the network. Thus, parties in such domains must generate policies under uncertainty about the network structure.

- *Contagion effects*: Beyond the optimization of resources, our understanding of resources' impact on contagion effects in the real-world remain a challenging area of study, particularly at the population-level such as in crowds. Qualitative and quantitative statements about the contagious effects of resources in such settings is crucial to properly optimizing their placement.

- *Population-level contagion modeling*: Modeling contagion phenomena at the population-level is extremely difficult due to the lack of empirical evidence and validation techniques. Most contagious effects do not have obvious physical signs, necessitating the use of proxy data or additional technology to provide unbiased measures. Even with a reasonable dataset, the stochastic or 'individual attribute'-based nature of most of these effects causes validation of models to also be very challenging.

- *Human-agent contagion effects*: To train our resources to better manage contagion effects, the use of simulations and virtual humans has been on the rise. Although many researchers have studied human-human contagion effects at the individual level, gaps in our knowledge remain in agent-human interactions. These gaps are particularly important in emotionally-charged contexts such as evacuations, fire fights, and violent protests where the emotional contagion effect may have unintended impacts on user behaviors.

In light of the aforementioned challenges, I divide the discussion of my contributions to the three pertaining to the optimization of resource allocations in networked competitive contagion scenarios (NCCS) and the three into contagion effects and modeling.

1.1 Contributions: Game-Theoretic Resource Allocation

With respect to game-theoretic resource allocation, my work provides three primary contributions. First, I extended past work in game-theoretic resource allocation into networked domains and developed efficient algorithms to address the computational challenges introduced by the exponential explosion in the actions [Tsai et al., 2010]. Second, I have introduced a scalable approach to calculating game-theoretic equilibria in NCCS's where one player seeks to maximize his influence while a second player seeks to mitigate the first player's influence by spreading his own influence, something entirely unaddressed by past researchers due to its computational complexity. I applied these techniques to realistic network structure data from counterinsurgency efforts in Afghanistan and provided the first fully game-theoretic treatment for this domain that scales to actual problem sizes faced by the U.S. Army. In this domain, network uncertainty is a critical challenge as insurgents typically have an informational advantage over incoming forces. Thus, my final contribution in game-theoretic resource allocation analyzes equilibria computation under asymmetric information about network structure and I find that, contrary to what evidence and intuition suggest, simple sampling heuristics perform near-optimally under numerous forms of uncertainty and graph structures.

1.1.1 Game-Theoretic Resource Allocation in Networks

Protecting targets against potential attacks is an important problem for security forces worldwide. Recently, game-theoretic approaches have been used to assist in devising these strategies, including applications in use by the Los Angeles International Airport [Pita et al., 2008] and the Federal Air Marshals Service [Tsai et al., 2009]. Both of these domains were modeled as a set of independent targets that an adversary is attempting to attack and a defender is trying to protect.

In many other domains, however, there is an interconnectedness between targets that can naturally be modeled as a network. For example, city maps can be modeled with intersections as nodes and roads as edges, where nodes are targets for attackers. While this could be modeled by specifying the targets in isolation as done in previous work, clearly an adversary must traverse the city's roadways to arrive at the target. Thus, security measures can attempt to interdict adversaries en route by placing checkpoints on the roads. However, each checkpoint no longer protects a single target and can potentially prevent attacks to multiple targets simultaneously.

After the devastating terrorist attacks in 2008, Mumbai police began deploying such randomized checkpoints as one countermeasure to prevent future attacks. The extension of game-theoretic resource allocation into this type of networked domain, however, creates immense computational challenges. Existing solvers take time polynomial in the number of actions of both players [Kiekintveld et al., 2009; Paruchuri et al., 2008; Conitzer and Sandholm, 2006]. In the roadway setting, every path from source to target is an attacker action and every set of r or fewer edges, where r is the maximum number of checkpoints, is a defender action. Since attacker actions grow exponentially with the size of the network and defender actions grow exponentially with r, existing methods quickly become intractable when applied to real-world domains.

In my work on this subject [Tsai et al., 2010], I developed an efficient procedure for generating checkpoint deployments based on two key ideas: (i) an approximation of the strategy space that is polynomial in the size of the network and is solved using a linear program; (ii) two efficient sampling techniques to map solutions back to the original space. Specifically, to avoid the exponential strategy space over all possible combinations of checkpoint placements (the joint distribution), my methods operate on the marginal probabilities of edges, i.e., the total probability of placing a checkpoint on an edge.

In addition to verifying the immense speedup of RANGER (polynomial with problem size) and its accompanying sampling techniques over the previous fastest method, DOBSS (exponential with problem size), I also provided an in-depth experimental analysis of the solution quality obtained and find RANGER, with slight variance based on the sampling method used, is able to produce optimal solutions in 89% of the 3,000 random graphs tested. Finally, I also applied this solution to a model of the Mumbai road network, depicted in Figure 4.4, and show that solutions obtained using RANGER differ substantially from a range of heuristic methods in terms of both solution quality as well as the actual distributions being generated.

Follow-up research based on this work [Jain et al., 2011a, 2013] has introduced the use of a double oracle method to solve this domain. Double oracle techniques provide the basis for my latest work in networked competitive contagion scenarios because it is able to leverage recent research in influence blocking maximization.

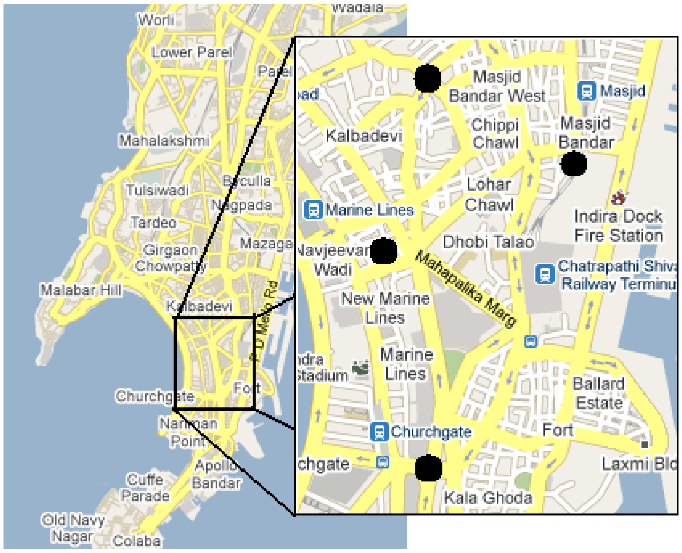


Figure 1.1: Southern Mumbai with historical targets marked.

1.1.2 Game-Theoretic Contagion Blocking in Networks

The third contribution was a first step towards game-theoretic resource allocation in competitive contagion problems with an example application to counterinsurgency operations. Counterinsurgency (COIN) is the contest for the support of the local leaders in an armed conflict and can include a variety of operations such as providing security and giving medical supplies [U.S. Dept. of the Army and U.S. Marine Corps, 2007]. Just as in word-of-mouth advertising [Trusov et al., 2009] and peacekeeping operations [Beardsley, 2011], these efforts carry a social effect beyond the action taken that can cause advantageous ripples through the neighboring population [Hung, 2010]. Moreover, multiple intelligent parties attempt to leverage the same social network to spread their message, necessitating an adversary-aware approach to strategy generation.

I modeled the interaction between two parties as a graph with one player (the influencer) attempting to spread influence while the other player (the mitigator) attempts to stop the probabilistic propagation of that influence by spreading their own influence. This 'blocking' problem models situations faced by governments/peacekeepers combatting the spread of terrorist radicalism and armed conflict with daily/weekly/monthy visits with local leaders to provide support and discuss grievances [Howard, 2011].

This model is a form of influence blocking maximization (*IBM*) problems [Budak et al., 2011; He et al., 2012], which are a competitive extension of the widely studied influence maximization problem [Chen et al., 2010b; Kimura et al., 2010]. Past work in influence blocking maximization has looked only at the best-response problems and has not produced algorithms to generate the game-theoretic equilibria necessary for this repeated-interaction domain, which is the focus of my work.

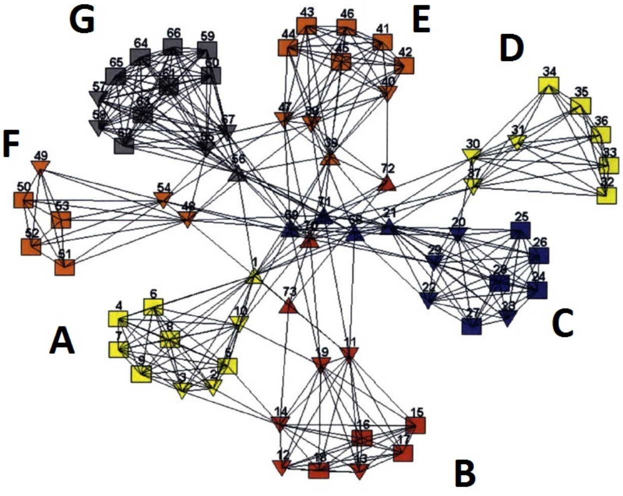


Figure 1.2: Real Afghan leadership graph from Hung et. al

I began by proving approximation quality bounds on the double oracle approach when one of the oracles is approximated and combine this with a greedy approximate oracle to produce a more efficient approximate algorithm. To further increase scalability, I introduced two heuristic oracles, LSMI and PAGERANK, that offer much greater efficiency. I concluded with an experimental exploration of a variety of combinations of oracles, testing runtime and quality on random scale-free graphs, a real-world leadership network in Afghanistan (Figure 5.3), synthetic leadership networks, and a real-world social network. I found that the performance of the basic PAGERANK oracle suffers minimal loss compared to LSMI in leadership networks that possess clusters of highly interconnected nodes, but performs far worse in sparsely interconnected real-world social networks and scale-free graphs. Finally, I found that an unintuitive blend of the two oracles offers the best combination of scalability and solution quality.

1.1.3 Bayesian Game-Theoretic Contagion Blocking in Networks

While the previous work provided the first treatment of these networked competitive contagion scenarios, my final contribution extended this line of research to address network uncertainty. Related prior research has assumed that full information about network structure is available to both players and that accurate network knowledge is crucial to generating high-quality strategies. As noted previously, however, real-world contagion information is extremely hard to gather. This is particularly true in counterinsurgency where the insurgents are typically an indigenous group that has an informational advantage, and the mitigators often have uncertainty about their knowledge of the social network [Hung, 2010].

In this work, I assume that the influencer (an insurgent group) has perfect knowledge of the graph structure, while the mitigator is uncertain about the influence network. In the resulting

Bayesian game, a type of the influencer corresponds to a particular instantiation of the influence graph, and the mitigator must reason over the distribution over these graphs (i.e., influencer types) in order to compute an optimal strategy.

Past work in influence maximization and social network analysis highlight the importance of graph structure in strategy generation [Kempe et al., 2003; Budak et al., 2011; He et al., 2012]. In addition, previous work on Bayesian security games has shown that not accounting for even small degrees of payoff uncertainty can lead to large drops in solution quality [Kiekintveld et al., 2011]. Thus, the expectation should be that strategies generated without modeling most of the uncertainty about graph structure will do far worse than the optimal solution to the Bayesian game. Supporting this intuition, I showed that there are cases where a mitigator who has incorrect information about a single edge can suffer unbounded loss and that quantifying the impact of changing a single edge in a given graph is #P-Hard. I also show empirically that, indeed, under our models of uncertainty, optimal mitigator strategies for different influencer types are vastly different.

However, while past work has focused on sophisticated algorithms for Bayesian security games [Jain et al., 2011b; Kiekintveld et al., 2011; Yin and Tambe, 2012], I showcase the opposite approach that runs directly counter to what intuition and my initial experiments suggest: ignoring the vast majority of uncertainty has minimal impact. Specifically, I show through extensive experiments that computing a mitigation strategy based on a game with only a few *randomly* sampled influencer types yields near-optimal rewards for widely varied models of uncertainty. I experimented on 3 different synthetic graph models with and without resource imbalances on both sides, 5 models of uncertainty, weighted/unweighted counting of nodes, varied edge weight distributions, varied graph sizes, varied degrees of uncertainty, and varied degrees of sampling.

I also conduct experiments on two real-world social networks using two different models graph construction. In all, I studied over 200 experimental settings and consistently observe the same result: simple sampling techniques perform near-optimally. This suggests that even in domains as challenging as this, models which ignore uncertainty may nevertheless be robust to it.

1.2 Contributions: Contagion

Beyond optimization of resource allocation, I have also studied contagion interactions to advance our understanding of the effectiveness of such resources at managing contagion. To study population-level contagion interactions, I created an evacuation simulation called ESCAPES, that models a novel combination of features: (i) emotional contagion; (ii) different agent types, including fear-mitigating authority figures; (iii) informational interactions; (iv) behavioral interactions [Tsai et al., 2011b]. Using ESCAPES, I have also developed the first empirical evaluation methods for computational contagion models [Tsai et al., 2011a] by using the ESCAPES evacuation simulation system and real world videos to compare models of contagion. Finally, to better understand agent-level contagion interactions that are crucial to simulation training of personnel, I have also examined emotional contagion from virtual humans to human users.

1.2.1 Contagion management in crowds

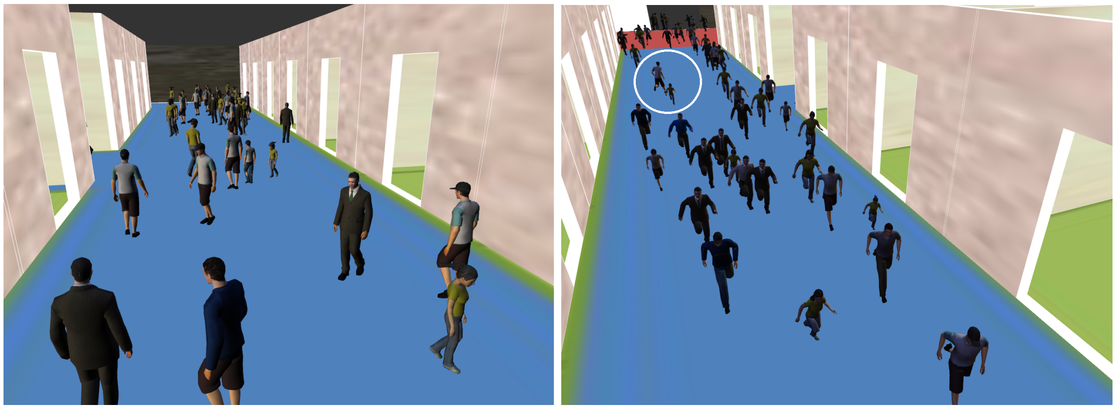
To better understand the interaction between fear contagion in a crowd and the fear-mitigation of authority figures during evacuations, I created the ESCAPES evacuation simulation. The ESCAPES system is a multi-agent evacuation simulation tailored to the needs of airport security officials based on existing psychological and evacuation research [Tsai et al., 2011b]. The first

Phenomenon	Ref.	Feature
People forget their entrance	[Chertkoff and Kushigian, 1999]	Misc.
First-time Visitors	[Diamond et al., 2010]	SoK / SCT
Heightened emotions → chaos	[Smith and Ellsworth, 1985]	Emotions / EC
Herding behavior	[Helbing et al., 2000]	SCT / Families
Pre-evacuation delay	[D.S.Mileti and J.L.Sorensen, 1990] [J.L.Bryan, 2002]	SoK / Families
Families gather before exiting	[Proulx and Fahy, 2008]	Families
Authorities calm people	[Smith and Ellsworth, 1985]	Auth / Emotions

Table 1.1: Phenomena modeled in ESCAPES

major aspect of any evacuation is fear, and although there is substantial debate on the existence of 'panic' in evacuations, the presence of fear is undisputed [Russell and Beigel, 1976]. For the purposes of this work, ESCAPES includes a model of fear and the way it spreads through a crowd. ESCAPES also includes multiple agent types including regular travelers, authority/security figures, and families, as these have been documented as having the most impact in an airport evacuation [Diamond et al., 2010]. Finally, in discussions with airport security officials, incomplete knowledge of the environment was cited as a major concern. Thus, agents possess incomplete knowledge of the exits and the event causing the evacuation. Screenshots of the simulator exhibiting normal agent behavior and evacuation behaviors are shown in Figure 1.3.

ESCAPES agent interactions include three separate phenomena: spread of knowledge, emotional contagion, and social comparison. Emotional Contagion (EC) is the well-documented phenomena that causes one person's emotional state to be impacted by neighboring people's emotional state [Hatfield et al., 1994]. I incorporate EC in the system as a logical byproduct of the inclusion of fear in the presence of crowds. Evacuation literature shows that the crucial seconds people spend before actively moving towards an exit greatly impact their survivability and is largely due to uncertainty about the nature of the evacuation [D.S.Mileti and J.L.Sorensen,



(a) Normal agent behavior (b) Evacuation behavior

Figure 1.3: ESCAPES simulator screenshots

1990]. Finally, in a situation where people don't have all the information, following others is a commonly seen phenomenon. Thus, I include a 'Spread of Knowledge' (SoK) component, which models the spread of information about an event and that an evacuation is truly necessary. Social Comparison Theory (SCT) is a theory of how one person impacts another at a broad level, positing that people perceived to be similar to each other will mimic each other [Festinger, 1954]. SCT is used to direct people's actions when they have no knowledge of the environment.

I show that inclusion of these factors leads to a number of emergent behaviors documented in evacuation literature, as summarized in Table 1.1. Finally, I conduct tests on a model of a terminal at Los Angeles International Airport and begin to provide answers to security officials' questions about authority figure policies.

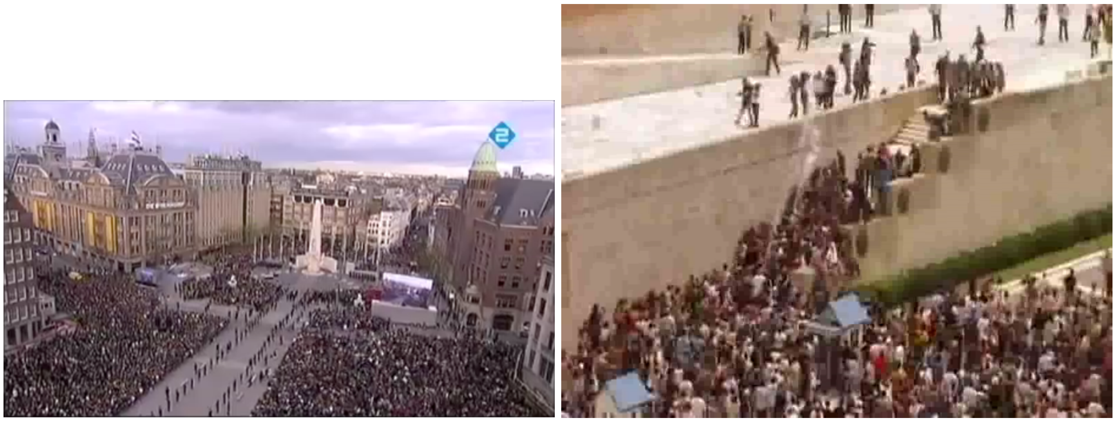
1.2.2 Empirical Evaluation of Emotional Contagion Models

The model of contagion propagation is a key element in a competitive contagion problem. However, detailed computational models of contagion models are difficult to formulate and even harder to validate convincingly with real data due to the nature of most contagious phenomena. In my work, I make the first empirical comparison of computational models of emotional contagion in crowd panics that have been proposed in the literature [Tsai et al., 2011a].

Researchers at VU University introduced (ASCRIBE) in Bosse et. al (2009) that used a deterministic, interaction-based model derived directly from a social psychology theory of emotional contagion [Barsade and Gibson, 1998]. This model is a prototypical example of the heat dissipation phenomena studied in thermodynamics wherein neighboring substances will transfer energy to each other at rates unique to each substance (i.e., specific heat). In contrast, Durupinar [Durupinar, 2010] used a traditional linear threshold model wherein successive interactions with emotionally 'infected' people raises the chance of infection with an emotion. Both of these models were briefly introduced in Chapter 2.

Although both models come from studies of contagion phenomena, they use fundamentally different mechanisms. While work could proceed using both approaches by extending existing models to accurately reproduce increasingly complex situations, it remains unclear which contagion paradigm should be used in emotional contagion. Perhaps a new mechanism should be designed, but the lack of data in this domain makes evaluation very difficult. I not only empirically compared these two paradigms but identify the key features that should be added to the underlying contagion mechanisms to further improve their fidelity in reproducing human emotional contagion.

I first use the ESCAPES evacuation simulation I developed to explore the impact of replacing the original ESCAPES model of emotional contagion with these two models on predicted outcomes, showing substantial differences in their predictions, and motivating the need for an accurate model of emotional contagion in this context. Even in simulation, I am able to identify key differences that indicate epidemiological / social contagion models are less suited to modeling emotional contagion. Next, I attempted to reproduce real video footage of a panic situation from Amsterdam using each of the models, showing the ASCRIBE model to indeed be superior to both the Durupinar model and the original ESCAPES model, beating out the Durupinar model by 14% *per agent per frame* during the 15s scene. To identify which of the key features causes the differences in the results, I also tested hybrid models to conclude that while adding a 'decay' feature (as found in the Durupinar model) to the ASCRIBE model does not improve it, removing proximity effects and a graduated effect of fear substantially worsen the model. Finally, I performed the same evaluation on a second video from Greece and show the ASCRIBE model to again be superior, outperforming the Durupinar model by 12% per agent per frame during the *four-second* scene.



(a) Amsterdam video

(b) Greece video

Figure 1.4: Amsterdam and Greece video screenshots

This work informs the need for distinguishing between diffusions with graduated effects and binary effects. In particular, for the contagion of phenomena such as emotions which naturally exhibits a graduated effect, a graduated contagion mechanism such as the one featured in ASCRIBE appears most suitable.

1.2.3 Emotional Contagion with Virtual Characters

My final contribution to the study of emotional contagion is an experimental evaluation of the emotional contagion that occurs between virtual characters and human users, something that has largely been overlooked by virtual human and social psychology researchers but is particularly critical as the use of simulation environments for training becomes more widespread. Pursuant of this goal, three sets of studies are conducted. The first study examines the pure contagion case by simply showing subjects a still image of a virtual character with either a happy expression or a neutral expression and then assessing the subject's self-reported happiness thereafter. The use of a still image as a manipulation follows from previous studies in human-human emotional contagion [Small and Verrochi, 2009; Wild et al., 2001]. The second study adds the presentation of a game-theoretic situation known as a Stag Hunt along with the character image to assess both the contagion and the behavioral impact of the virtual character in a strategic setting. While studies have shown that emotional contagion can impact one's propensity to trust and enhance perceived cooperation among other findings [Barsade, 2002; Dunn and Schweitzer, 2005], there has been far less work showing behavioral impacts in strategic situations. Thus, we also attempt to examine whether behavioral impacts arise in strategic situations to better understand its potential impacts in real-world agent applications. Finally, the third study examines the post-hoc hypothesis that the presentation of a decision to the user dampens the emotional contagion effect. Specifically, we present the same strategic situation as in the second study, but with the decision already made for the subject.

In this work, I provide the first experimental results supporting the existence of emotional contagion between virtual agents and humans. Results show a very large increase in self-reported happiness from only adding a smile to an otherwise identical still image of a virtual character. In the second study, when the character is placed in the context of a strategic decision, both subject behavior and subject self-reports of happiness are only impacted significantly by one character. The last study, which removes the user's decision from the previous experiment, finds that the character's expression's affect on emotion returns significantly, implying that a strategic decision posed to users will dampen the emotional contagion effect beyond only reading about a situation. These results serve as a preliminary study to alert agent researchers to the impacts that virtual character emotions may have on human users.

1.3 Summary

Extending game-theoretic resource allocation techniques into contagion-based situations in social networks opens the door to an entirely new set of real-world problems for game theory to help address. In my work I have proposed game-theoretic allocation algorithms for large-scale networked domains without contagion effects, specifically deploying security checkpoints in an urban road network. Combining this with my studies into contagion phenomena, I have provided the first game-theoretic treatment of contagion mitigation in social networks. Finally, I have extended this approach to account for real-world uncertainty in network structure using sampling techniques that produce near-optimal strategies. These techniques provide principled, robust tools for real-world decision-makers where none have existed and rule-of-thumb heuristics have been the norm.

A thorough understanding of contagion effects is critical to understanding the effectiveness of resources in these contexts; thus, I have also advanced our understanding of contagion phenomena in three primary ways. First, I created an evacuation simulation, ESCAPES, to explore the interaction of pedestrian fear contagion and authority fear mitigation during an evacuation. Second, I used ESCAPES to developing empirical evaluation methods for comparing and calibrating computational contagion models that are critical in crowd simulations and evacuation modeling. Finally, I have also conducted an examination of agent-human emotional contagion to inform the rising use of simulations for personnel training in emotionally-charged situations.

1.4 Guide to Thesis

This thesis is organized as follows. Chapter 2 introduces the necessary background material for the research presented here. Chapter 3 provides an overview of research related to this field of study. Chapters 4 discusses my extension of game-theoretic resource allocation to large-scale networked domains. The work described in Chapter 5 introduces the first analysis of game-theoretic resource allocation for networked competitive contagion scenarios with an example application to counterinsurgency. Chapter 6 presents my extension of the work on NCCS's to address network uncertainty where I show that simple sampling heuristics produce near-optimal solutions with a fraction of the runtime. Chapter 7 begins the discussion of my work in contagion phenomena by introducing the ESCAPES evacuation simulation I created to explore the interaction between fear contagion and fear mitigation during evacuations. Chapter 8 outlines the empirical technique I developed to compare contagion models using real-world data. Wrapping up my primary contributions, Chapter 9 details my study of agent-human emotional contagion between virtual characters and human users.

Chapter 2: Background

In the work presented herein, I draw upon two particular areas of research which require some introduction. First, my work in resource allocation in networked competitive contagion scenarios extends prior research in game-theoretic resource allocation. As such, I outline the relevant background from this literature in Section 2.1 by introducing Zero-sum Games, Stackelberg Games (Section 2.1.1), Bayesian Games (Section 2.1.2), and the Double Oracle Method for solving these games (Section 2.1.3). Second, diffusion studies feature a plethora of propagation mechanisms that broadly fall into two classes that I describe in more detail in Section 2.2.

2.1 Game Theory

In standard game theoretic literature, there exist multiple players who can choose actions that will have a resulting payoff that depends on the actions of the other players. The full description of available actions and corresponding payoffs is often represented in *normal form* as in Table 2.1 and is usually referred to as a *payoff matrix*. Here there are two players, referred to commonly as the row player and the column player. The row player here has actions A and B while the column player can choose between actions C and D. The actions are often referred to as the *pure strategies* of the players and will be iterated as $\omega_R \in \Omega_R$ for the row player and $\omega_C \in \Omega_C$ for the column player. The payoffs for each player for each of the four possible outcomes is given by the paired numbers in each cell of the table. For example, in this game, if the row player plays A and the column player plays C, then the row player receives a payoff of 2 and the column player receives a payoff of 1. We will refer to these payoffs also as utilities and denote the utility of a specific pair of player actions by $u_R(\omega_R, \omega_C)$ for the row player and $u_C(\omega_R, \omega_C)$ for the column player.

	C	D
A	2,1	4,0
B	1,0	3,2

Table 2.1: Payoff table for an example normal form game.

Given the payoff matrix, we must also specify any additional information the players possess and their model of the other players. In simultaneous-move situations, where each player must decide on a strategy with no additional information, each player determines a strategy based only on the assumption that the other player will act rationally and maximize their expected payoff. In this game, this assumption would lead both players to conclude that the row player should never play B, since regardless of what the column player plays, A will always produce a higher reward for the row player. Given that the row player will always play A, the column player will always play C. Thus, the optimal strategies for the players are to play 100% A for the row player and to play 100% C for the column player. Although this particular game has an equilibrium solution with only one pure strategy for each player, this may not always be the case. In general, we solve for a probability distribution over available pure strategies, which is commonly referred to as a *mixed strategy* and I denote ρ_R for the row player and ρ_C for the column player. In this work, I will refer to the probability associated with a particular action ω in a given mixed strategy ρ as

$\rho(\omega)$. The *support set* of a mixed strategy is the set of pure strategies for which the associated probability is greater than zero. The solution concept for simultaneous-move games is known as the Nash equilibrium and guarantees that each player's strategy is a best-response to the other's and no single player has an incentive to change their strategy. Referring to the utility derived from a given pair of mixed strategies as $U_C(\rho_R, \rho_C)$ and $U_R(\rho_R, \rho_C)$ and optimal mixed strategies as ρ^*, this can be formalized as:

$$U_R(\rho_R^*, \rho_C^*) \geq U_R(\rho_R, \rho_C^*), \forall \rho_R \neq \rho_R^* \tag{2.1}$$

$$U_C(\rho_R^*, \rho_C^*) \geq U_C(\rho_R^*, \rho_C), \forall \rho_C \neq \rho_C^* \tag{2.2}$$

The unique situation wherein payoffs for the two players always sum to zero ($u_c(\omega_R, \omega_C) = -u_r(\omega_R, \omega_C)$) is referred to as a zero-sum game. Zero-sum games have a number of critical properties from a computational standpoint, the most important of which is that a Nash equilibrium can be calculated in time polynomial in the size of the game matrix via a Maximin linear program:

$$\begin{aligned} \text{Maximize} \quad & U_R \\ \text{s.t.} \quad & U_R \leq \sum_{\omega_R \in \Omega_R} \rho_R(\omega_R) \cdot u_R(\omega_R, \omega_C), \forall \omega_C \in \Omega_C && (2.3) \\ & \sum_{\omega_R \in \Omega_R} \rho_R(\omega_R) = 1 && (2.4) \end{aligned}$$

2.1.1 Stackelberg Games

In many applications of game theory to resource allocation, a Stackelberg game is used to model the situation instead of a simultaneous-move game [von Stackelberg, 1934]. In Stackelberg games, one of the players (i.e., the leader) must declare his strategy first, after which the other player (i.e., the follower) can decide his strategy using this information. While this may seem like a disadvantage to the leader, it has been shown that this first-mover situation can never hurt a player. Consider the game matrix from before in Table 2.1, as an example, and let the row player be the leader in a Stackelberg situation. Instead of unequivocally playing A which was previously optimal, the row player can now declare that he will play B 100% of the time. This would induce the column player to play D, giving the row player an equilibrium payoff of 3 instead of the 2 he received in the simultaneous-move situation.

The solution concept adopted most often is that of a Strong Stackelberg Equilibrium (SSE). Beyond the mutual best-response that Nash equilibrium also entails, Strong Stackelberg Equilibrium also assumes that the follower will break ties in favor of the leader to resolve the selection problem that arises when multiple equilibria exist. Most importantly for my work, it has been shown that an SSE in a zero-sum game is equivalent to Nash equilibrium [Yin et al., 2010]. Since a Nash equilibrium can be efficiently found via a Maximin linear program for zero-sum games, a Strong Stackelberg Equilibrium can also be found in polynomial time for zero-sum games.

2.1.2 Bayesian Games

Many real-world situations require a player to optimize against an array of adversaries or, equivalently, a single adversary about which the player is uncertain. In these cases, a Bayesian game model is adopted that explicitly enumerates the multiple payoff matrices against which the player must optimize as well as the probability distribution over which these payoff matrices occur. One example is shown in Table 2.3, where the row player faces two distinct payoff matrices. In this

	C	D
A	2,1	4,0
B	1,0	3,2

Table 2.2: versus column player 1

	C	D
A	1,2	4,4
B	3,2	3,1

Table 2.3: versus column player 2

particular case there are two Bayesian *types* that I will generally denote $\lambda \in \Lambda$ and each occurs with some known probability P_λ.

In my work, I make use of Bayesian Zero-sum games which can be solved in time polynomial with respect to the product of the size of the game matrix and the number of types. Unsurprisingly, this is achieved through a trivial extension of the Maximin linear program presented earlier. In the domains I address, however, the size of the game matrix as well as the number of types grow exponentially with respect to the underlying domain representation. This renders even a polynomial time solver unusable on anything but trivial problem sizes and necessitates algorithmic advances to handle real-world problem sizes.

2.1.3 Double Oracle Method

As mentioned, optimal mixed strategies in zero-sum games can be efficiently determined using a Maximin (or equivalently a Minimax) algorithm. In domains that feature an action space that is far too large to hold in memory, a maximin formulation does not provide a realistic solution. In recent extensions of my 2010 work on urban road network security, Jain et. al (2011) use a double oracle approach that builds the action space incrementally. This state-of-the-art method is also used in my work in networked competitive contagion scenarios.

Double oracle algorithms for zero-sum games also use a Maximin linear program at the core, but the payoff matrix is grown incrementally by two oracles. This process is shown in Algorithm 1, where the two players are referred to as the *defender* and the *attacker*. **D** is the set of defender actions generated so far, and **A** is the set of attacker actions generated so far. MaximinLP(**D**,**A**) solves for the equilibrium of the game that only has the pure strategies in **D** and **A** and returns ρ_d and ρ_a, which are the equilibrium defender and attacker mixed strategies over **D** and **A**. DefenderOracle($\cdot$), generates a defender action that is a best response against ρ_a among *all* possible actions. This action is added to the set of available pure strategies for the defender **D**. A similar procedure then occurs for the attacker. Convergence occurs when neither best-response oracle generates a pure strategy that is superior to the given player's current mixed strategy against the fixed opponent mixed strategy. The number of attacker and defender actions in the payoff matrix can grow to the full matrix in the worst case, but in practice is generally much smaller. It has been shown that with two optimal best-response oracles, the double oracle algorithm converges to the Maximin equilibrium [McMahan et al., 2003].

Algorithm 1: DOUBLE ORACLE ALGORITHM

```
Initialize D with random defender allocations.
Initialize A with random attacker allocations.
repeat
    $(\rho_d, \rho_a)$ = MaximinLP(D,A)
    $\mathbf{D} = \mathbf{D} \cup \{$DefenderOracle($\rho_a$)$\}$
    $\mathbf{A} = \mathbf{A} \cup \{$AttackerOracle($\rho_d$)$\}$
until convergence
return $(\rho_d, \rho_a)$
```

2.1.4 Double Oracle: Example

To illustrate the double oracle algorithm in more detail, consider the game described by the payoff matrix featured in Table 2.4. As per standard game-theoretic notation, the row player's available actions are 1 and 2 and the column player's available actions include A, B, and C. If the row player plays 1 and the column player plays A, then the row player receives a payoff of 3 and the column player a payoff of -3. Although this game could be solved by a single Maximin run, I will describe

	A	B	C
1	3,-3	-1,1	2,-2
2	1,-1	2,-2	-2,2

Table 2.4: Example game’s full payoff matrix

the solution procedure used by the double oracle algorithm to clarify the process. Initially, each player’s actions are randomly seeded with a single action from the complete action space of the original game. Suppose the row player, is seeded with action ‘1’ and the column player, is seeded with action ’C’ as shown in Table 2.5. Then $\mathbf{D} = \{1\}$ and $\mathbf{A} = \{C\}$. This subgame is trivially solved using a Maximin linear program that produces the optimal strategy for both players, which is to simply play their only available action 100% of the time ($\rho_R = \rho_a = \{1.0\}$).

Next the algorithm consults two oracles for the next action to add to the subgame for each player. In this case, the column player’s best-response oracle produces ‘B’ as the optimal action for the attacker to take when the defender is playing a strategy of 100% ‘1’. The row player’s best-response oracle, by contrast, produces ‘1’ as the best-response to the current adversary strategy of 100% ‘C’ and chooses to add action ‘1’ which already exists in the subgame. The subgame is now composed of one action for the defender ($\mathbf{D} = \{1\}$) and two actions for the attacker ($\mathbf{A} = \{C, B\}$). A Maximin solver is again run to determine the optimal strategy for each player, producing a new pure-strategy equilibrium ($\rho_R = \{1.0\}, \rho_C = \{0.0, 1.0\}$).

Both oracles are consulted again, with the column player’s best-response oracle again returning ‘B’ as the optimal action to the current row player strategy (play ‘1’ 100%), but the row player’s best-response oracle now returning action ‘2’ as the best-response to the current column player strategy (play ‘B’ 100%). The subgame grows to the 2x2 matrix shown in Table 2.6 and the Maximin linear program is again run to solve it, producing new optimal strategies for each

	C
1	2,-2

Table 2.5: Initial subgame

	B	C
1	-1,1	2,-2
2	2,-2	-2,2

Table 2.6: At convergence

player ($\rho_d = \{\frac{4}{7}, \frac{3}{7}\}, \rho_a = \{\frac{4}{7}, \frac{3}{7}\}$). Another query to each oracle reveals that both players' best-responses are already included in the subgame. At this point the algorithm has converged and the Maximin equilibrium strategies for both players in the full game have been determined.

The payoff matrix generated at convergence is shown in Table 2.6. Notice that the attacker action 'A' was never added to the game. Not only does this limit the size of the payoff matrix stored in memory, but this also means that no payoffs associated with action 'A' need to be generated.

2.2 Propagation Dynamics

In studies of problems featuring diffusion, such as competitive contagion games, the propagation dynamics must first be decided upon. In the absence of context-specific data that would enable a complete modeling of the propagation dynamics in an actual social network, however, canonical propagation dynamics are used to illustrate the applicability of algorithms across the parameter space. The literature most often features one of two classes of propagation dynamics to model diffusion processes in a social network: linear threshold models and independent cascade models. Finally, a third model, ASCRIBE, will also be described due to its use in my work on empirical comparisons of contagion models.

Granovetter and Schelling were among the first to introduce linear threshold models [Granovetter, 1978; Schelling, 1978]. In these models, a node x in the network is influenced by each neighbor, $y \in Y$, based on a weight $w_{x,y}$ to each neighbor, such that $\sum\limits_{y \in Y} w_{x,y} \leq 1$. Each node x is then assigned a threshold θ_x based on a given probability distribution representing the total weighted fraction of x's neighbors that must be influenced in order for x to become influenced. The diffusion process proceeds deterministically in discrete time steps such that influenced nodes remain influenced and uninfluenced nodes are influenced if $\sum\limits_{z \in Z} w_{x,z} \geq \theta_x$, where Z is the set of x's influenced neighbors.

Independent cascade models, based on work in interacting particle systems from probability theory, accounts for each neighbor's influence upon a node independently and probabilistically. Specifically, for each neighbor $y \in Y$ of x, if y is successfully influenced in time step $t-1$, there is a probability $p_{y,x}$ that x will be influenced by y in time step t. These pairwise influence checks are performed independently and the probability for each interaction is a fixed parameter of the system that does not depend on past influence attempts. Each pairwise influence can occur only once, so that in subsequent rounds, y will not perform additional checks to probabilistically influence x.

The ASCRIBE model from Bosse et. al (2009) was specifically proposed to model the contagion of emotions in a crowd, but will be described in terms of influence in a graph for consistency. The primary difference with the previous two classes of models is that influence is modeled on a continuous spectrum from 0 to 1. Then, much like heat dissipation in physics, influence actually *transfers* from high influence nodes to neighbors of low influence, with all nodes in a neighborhood eventually converging towards a weighted average of the neighborhood's influence levels. The convergence is described in more detail in Bosse et. al (2009) and are omitted here due to space considerations.

Chapter 3: Related Work

3.1 Game-Theoretic Security Allocation

Game theory has been applied to a range of security problems, and different solution concepts and algorithms have been proposed. The most relevant of these works focus on algorithms for computing strong Stackelberg equilibria, which have been the most common paradigm in game-theoretic resource allocation. The ARMOR system at the Los Angeles International Airport uses the DOBSS algorithm from Paruchuri et al. (2008), which was shown to be superior to the previous state-of-the-art technique, Multiple LPs, introduced by Conitzer and Sandholm (2006). This first-generation system used this general-purpose Bayesian Stackelberg game solver to great effect, however, later applications would require far more scalability than was offered by DOBSS [Pita et al., 2008]. For example, the initial version of the IRIS system used by the United States Federal Air Marshal Service [Tsai et al., 2009] used the ERASER-C algorithm from Kiekintveld et al. (2009) and has since upgraded to the ASPEN algorithm from Jain et al. (2010). The PROTECT system developed for the U.S. Coast Guard [Shieh et al., 2012] features a new breed of algorithms that explicitly models human adversaries using quantal response based on work by Yang et al. (2012b). These works (and others) address numerous facets of the allocation problem, such as scalability, uncertainty [Jain et al., 2011b; Kiekintveld et al., 2011; Yin and Tambe, 2012], and human adversaries [Pita et al., 2012]. However, they do not directly apply to domains with a networked structure because they cannot handle the explosion in the number of actions that results.

3.2 Security Allocation in Network-based domains

Some work in game-theoretic security allocation have dealt with networked domains. Basilico et al. 2011, for example, develop abstractions to facilitate scalable algorithms for policy generation in patrolling security games. As mentioned previously, Jain et al. (2011a; 2013) and Yang et al. (2012a) follow up on my previous work on game-theoretic resource allocation in urban road networks [Tsai et al., 2010].

Halvorson et al. (2009) also use a double oracle approach to solve a version of a hider-seeker problem and, importantly, show an approximation guarantee for double oracle algorithms. Actions in these works, however, were all deterministically defined and did not feature a probabilistic contagion component. The contagion component adds a dimension of immense difficulty to security allocation problems that makes previous methods unusable for anything but the smallest of domains.

Aside from the literature on game-theoretic allocation methods, my urban network security work is also based on insights from network interdiction [Washburn and Wood, 1995; Israeli and Wood, 2002]. These are the special case of my urban network model when there is a single target, or — equivalently — all targets have identical values. For such games, Washburn and Wood (1995) give an algorithm finding optimal strategies for both players based on Min-Cut computations. However, different target values and a general-sum game can cause their algorithm to perform arbitrarily poorly.

3.3 Influence Maximization

This ‘spreading’ aspect of networked competitive contagion scenarios is very closely related to influence maximization. In influence maximization, a player attempts to optimize a selection of beginning ‘seed’ nodes from which to spread his influence in a known graph. This problem saw its first treatment in computer science as a discrete maximization problem by Kempe et al. (2003) who proposed a greedy approximation, followed-up by numerous proposed speed-up techniques [Chen et al., 2010b; Kimura et al., 2010; Leskovec et al., 2007]. Although these are one-player games, I draw from their techniques to address efficiency issues in my work.

A competitive variant of these types of games are inoculation games, which typically feature a defender attempting to protect nodes in a graph and, usually, a random outbreak of a disease on a node in the graph. These games model nature as the adversary, which chooses an initial set of nodes with some predefined probability distribution that the defender is optimizing against [Aspnes et al., 2005, 2007; Chen et al., 2010a; Kumar et al., 2010]. Variations on this include distributed inoculation games where each node acts independently, in which results such as price of anarchy are generally considered [Aspnes et al., 2005; Chen et al., 2010a]. Inoculation games do not typically include an optimizing adversary, amounting to only an attacker *or* defender best-response problem.

There have been a number of recent works focused on competitive models of influence maximization where both players attempt to maximize influence in a social network where only one player can hold influence over each node [Bharathi et al., 2007; Kostka et al., 2008; Borodin et al., 2010]. Bharathi et al., for example, examine the first-mover and last-mover best-responses in the context of competitive influence maximization. Although they prove the last-mover's best-response is submodular and amenable to the greedy approximation scheme first proposed by Kempe et al., they do not provide a general algorithm for determining equilibrium strategies. Borodin et al. examine threshold models and show NP-hardness of approximations better than a square root of the optimal for a broad class of such models. Again, however, they do not provide a game-theoretic treatment of the problem and no algorithms for equilibrium strategy generation are explored.

Another class of competitive influence maximization problems are influence blocking maximizations. Influence blocking maximization problems again feature two players, each of whose actions spread across the given network. Instead of both players attempting to maximize, however, influence blocking maximization has one player trying to minimize the other player's influence spread instead of attempting to maximize his own. Such problems have been explored with both independent cascade and linear threshold models of propagation [Budak et al., 2011; He et al., 2012]. Both of these works only explored the defender's best-response problem.

3.4 Beyond Strategy Optimization

Social scientists have studied other important aspects of influence in social networks extensively. I will briefly mention a few such areas here. In social psychology, Kelman's seminal work noted three broad categories of social influence [Kelman, 1958] and was followed by a plethora of related research examining the driving forces behind specific examples of social influence [Wood et al., 1994; Fowler and Christakis, 2008]. Consensus formation in societies has been a major area of research in political science that also contributes heavily to the study of influence and persuasion [DeGroot, 1974; Lupia and Mccubbins, 1998]. At the intersection with computer science, experimental work on the impact of network structure on consensus formation and cooperation is particularly relevant to the networked contagion games I examine [Kearns et al., 2006; McCubbins et al., 2008; Cosley et al., 2010; Suri and Watts, 2010; Enemark et al., 2011]. Finally, with the growing use of online retailers and user recommendations, marketing studies have begun shedding light on the nuances of the contagion of purchasing decisions [Goldenberg et al., 2001; PHELPS et al., 2004; Kiss and Bichler, 2008; Trusov et al., 2009; Yang et al., 2012c].

Although only a very limited set of works is mentioned, the literature in the social sciences on influence, network effects, and persuasion is far more vast than can be fully addressed here. As would be expected, these works do not study strategy generation for these domains. Instead, they focus on understanding mechanisms for facilitation and constitute directions of study orthogonal to my own.

3.5 Emotional Contagion

Seminal works in social psychology first began the discussion around emotional contagion. In particular, Hatfield et al. (1994) first codified the observed phenomena that were just beginning to receive researcher attention. Follow-up work by the co-authors as well as in related fields such as [Barsade and Gibson, 1998; Grandey, 2000; Pugh, 2001] in managerial and occupational sciences continued to detail the effects of the phenomenon in new domains. Recently, there have been works beginning to quantify emotional contagion and explore cross-cultural variations in attributes that affect emotional contagion [Doherty, 1997; Lundqvist, 2008]. Additionally, researchers in related fields such as managerial and occupational sciences continued to detail the effects of the phenomenon in new domains [Barsade, 2002; Small and Verrochi, 2009]. A large body of social psychological studies of emotional contagion also feature an image or video of only a person's face as the origin of the contagion [Hess et al., 1998; Small and Verrochi, 2009; Wild et al., 2001].

From a computational perspective, work from VU University and the University of Pennsylvania are two of the most recent models of emotional contagion upon which a few follow-up works have been based [Durupinar et al., 2008; Bosse et al., 2009a,b; Durupinar, 2010; Bosse et al., 2011]. The ASCRIBE model resembles heat dissipation models found in basic physics wherein each substance has its own heat dissipation rate and heat absorption rate. The model from Durupinar draws inspiration from a long line of contagion models [Dodds and Watts, 2005; Kermack and McKendrick, 1927] that was popularized in the diffusion of innovations [Rogers, 1962] literature and has also seen heavy use in other types of social (e.g., belief, behavior, idea) contagion [Schelling, 1973].

There also exists a large body of work on the interaction between virtual agents and humans [Gratch et al., 2006; Wang and Gratch, 2009; de Melo et al., 2011]. The entire area of virtual rapport [Gratch et al., 2006; Wang and Gratch, 2009], for example, focuses on user opinions of the virtual agents and their interaction. The primary goal is to create agents that users enjoy, appreciate, and relate to. Recent work has looked at the impact of agent expressions in a strategic negotiation setting [de Melo et al., 2011] as well. However, their work focuses on the behavioral impact of varying the intent of agent expressions on user behavior without examining the emotional impact on users.

Chapter 4: Game-Theoretic Resource Allocation in Networks

This chapter introduces my extension of game-theoretic resource allocation techniques into networked domains with no contagion effects. In particular, the challenge of urban road network security, where police must place checkpoints on roads to intercept would-be attackers from reaching their destinations is naturally modeled as a graph. One key challenge in most networked domains is the massive action sets that both players inevitably inherit due to the compact representation of the problem as a network. In this particular setting, every path from an entry point to a target is an attacker action and every set of r or fewer edges, where r is the maximum number of checkpoints, is a defender action. Since attacker actions grow exponentially with the size of the network and defender actions grow exponentially with r, existing methods quickly become intractable when applied to real-world domains [Kiekintveld et al., 2009; Paruchuri et al., 2008; Conitzer and Sandholm, 2006]. With both players having an exponential number of actions, new techniques had to be developed. This work led to a line of research into scalable techniques to address the massive scale of both players' action sets that exists in any networked problem [Jain et al., 2010; Yang et al., 2012a; Jain et al., 2013].

This chapter presents the first techniques in this line of inquiry, introducing the RANGER algorithm and accompanying sampling techniques Radius and Comb Sampling to generate effective deployment strategies with efficiency guarantees. Section 4.1 formally describes the game model being addressed. Section 4.2 begins the presentation of the algorithm by introducing RANGER, a linear program that reasons over the marginal distribution of the defender's mixed strategy to efficiently calculate a heuristic solution. Since deployment requires a a sampling technique for the marginal distribution generated, Section 4.3 introduces two novel sampling techniques that, when combined with RANGER, produce extremely high quality deployment strategies, as shown in the experiments in Section 8.3.

4.1 Problem Definition

A graph-based security game models an attacker and a defender who take actions on a graph $G = (V, E)$, with $n = |V|$ nodes and $m = |E|$ edges. The attacker starts at a source node $s \in S \subseteq V$ *of his choosing* and travels along a path in an attempt to reach a target $t \in T \subseteq V$. The attacker's pure strategies, $\omega_C \in \Omega_C$, are thus all s-t paths from some source s to some target t. Since the attacker can choose which source to enter from, for analysis, I merge all sources into a *single source* without loss of generality by reforming the graph so that all original source-incident edges be incident to the new source. The defender tries to capture the attacker before he reaches a target by placing up to r resources on *edges* of the graph. The defender's pure strategies are subsets of r or fewer edges, designated $\omega_R \in \Omega_R$. Assuming that the defender plays ω_R and the attacker ω_C, the attacker is captured whenever $\omega_R \cap \omega_C \neq \emptyset$, and succeeds in his attack when $\omega_R \cap \omega_C = \emptyset$.

Payoffs depend on the target attacked and whether the attacker is captured or not. If an attack on t is *successful*, the defender receives $D_S(t)$ and the attacker receives $A_S(t)$. If the attack is *unsuccessful* the attacker receives $A_U(t)$ and the defender receives $D_U(t)$. I make the natural restrictions that $D_S(t) \leq D_U(t)$ and $A_S(t) \geq A_U(t)$. Note that there is no restriction on the game being zero-sum.

In a world of increasingly sophisticated and determined attackers, a good defender strategy must take into account that the attacker will observe and exploit patterns in the defender's behavior. In other words, it is natural to assume that the defender must commit to a strategy first, which the attacker can observe and base his strategy on. Thus, the game is naturally modeled as a Stackelberg game, an approach also taken (for the same reasons) in past work in security settings [Kiekintveld et al., 2009; Paruchuri et al., 2008]. The defender is modeled as the *leader* and moves first, by selecting a mixed strategy $\rho_R \in P$ that assigns a probability to each pure strategy $\omega_R \in \Omega_R$. The attacker is the *follower* and chooses a strategy after observing the defender's mixed strategy. There is always a pure-strategy best response for the attacker, so I restrict the attacker to pure strategies without loss of generality. Thus, the attacker's Stackelberg strategy is a function $f : \rho_R \mapsto \omega_C$; let F denote the set of all such functions. For any pair of strategy profiles (ρ_R, f), the expected rewards for the defender (R_D) and attacker (R_A) are given by:

$$R_D(\rho_R, f) = p \cdot D_U(t) + (1-p) \cdot D_S(t) \tag{4.1}$$

$$R_A(\rho_R, f) = p \cdot A_U(t) + (1-p) \cdot A_S(t) \tag{4.2}$$

where t is the target at the end of the path specified by $f(\rho_R)$, and p the probability that the attacker is captured on the path to t given the defender's strategy ρ_R. The solution concept I use, a *Strong Stackelberg equilibrium*, is formally defined as follows:

Definition 1. *A pair of strategies* (ρ_R, f) *form a* Strong Stackelberg Equilibrium *(SSE) if they satisfy the following:*

1. *The leader plays a best strategy:*

   $R_D(\rho_R, f) \geq R_D(\rho'_R, f)$ *for all distributions* ρ'_R.

2. *The follower plays a best response:*

   $R_A(\rho_R, f) \geq R_A(\rho_R, f')$ *for all functions* $f' \in F$.

3. *The follower breaks ties in favor of the leader:*

   $R_D(\rho_R, f) \geq R_D(\rho_R, f')$ *for all functions* $f' \in F$ *that are best responses to* ρ_R.

While counterintuitive, Condition 3 is justified because the leader can often induce the favorable strong equilibrium by selecting a strategy arbitrarily close to the equilibrium that causes the follower to strictly prefer the desired strategy [von Stengel and Zamir, 2004]. SSE are more commonly used in the literature than their counterpart, Weak Stackelberg Equilibria, which does not require Condition 3. Also, SSE are guaranteed to exist and uniquely define the strategy and payoffs for the leader [Leitmann, 1978; Breton et al., 1988]. The optimal policy for the defender in a graph-based security game is the SSE strategy. Unfortunately, as Ω_C has size $\Theta(m^r)$, and Ω_R has size exponential in n, existing methods for computing SSE do not scale to realistic problem sizes.

I introduce a new solution concept, *Strong Stackelberg Approximation* (SSA) that avoids these difficulties by introducing an approximation for follower. SSA assumes that the attacker observes and plays a best response to the marginal distribution of the defender's strategy (i.e., the aggregate coverage probability on each edge) rather than the joint distribution ρ_R. This assumption can be motivated as either a limitation on the observations of the attacker, who may not have the capability to observe the joint distribution, or as a computational limitation of the attacker who many not be able to compute an optimal response to the joint distribution. Specifically, SSA replaces R_A in condition 2 of SSE with the approximation:

$$\hat{R_A}(\rho_R, f) = \hat{p} \cdot A_U(t) + (1 - \hat{p}) \cdot A_S(t), \tag{4.3}$$

where $\hat{p} = \min(\sum_{e \in f(\rho_R)} x_e, 1)$ is an approximate capture probability on path $f(\rho_R)$. $x_e = \sum_{\omega_R \in \Omega_R, e \in \omega_R} \rho_R(\omega_R)$ is the marginal probability associated with edge e, where $\rho_R(\omega_R)$ is the probability associated with ω_R in ρ_R. I denote this mapping by $z : P \mapsto \mathcal{X}$ and the marginal distribution as $\vec{x} = \langle x_e \rangle$.

4.2 RANGER

Now I introduce RANGER (Resource Allocation for Network-based Games with Efficient Representations), an efficient method for finding an optimal set of marginal checkpoint probabilities for the defender. By reasoning over $\vec{x}$, it avoids the exponential size of the defender space. It then approximates the probability of capturing an attacker on a path by summing the marginal probabilities on the edges in the path. While this is not correct in general, joint distributions can be reconstructed in a ways that make it correct under specific circumstances. This yields an efficient approximation for the optimal attacker path to each target and avoid reasoning over the exponential space of *all* paths.

In the RANGER MILP below, x_e is the marginal probability of placing a checkpoint on edge e. The d_v are, for each vertex v, the minimum sum of checkpoint probabilities along any path from a source to v. This is enforced by the constraints (4.6)-(4.8). Constraint (4.11) enforces that at most r checkpoints are placed. The vector $\vec{q}$ describes the probability of the attacker attacking each target. The fact that the attacker picks exactly one target deterministically is enforced by constraints (4.9)-(4.10). Constraint (4.5) forces $\vec{q}$ to be the vector choosing the highest expected payoff target for the attacker, and Constraint (4.4) captures the payoff for the defender. (4.4) and (4.5) rely on M being a large constant, so that they are binding only for the target t selected by the attacker. Together, Constraints (4.4) and (4.5) imply that the leader and defender strategies are mutual best responses in any optimal solution.

$$\begin{aligned}
\text{Maximize} \quad & R \\
\text{s.t.} \quad & R - (1 - d_t) \cdot D_S(t) \\
& - d_t \cdot D_U(t) \leq (1 - q_t) \cdot M && (4.4) \\
& 0 \leq \hat{R_A} - (1 - d_t) \cdot A_S(t) \\
& - d_t \cdot A_U(t) \leq (1 - q_t) \cdot M && (4.5) \\
& d_s = 0, \forall s \in S && (4.6) \\
& d_v \leq \min(1, d_u + x_e), \forall e = (u, v) && (4.7) \\
& 0 \leq x_e \leq 1, \forall e \in E && (4.8) \\
& q_t \in \{0, 1\} && (4.9) \\
& \sum_{t \in T} q_t = 1 && (4.10) \\
& \sum_{e \in E} x_e \leq r && (4.11)
\end{aligned}$$

Theorem 1. *A pair of attacker and defender actions $(\vec{x}^*, \vec{q}^*)$ that is optimal for the RANGER MILP corresponds to at least one SSA against an approximating attacker.*

Proof. Define f^* as the function specifying an attacker action for every leader strategy, where the attacker action ($\vec{q}$) is determined by RANGER if we fix the leader coverage strategy to a given $\vec{x}$. Since RANGER maximizes the defender reward, the attacker will implicitly choose the best-response that maximizes the defender's reward, satisfying Condition 3 of an SSA. Constraint 2 specifies that the attacker's response maximizes his reward assuming he is an approximating attacker. Thus, by construction, $(\vec{x}, f^*)$ satisfies Condition 2 of an SSA for all $\vec{x} \in \mathcal{X}$.

Let $R(\vec{x})$ is the R found by RANGER when fixing $\vec{x}$. Since d_v is an overestimate of the actual capture probability, $\max_{\vec{x} \in \vec{\mathcal{X}}} R(\vec{x}) \geq \max_{\rho_R \in P} R_D(\rho_R, f^*)$, or $R_{RANGER} \geq R_{SSA}$, where R_{RANGER} is the defender reward found by RANGER and R_{SSA} is the defender reward at SSA. As Theorem 3 shows, we can always construct a $\rho_{R:\vec{x}^*}$ corresponding to $\vec{x}^*$ such that $R(\rho_{R:\vec{x}^*}) = R_{RANGER}$ against an approximating attacker. Since $R_{SSA} \geq R(\rho_R)$ for all ρ_R, $R_{SSA} \geq R_{RANGER} \geq R_{SSA}$, so $\rho_{R:\vec{x}^*}$ is an SSA strategy, satisfying Condition 1. □

RANGER is an exponential contraction of both the attacker and defender strategy spaces as discussed. This can be seen most clearly by noticing that RANGER has variables polynomial in the number of vertices ($|S|$, length of $\vec{q}$, number of d_v) and edges (x_e) in the graph. Prior formulations would have had m^r variables for the defender and exponentially many integer variables ($\vec{q}$) to explore.

4.3 Reconstructing Joint Distributions

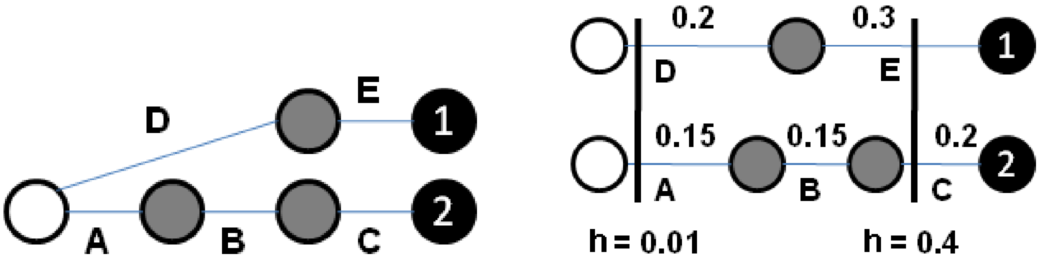
Using the marginal distribution calculated by RANGER, we must generate a joint distribution over $\mathcal{L}$ to sample from or create sampling procedures directly from $\vec{x}^*$. In so doing, we must adhere as closely to the assumption inherent in RANGER as possible - that no ω_C sampled will ever have two checkpoints on a single s-t path.

This problem has been modeled as a Random Assignment problem, as studied in Combinatorial Optimization, where we want to assign a objects to a agents given a bistochastic matrix describing the desired probability for each agent to receive each object [Birkhoff, 1946; von Neumann, 1953; Budish et al., 2009]. The bistochastic matrix can be used to model the marginals of our problem of assigning checkpoints to edges, but existing methods cannot efficiently account for the dependencies that exist in our domain.

The Birkhoff-von Neumann Theorem [Birkhoff, 1946; von Neumann, 1953] proves that the basic version of this problem always has a solution, implying that there always exists a joint distribution underlying a marginal distribution when there are no constraints. Recent work has extended the Birkhoff-von Neumann Theorem to domains with specific constraints beyond the original problem and proven that these extensions constitute a maximal domain for the theorem [Budish et al., 2009]. Graph-based security games, however, fall outside of this work because the impact of placing one checkpoint on another cannot readily be captured in a bistochastic matrix. Indeed, some marginal distributions cannot be met by any joint distribution. Thus, I introduce two novel algorithms that efficiently produce joint distributions under specific constraints of the problem.

4.3.1 Radius Sampling

Radius Sampling (RS) is a sampling procedure that guarantees RANGER's assumption is met for single-source graphs with certain conditions we describe shortly. This means that the probability for capture along any s-v path will be at least d_v. The idea is to define rings of different radii,



(a) Example graph

(b) Laying out paths with capture probabilities

Figure 4.1: Radius Sampling example graph.

h, such that each ring corresponds to a set of edges. For any $h \geq 0$, define the ring of radius h around s as $R_h := \{e = (u, v) | d_u \leq h < d_v\}$, i.e., the set of edges from a node with probability of capture at most h from s to a node with probability of capture more than h from s.

Define $\alpha := \int_0^\infty |R_h| dh$ (a normalization constant), and the density function $\phi(h) := \frac{|R_h|}{\alpha}$. Notice that

$$\alpha \quad = \quad \sum_{e=(u,v)} (d_v - d_u) \quad \leq \quad \sum_e x_e \leq r. \tag{4.12}$$

The algorithm works as follows: Choose a radius h from $[0, \infty]$ according to the density function ϕ. Now, choose r of the edges in R_h uniformly at random (or all edges in R_h if $|R_h| \leq r$). Place checkpoints on these edges. Call the resulting set ω_{radius}. Notice that both h and ω_{radius} are random variables; ω_{radius} is a set of at most r edges.

The algorithm generates a joint distribution for the defender based on the given marginal probabilities under the condition that for all radii h, either $R_h = \emptyset$ or $|R_h| \geq r$. This simply means that in any ring we might choose, there must be at least r edges, otherwise we will have unassigned checkpoints, and we will not be able to meet the capture probabilities desired. The method can still be used with some augmentations, but there will be no guarantees regarding the joint distribution generated.

As an example, consider the graph given in Figure 4.1a with 6 vertices, 2 targets, one source, and disjoint paths to the targets as show in Figure 4.1a. Visually, we can think of the x_e values as laying out the paths as shown in Figure 4.1b, where marginal probabilities are given for each edge. Then we randomly select an h. For example, if $h = 0.01$, we would place a checkpoint on edge A or edge D with probability $1/2$ each. Similarly, if $h = 0.4$, we would choose C or E with probability $1/2$ each.

The idea of RS is to ensure that for no optimal path from s to some other node do we ever sample multiple edges, which was the implicit assumption for RANGER to be accurate. For non-optimal paths, the probability for capture will still be at least the d_v predicted by RANGER, but this may not be the sum of the marginals along the path. If $|R_h| \geq r$ for all h with $|R_h| > 0$, RS achieves RANGER's expected reward value.

Theorem 2. *If for all h, $|R_h| \geq r$ or $|R_h| = 0$, then RS ensures that for any path P ending at node v, the probability of capturing an attacker on P is at least d_v.*

The proof follows readily from the lemma below.

Lemma 1. *Under the assumptions of Theorem 2, let ω_C be any s-v path and w the node maximizing d_w among all nodes on ω_C. The capture probability along ω_C is at least d_w.*

Proof. I prove the lemma by induction on $|\omega_C|$, the number of edges on path ω_C. In the base case $|\omega_C| = 0$, the only node v with a path from s is s itself, and the statement holds because $d_s = 0$.

For the inductive step, let ω_C be a path of length $\ell + 1$ and $e = (v', v)$ the last edge of ω_C. Let $\omega'_C = \omega_C \setminus \{e\}$ be the path of length ℓ from s to v', and w' the node on ω'_C maximizing $d_{w'}$. By Induction Hypothesis, $\text{Prob}[\omega_{radius} \cap \omega'_C \neq \emptyset] \geq d_{w'}$.

Distinguish two cases. If $d_{w'} \geq d_v$, then

$$\begin{aligned} \text{Prob}[\omega_{radius} \cap \omega_C \neq \emptyset] &\geq \text{Prob}[\omega_{radius} \cap \omega'_C \neq \emptyset] \\ &\geq d_{w'} \geq d_v, \end{aligned}$$

implying the claim.

If $d_v > d_{w'}$, then consider the event $\mathcal{E} = [h > d_{w'} \text{ and } e \in \omega_{radius}]$. In essence, $\mathcal{E}$ is the event when we include e in the ω_{radius} and we can be sure that no edge from ω'_C is also sampled. The probability of $\mathcal{E}$ is

$$\begin{aligned} \int_{d_{w'}}^{d_v} \text{Prob}[e \in \omega_{radius} \mid h = x]\phi(x)dx &= \int_{d_{w'}}^{d_v} \tfrac{r}{|R_x|} \cdot \tfrac{|R_x|}{\alpha} dx \\ &= \int_{d_{w'}}^{d_v} \tfrac{r}{\alpha} dx \\ &\geq d_v - d_{w'}. \end{aligned}$$

Here, I substituted the definitions of the sampling process, and then used that $\frac{r}{\alpha} \geq 1$ from Equation (4.12).

Whenever ω_{radius} intersects ω'_C, by definition, we must have that $h \leq d_{w'}$ (because no edge $e' \in \omega'_C$ is in R_h for $h > d_{w'}$). Thus, the events $\mathcal{E}$ and $[R_h \cap \omega'_C \neq \emptyset]$ are disjoint, and

$$\begin{aligned} \text{Prob}[R_h \cap \omega_C \neq \emptyset] &\geq \text{Prob}[[R_h \cap \omega'_C \neq \emptyset] \cup \mathcal{E}] \\ &= \text{Prob}[R_h \cap \omega'_C \neq \emptyset] + \text{Prob}[\mathcal{E}] \\ &\geq d_{w'} + (d_v - d_{w'}) \\ &= d_v. \end{aligned}$$

The penultimate step used the induction hypothesis as well as the inequality $\text{Prob}[\mathcal{E}] \geq d_v - d_{w'}$ derived above. □

4.3.2 Comb Sampling

Radius Sampling provides a guarantee that is stronger than needed if we assume an approximating attacker. Assuming an approximating attacker can be justified by the fact that the joint distribution is much harder to observe than the marginals, and, even if an attacker knew the full joint distribution, it is unclear he could take full advantage of this information, since computing the best path is then an NP-hard problem as can be shown by a reduction from GRAPH 3-COLORING (included in Appendix A).

If the attacker adds marginal probabilities and chooses a path accordingly, then a simpler procedure works, which I call *Comb Sampling* (CS). First, as long as we ensure that the marginal probabilities of the samples match $\vec{x_e}$, the attacker will choose the s-t path ω_C predicted by RANGER. Then, we need only ensure that the capture probability on ω_C is d_t, which can be done by never sampling any $\omega_R \in \Omega_R$ that has two edges on ω_C. The approach can be conceptualized as laying the marginals in a row, end to end, and randomly placing a comb onto the row to sample an $\omega_R \in \Omega_R$.

Let $e_1, \ldots, e_t$ be the edges on the attack path (in arbitrary order), and $e_{t+1}, \ldots, e_m$ the remaining edges, in arbitrary order. For each $1 \leq j \leq m$, let $X_j = \sum_{i<j} x_i$, and define the interval $I_j = [X_j, X_j + x_j)$. Because $\sum_i x_i = r$ (w.l.o.g.), the I_j form a disjoint cover of the interval $[0, r)$. Now generate a deployment, ω_{comb}, as follows: Pick a number $y \in [0, 1)$ uniformly at random, and include in ω_{comb} all edges e_j such that $y + k \in I_j$ for some integer k. In other words, include exactly the edges which “own” the intervals containing the points $y, y + 1, y + 2, \ldots, y + r - 1$. This samples exactly r edges.

Theorem 3. *Given a marginal distribution over edges $e \in E$, CS will exactly meet the marginal probabilities, x_e, for all edges.*

Proof. Consider any edge e_j, and two cases. If $I_j \subseteq [k, k+1)$ for some k (i.e., I_j contains no integer point), then e_j is included if and only if $k + y \in I_j$, which happens with probability $|I_j| = x_j$. On the other hand, if $I_j = [X_j, k) \cup [k, X_j + x_j)$, then e_j is included if and only if $y + k - 1 \in [X_j, k)$ or $y + k \in [k, X_j + x_j)$; because $x_k \leq 1$, this happens with probability $(k - X_j) + (X_j + x_j - k) = x_j$. □

Theorem 3 ensures that the attacker will follow the path predicted by RANGER. Now consider the attack path $\omega_C = \{e_1, \ldots, e_t\}$. If $\sum_{j=1}^{t} x_j \geq 1$, then for any y, some edge $e_j \in P$ will be included, so the attacker is always captured. Otherwise, an edge from ω_C is included if and only if $y < \sum_{j=1}^{t} x_j$, which happens with probability $\sum_{j=1}^{t} x_j$, i.e., the sum of marginals on the attack path.

4.4 Experiments

I first evaluate the runtime performance of RANGER against other state-of-the-art solution techniques. Then, I examine the quality of the solutions generated based on the sampling methods discussed. Finally, I compare the performance of RANGER and other algorithms on a real-world problem by computing policies for use in Mumbai.

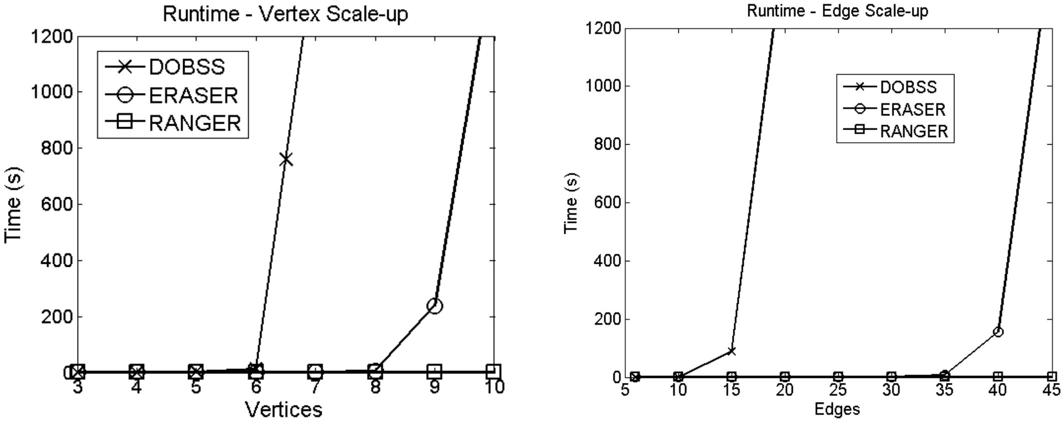
4.4.1 Runtime Comparison

First, I evaluate the runtime performance of RANGER against the fastest-known exact algorithm for solving general Bayesian Stackelberg games, DOBSS [Paruchuri et al., 2008], as well as a faster solver for security games, ERASER [Kiekintveld et al., 2009]. DOBSS serves as a benchmark, since it provides the optimal solution against exact attackers. ERASER exploits structural properties that exist in many security domains to create a compact game representation. However, in order for the solution to be correct, it also requires that defender actions be independent from each other, which is not the case in this domain, since placing two checkpoints on one path will violate this. Nevertheless, ERASER serves as another approximate algorithm that runs much more efficiently than DOBSS, so I compare RANGER's runtime against it.

In the first set of experiments, I use complete graphs with the number of vertices varying from 3 to 10 (3 to 45 edges). Every experiment has one source, one target, and two checkpoints. Figure 4.2a shows scaling of each method's runtime with respect to the number of vertices. The x-axis shows the number of vertices in the graph and the y-axis shows runtime in seconds. DOBSS is only able to solve the problem with up to 6 vertices within the time limit, while ERASER can only handle up to 9 vertices. RANGER is capable of solving games with 400 vertices within the time limit (not shown). Figure 4.2b shows the second set of experiments, where I generated 10 random graphs of each edge-size, allowing other parameters to vary, yielding a variety of graph types for each size. Again the problem quickly becomes intractable for DOBSS.

4.4.2 Quality Comparison

I next evaluate the quality of the algorithms discussed against approximating attackers and exact attackers. Against approximating attackers, we showed that RANGER's reward can be achieved precisely using CS, but CS and RS's behavior against exact attackers cannot be guaranteed. Thus, I evaluate their quality against exact attackers here. As a benchmark, I also compare against a simple Independent Sampling strategy, wherein for each checkpoint, edge e is chosen mutually



(a) Scale up for complete graphs.

(b) Scale up by edges

Figure 4.2: Runtimes for RANGER, ERASER, and DOBSS.

exclusively with probability $\frac{x_e}{r}$. The DOBSS method is only guaranteed to be optimal for exact attackers, so I check its quality against approximating attackers, labeling these results DOBSS Marginal.

I generated 1,000 random graphs and compared the results found by DOBSS and RANGER as well as the attainable reward after using each of the sampling methods. Each graph has a random configuration of vertices ($|V|$ = 3 - 20), edges (5 - 20), checkpoints (1-20), sources ($|S| = 1 - (|V| - 1)$), targets (1 - $|V| - |S|$), and targets had defender/attacker favorable rewards of between 1 and 10 and loss rewards from -1 to -10. Again, games were not necessarily zero-sum. Games were restricted in size to ensure that DOBSS could efficiently compute an optimal policy.

For each of 1,000 random graphs, I ran each sampling method on the marginals produced by RANGER, calculate the *actual* capture probabilities given the joint distribution, find the exact attacker's optimal choice, and calculate the defender's expected reward. I compare these rewards against DOBSS to evaluate how far we are from optimal.

For DOBSS Marginal, I calculate the marginal probabilities from the joint distribution given and do a optimal-path calculation to determine an *approximating attacker*'s optimal action and the corresponding reward for the defender.

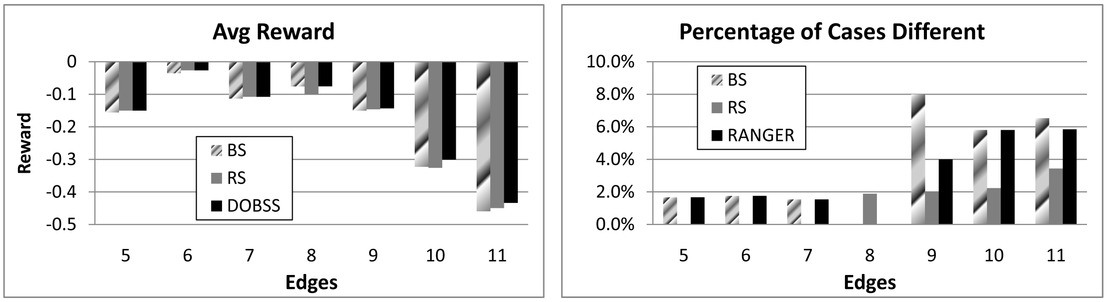
For Independent Sampling, the actual probability of capture will be $1-(1-p/r)^r$, where p is the sum of RANGER's marginal probabilities on the edges on the path P. To see this, recall that since r checkpoints are placed independently, and the probability that the j^{th} checkpoint is not on P is $1-\sum_{e\in P} x_e/r = 1-p/r$, the probability that there is no checkpoint on P is $(1-p/r)^r$. Thus, the probability for capture is $p' = 1-(1-p/r)^r$.

For Radius Sampling, I find all rings R_h and the probability for selecting each. Then, I calculate the probability of selecting edges within each ring. From these probabilities, I then obtain the marginal probabilities for each edge; when the conditions of Theorem 2 are violated, these marginal probabilities might be less than RANGER's values. Finally, I add the marginal probabilities on any path to find the actual probability of capture.

For Bucket Sampling, recall that we have a joint probability distribution. For each path, I determine which joint actions place a checkpoint on the path and sum the probabilities associated with these actions. In general, this would be an exponential procedure, since there are exponentially many possible joint actions and exponentially many paths that must be calculated. However, Bucket Sampling produces a joint distribution using only $O(m)$ joint actions, making experiments relatively efficient. In practice, we could randomize the order in which the x_e's are processed to create a more complex joint distribution, but experimental evaluation would become computationally infeasible.

For each of the sampling methods, the value reported is the expected defender reward based on the *exact attacker*'s action as determined by the procedures outlined previously. The DOBSS value reported is simply the expected defender reward calculated by the corresponding MILP.

Independent Sampling performed statistically significantly worse than every other method. RS was able to meet RANGER's reported reward even against exact attackers in all but 57 cases, while CS met it in all but 56 cases. DOBSS Marginal never differed from DOBSS, but we can do better than DOBSS against an approximating attacker, as RANGER achieved slightly higher rewards in some cases. Thus, if our assumption about an approximating attacker holds, it would be beneficial to use RANGER instead of DOBSS.



(a) Avg reward of games where DOBSS' reward was not met

(b) Percentage of games where DOBSS' reward was not met

Figure 4.3: Breakdown by game size.

Figure 4.3a shows rewards averaged across games of each game size. Rewards for larger games worsen for every technique, since it becomes harder to defend. In Figure 4.3b, I show the percentage of cases in which CS, RS, and RANGER reward values differ from DOBSS, by the number of edges in the graph. In all cases shown, the rewards using our sampling algorithms differed from the DOBSS reward for no more than 8% of the games.

4.4.3 Mumbai

As a real-world trial, I use my algorithms to create security policies for the southern tip of Mumbai, shown in Figure 4.4, which has historically been an area of heavy terrorist activity. The region is modeled as a graph with 35 nodes and 58 edges. Attackers can potentially enter from *any* entry node, chosen based on historical and likely entry points. Target nodes were chosen based on historical attacks. Four configurations were tested, each with the same set of 5 targets and varying locations for 4-5 sources. Results are shown in Figure 4.5 .

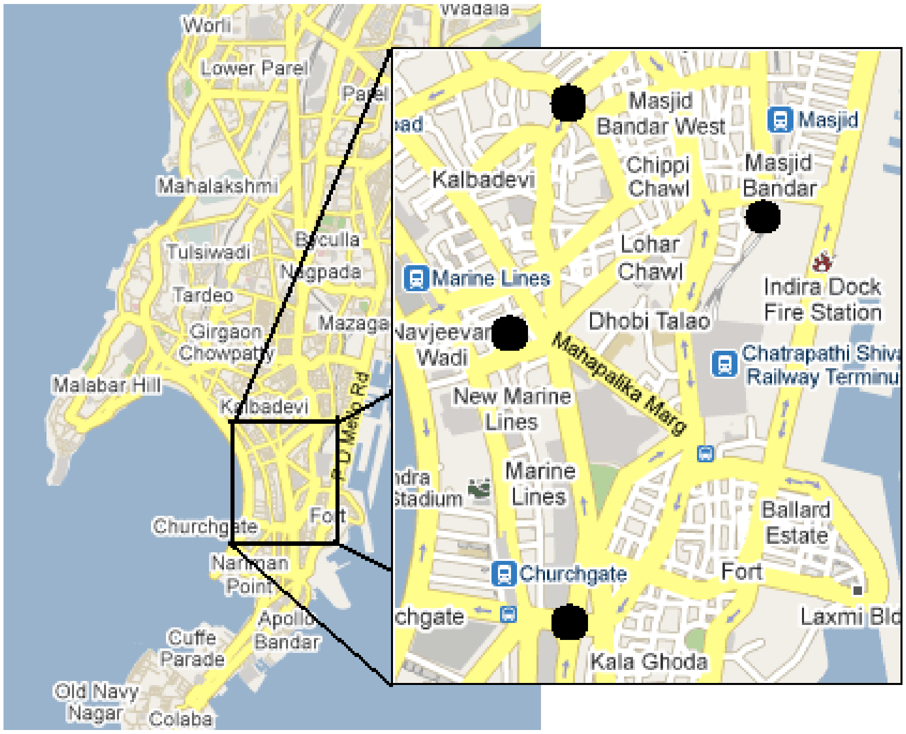


Figure 4.4: Example strategy for southern Mumbai.

DOBSS is unable to solve even the simplest case within the 20-minute limit; thus, I include only Comb Sampling and Radius Sampling's expected reward, Minimum Cut, as well as three natural defense strategies, *Uniform Random*, *Entry-Incident* and *Weighted-Target-Incident*.

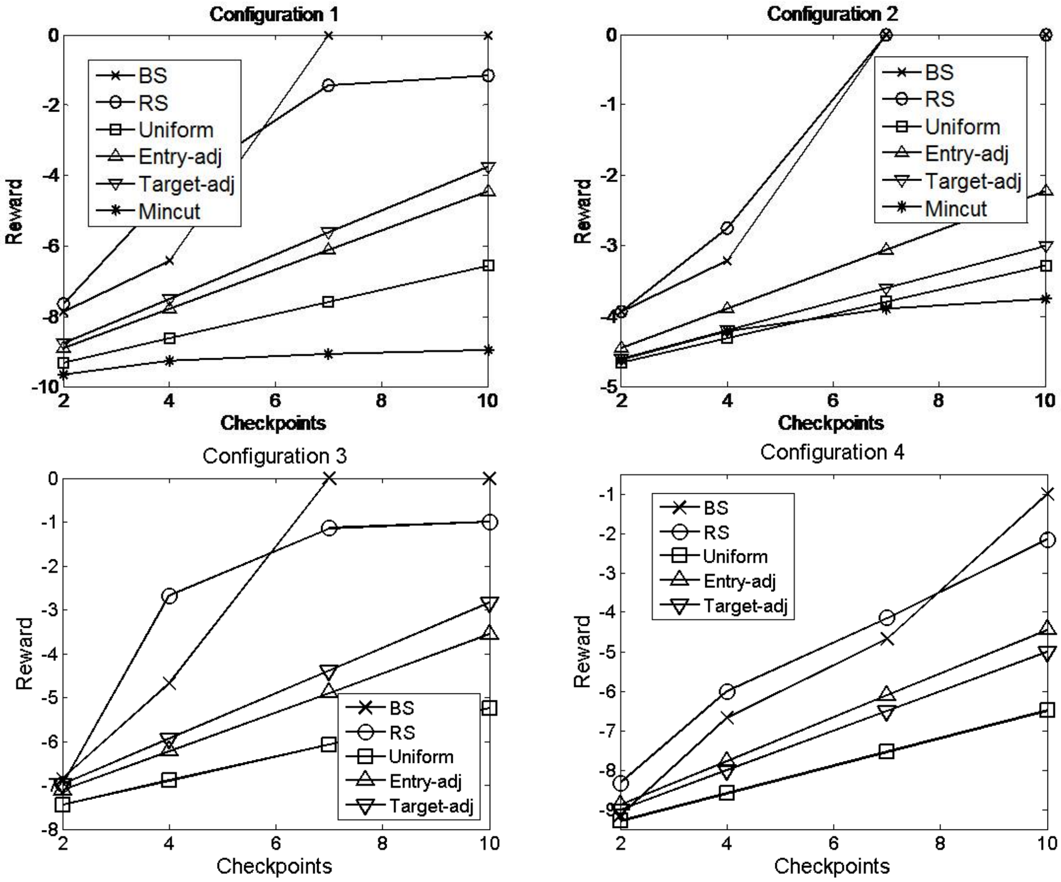


Figure 4.5: Mumbai results

Minimum Cut, as introduced by [Washburn and Wood, 1995], contracts all sources into a super-source and all targets into a super-target and finds the minimum cut on the resulting graph, uniformly randomizing resources across it. This effectively ignores target value variation, but is extremely efficient. Uniform Random places checkpoints uniformly randomly across all edges in the graph. Entry-Incident places checkpoints on edges incident to entry nodes with equal probability. Weighted-Target-Incident places checkpoints on edges incident to target nodes, weighted according to their payoff value. The x-axis labels show the number of checkpoints allowed for the instance and the y-axis shows the expected reward. RANGER ran in roughly 0.2 seconds in all trials.

In all four configurations, the predefined strategies perform substantially worse than Comb Sampling and Radius Sampling as the number of checkpoints increases. Minimum Cut, which does not prioritize higher value targets, performs worse than the Uniform Random strategy which coincidentally places more coverage on higher value targets in this domain because there are more roads to them.

RANGER actually exploits resources differently and to better effect. For example, in one configuration, with 2 checkpoints RANGER placed probability on 7 streets, with 10 checkpoints it used 16 streets, with 20 checkpoints it used 14 streets. Also, note the non-trivial edge selection evident in the solution shown in Figure 4.4's. Although I cannot provide guarantees on the performance of Radius or Comb Sampling against an exact attacker in general, the techniques yield mixed strategies of very high quality in practice that outperform the sensible alternatives explored here.

Chapter 5: Game-Theoretic Contagion Blocking in Networks

In this chapter, I introduce the first piece of work pertaining to game-theoretic resource allocation in networked domains with contagion. Specifically, I present algorithms to calculate game-theoretic equilibria in contagion mitigation games where one player is attempting to mitigate the spread of a second player's influence. Prior work in this line of research include not only game-theoretic resource allocation but also influence maximization, where the challenge is to select the optimal set of nodes from which to spread one's influence [Kempe et al., 2003; Chen et al., 2010b; Kimura et al., 2010; Leskovec et al., 2007]. Competitive variants have examined the best-response problem to an adversary's strategy or action, which is always shown to be NP-Hard, but have not examined a mutual best-response solution [Bharathi et al., 2007; Kostka et al., 2008; Borodin et al., 2010; Budak et al., 2011; He et al., 2012]. Therefore, the major contributions of this work are in opening up a new area of research that combines recent research in security games and influence blocking maximization and showing that such an approach is feasible despite the immense challenges in these problems.

The particular application I use as an example in this work is that of counterinsurgency, which is the contest for the support of the local leaders in an armed conflict and can include a variety of operations such as providing security and giving medical supplies [U.S. Dept. of the Army and

U.S. Marine Corps, 2007]. Just as in word-of-mouth advertising and peacekeeping operations, these efforts carry a social effect beyond the action taken that can cause advantageous ripples through the neighboring population [Hung, 2010]. I model the interaction as a graph with one player attempting to spread influence while the other player attempts to stop the probabilistic propagation of that influence by spreading their own influence. This ‘blocking’ problem models situations faced by governments/peacekeepers combatting the spread of terrorist radicalism and armed conflict with daily/weekly/monthy visits with local leaders to provide support and discuss grievances [Howard, 2011].

After formally defining the problem in Section 5.1, I leverage scalable techniques introduced in Jain et al. 2010 and formulate a double oracle solution and introduce approximate as well as heuristic oracles for both players that allow for far greater scalability than a naïve approach in Section 5.2. Section 5.3 introduces a suite of exact, approximate, and heuristic oracles that can be used to achieve varying levels of guarantees and scalability. Experiments are presented in Section 8.3 and the chapter concludes with an analysis of the strategies developed by my methods in Section 5.5.

5.1 Problem Definition

The counterinsurgency domain I focus on includes one party that attempts to subvert the population to their cause and another party that attempts to thwart the first party’s efforts [Hung et al., 2011; Howard, 2011; Hung, 2010]. I assume that each side can carry out operations such as provide security or give medical supplies to sway the local leadership’s opinion. Furthermore, local leaders will impact other leaders’ opinions of the two parties. Specifically, one leader will convert other leaders to side with their affiliated party with some predetermined probability, giving each party's actions a 'spreading' effect. Since resources for COIN operations are very limited relative to the size of the task, each party is faced with a resource allocation task. Hung (2010) models the leadership network of a single district in Afghanistan (based on real data) with 73 nodes and notes that recent organizational assignments show that a single battalion operates in 4-7 districts and divides into 3-4 platoons per 1-2 districts. This translates into 5-30 teams responsible for a network with 300-500 nodes.

I model the counterinsurgency domain as a two-player influence blocking maximization problem, which allows me to draw from the extensive influence maximization literature. An *IBM* takes place on an undirected graph $G = (V, E)$. One player, the influencer, will attempt to maximize the number of nodes supporting his cause on the graph while the second player, the mitigator, will attempt to minimize the influencer's influence. Vertices represent local leaders that each player can attempt to sway to their cause, while edges represent the influence of one local leader on another. Note that these leaders do not report to one another and hence an undirected edge provides an apt representation of their influence relationship. Specifically, each edge, $e = (u, v)$, has an associated probability, p_e , which dictates the chance that leader u will influence leader v to side with u's chosen player. Since the graph is undirected, this is also the probability that u influences v to side with u's chosen player. Only uninfluenced nodes can be influenced.

In an IBM, the two players each choose a subset of nodes as their pure strategies ($\omega_I, \omega_M \subseteq V$), which I will also refer to as actions. Each action is composed of nodes (also referred to as 'sources') where the allowable number of nodes is referred to as the number of 'resources' a player has and is given for each player ($|\omega_I| = r_I$, $|\omega_M| = r_M$). Figure 5.1 shows an example of a pure strategy for one player as the selection of the two nodes, D and F, filled in. The other

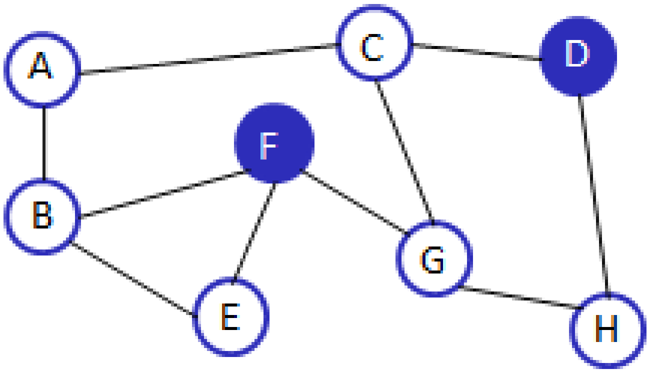


Figure 5.1: Example pure strategy for one player

player would similarly choose a set of nodes on the same graph from which to begin spreading his influence.

Each node in $\omega_I \cap \omega_M$ has a 50% chance of being influenced by each player, while all other nodes in ω_I support the influencer and all other nodes in ω_M support the mitigator. The influence then propagates via a synchronized independent cascade, where at time step t_0 only the initial nodes have been influenced and at t_1 each edge incident to nodes in $\omega_I \cup \omega_M$ is ‘activated’ with probability p_e. Uninfluenced nodes incident to activated edges become supporters of the influencing node’s player. If a single uninfluenced node is incident to activated edges from both player’s nodes, the node has a 50% chance of being influenced by each player. This process is detailed in Algorithm 2.

For a given pair of pure strategies, the influencer’s payoff is equal to the *expected* number of nodes influenced to the influencer’s side and the mitigator’s payoff is the opposite of the influencer’s payoff. Denote the function to calculate the expected number of influencer-influenced nodes as $\sigma(\omega_I, \omega_M)$. Each player chooses a mixed strategy, ρ_I for the influencer and ρ_M for the mitigator, over their pure strategies (subsets of nodes of size r_I or r_M) to maximize their expected payoff. At equilibrium, each player's mixed strategy will be a best-response to the other player's. The mitigator's mixed strategy is a policy by which COIN teams can randomize their deployment each day/week/month, depending on the frequency of missions. The focus of the rest of this work will be to develop optimal, approximate, and heuristic oracles that can be used in double oracle algorithms to generate strategies for these influence blocking maximization problems.

Algorithm 2: INFLUENCE PROP.: ω_I, ω_M, $G = (N, E)$

```
$E^* = \emptyset$, $E^{active} = \emptyset$
$I \leftarrow \{s | s \in \omega_I \wedge s \notin \omega_M\}$, $M \leftarrow \{s | s \notin \omega_I \wedge s \in \omega_M\}$
for $\{s | s \in \omega_I \cap \omega_M\}$ do
    // randomly add $s$ to one of the player's sets
    RandomAdd($s, I, M$)
$N^{new} = I \cup M$
while $N^{new} \neq \emptyset$ do
    for $\{(u, v) | u \in N^{new}$,$(u, v) \notin E^*\}$ do
        // activate the edge based on its probability
        $E^{active}$.add(RandomActivate(($u, v$)))
        $E^*$.add(($u, v$))
    $N^{new} = \emptyset$
    for $\{s | s \notin I \cup M, \exists(u, s) \in E^{active}\}$ do
        $N^{new}$.add($s$)
        // Add $s$ to appropriate set
        AddToSet($s, I, M$)
```

5.2 Double Oracle Approach

The most commonly used approach for a zero-sum game is a naïve Maximin strategy, shown in Algorithm 4. In Algorithm 4, U_R is the mitigator's expected payoff, Ω_I is the set of all influencer actions iterated with ω_I, Ω_M is the set of all mitigator actions iterated with ω_M, and $u(\omega_M, \omega_I)$ is the utility for the mitigator when actions ω_I and ω_M are played. In this problem, the mitigator has a utility equivalent to the opposite of the influencer's which is equivalent to the expectation of the propagation process, $\sigma(\cdot)$. That is, $u(\omega_M, \omega_I) = -\sigma(\omega_M, \omega_I)$. The primary constraint is Constraint 1, which restricts U_R to be no greater than the expected utility achieved by the mitigator in the worst outcome. ρ_M is the mitigator's mixed strategy, where $\rho_M(\omega_M)$ specifies the probability associated with action ω_M in mixed strategy ρ_M. This linear program, however, requires precalculating the payoffs for every pair of player actions to instantiate all constraints before it can efficiently solve for a Nash equilibrium. This naïve approach admits two faults.

Algorithm 3: MAXIMIN LINEAR PROGRAM

```
1
        MAXIMIZE      $U_R$
        SUBJECT TO:
        $\forall \omega_I \in \Omega_I$    $U_R \leq \sum_{\omega_M \in \Omega_M} \rho_M(\omega_M) \cdot u(\omega_M, \omega_I)$    (5.1)
                      $0 \leq \rho_M(\omega_M) \leq 1, \forall \omega_M \in \Omega_M$    (5.2)
                      $\sum_{\omega_M \in \Omega_M} \rho_M(\omega_M) = 1$    (5.3)
```

First, the payoff for a given pair of pure strategies in our problem is computationally intractable to calculate accurately. As shown by Chen et al. (2010b), calculating the analogous expectation in a basic influence maximization game exactly is $\#P$-Hard. Since influence maximization is a special case of influence blocking maximization, it is trivial to show that calculating $\sigma(\cdot)$ exactly is also $\#P$-Hard. The standard method for estimating these expectations is a Monte Carlo approach that was adapted for the *IBM* problem by Budak et al. (2011) and which I also adopt here. It involves simulating the propagation process thousands of times to reach an accurate estimate of the expected outcome. Although it runs in time polynomial in the size of the graph and is able to achieve arbitrarily accurate estimations, the thousands of simulation trials required for accurate results causes this method to be extremely slow in practice.

Second, the Maximin algorithm stores the entire payoff matrix in memory which can be prohibitive for large graphs. For example, with 1000 nodes and 50 resources per player, each player has $\binom{1000}{50}$ actions. To overcome similar memory problems, double oracle algorithms have been proposed in the past [Jain et al., 2011a; Halvorson et al., 2009] and form the basis for our work. More details on the double oracle algorithm can be found in Chapter 2.

5.2.1 Double Oracle: Approximation

Now I prove an approximate double oracle setup that admits a quality guarantee. Denote the mitigator and influencer's mixed strategies at convergence as ρ_M and ρ_I. Also, denote the mitigator's expected utility given a pair of mixed strategies as $u_M(\rho_M, \rho_I)$. Assume that the *mitigator*'s oracle, D_{AR}, is an α-approximation of the optimal best-response oracle, D_{BR}, so that $D_{AR}(\rho_I) \geq \alpha \cdot D_{BR}(\rho_I)$. The following theorem is a generalization of a similar result in Halvorson et al. (2009).

Theorem 4. *Let (ρ_M, ρ_I) be the output of the double oracle algorithm using an approximate mitigator oracle and let (ρ_M^*, ρ_I^*) be the optimal mixed strategies. Then: $u_M(\rho_M, \rho_I) \geq \alpha \cdot u_M(\rho_M^*, \rho_I^*)$.*

Proof. Since we know D_{AR} is an α-approximation, $u_M(\rho_M, \rho_I) \geq u_M(D_{AR}(\rho_I), \rho_I) \geq \alpha \cdot u_M(D_{BR}(\rho_I), \rho_I)$. Since (ρ_M^*, ρ_I^*) is a maximin solution, we know that $\forall \rho_M', \rho_I' : u_M(\rho_M^*, \rho_I') \geq u_M(\rho_M^*, \rho_I^*) \geq u_M(\rho_M', \rho_I^*)$. Thus: $u_M(D_{BR}(\rho_I), \rho_I) \geq u_M(\rho_M^*, \rho_I) \geq u_M(\rho_M^*, \rho_I^*)$, implying $u_M(\rho_M, \rho_I) \geq \alpha \cdot u_M(\rho_M^*, \rho_I^*)$. $\square$

5.3 Oracles

A major advantage of double oracle algorithms is the ability to divide the problem into best-response components. This allows for easily creating variations of algorithms to meet runtime and quality needs by combining different oracles together. Here, I present four oracles that we can combine to create a suite of algorithms.

5.3.1 EXACT Oracle

Solving for a best-response in an influence blocking maximization problem was shown to be NP-Hard by Budak et al. (2011), but an optimal oracle may be useful when paired with an efficient second oracle, given the approximation result just shown. The first oracle I introduce is an optimal best-response oracle. This oracle, which we call EXACT , determines the best-response by iterating through the entire action set for a given player. For each action, the expected payoff against the opponent's strategy is calculated, which requires n calculations of $\sigma(\cdot)$, where n is the size of the support for the opponent's mixed strategy. In this oracle, $\sigma(\cdot)$ is evaluated via the Monte Carlo estimation method, the benchmark technique in influence maximization. This technique involves simulating the propagation process n times, where n is generally 10,000-20,000, and using the average propagation of the simulated trials as the estimate. The ϵ-error of the Monte Carlo estimation exists in the Maximin approach as well, but can be made arbitrarily small with sufficient simulations[Kempe et al., 2003].

This oracle can be used for both the mitigator and the influencer to create an incremental, optimal algorithm that can potentially be superior to Maximin because of the incremental approach. However, the oracle will perform redundant calculations that can cause it to run slower than Maximin when the equilibrium strategies support size is very large.

5.3.2 APPROX Oracle

Here I describe approximate oracles that draw from research in influence maximization, competitive influence maximization, and influence blocking maximization. Budak et al. (2011) showed that the best-response problem for the mitigator is submodular when both players share the same probability of influencing across a given edge. Thus, a greedy hill-climbing approach that provides the highest marginal gain in each round provides a $(1 - \frac{1}{e})$-approximation. This is outlined in Algorithm 4, where MCEst($\cdot$) is the Monte Carlo estimation of $\sigma(\cdot)$, ρ_I is the current influencer mixed strategy, and Action() retrieves a pure strategy, ω_I, and Prob() retrieves a pure strategy's associated probability. The Lazy-Forward speed-up to the greedy algorithm introduced by Leskovec et al. (2007) to tackle influence maximization problems is also implemented, but we do not show it in Algorithm 4 for clarity.

For the influencer's best-response problem, we note that given a fixed mitigator strategy, the best-response problem of the influencer in an *IBM* is exactly the best-response problem of the last player in a competitive influence maximization from Bharathi et al. (2007), which they showed to be submodular. Thus, the influencer's best-response problem can also be approximated with a greedy algorithm with the same guarantees. These oracles are referred to as APPROX .

By combining an APPROX oracle for the mitigator and an EXACT oracle for the influencer, we can create an algorithm that generates a strategy for the mitigator more efficiently than the naive one and guarantees a reward within $(1 - \frac{1}{e})$ of the optimal strategy's reward by Theorem 4. An algorithm with two APPROX oracles no longer admits quality guarantees, but the iterative process still maintains the best-response reasoning crucial to adversarial domains.

Algorithm 4: APPROX -DefBR(ρ_a)

```
$\omega_M = \emptyset$
while $|\omega_M| < r_d$ do
    for $n \in (N - \omega_M)$ do
        U($n$) = $\sum_{\omega_I \in \Omega_I} \rho_I(\omega_I) \cdot$ MCEst($\omega_I, \omega_M \cup \{n\}$)
    $n^* = \arg\max_{n \in N}$ U($n$)
    $BR = BR \cup \{n^*\}$
```

5.3.3 LSMI Oracle

I introduce my main heuristic oracle, LSMI, which is also the name of the heuristic it is based on: Local Shortest-paths for Multiple Influencers (LSMI($\cdot$)). This oracle uses APPROX oracle's Algorithm 4. However, LSMI($\cdot$) is used to replace the MCEst($\cdot$) function and provides a fast, heuristic estimation of the marginal gain from adding a node to the best response. The heuristic is based on two assumptions: very low probability paths between two nodes are unlikely to have an impact and the highest probability path between two nodes estimates the relative strength of the influence. The probability associated with a path is defined as $p = \prod_e p_e$ over all edges e on the path. We then combine these heuristic influences from two players in a novel, efficient way.

The two heuristic assumptions have been applied successfully for one-player influence maximization in various forms, one of the most recent being Chen et al. (2010b). As an application of the first assumption, when calculating the influence of a node, they only consider nodes reachable via a path with an associated probability of at least some θ. As an application of the second assumption, Chen et al. (2010b) assume that each source will only affect nodes via the highest probability path. To improve the accuracy of this estimation, they disallow other sources from being on the path since the closer source's influence will supersede the further source's along the same path. I use these ideas as well, but Chen et al. (2010b)'s approach to the critical step of combining these influences efficiently relies on there being only one type of influence. In a two-player situation such as mine, there are two probabilities associated with each node, and the winning influencer depends not only on the probability but on the distance to sources as well. This ordering effect is a new issue that necessitates a novel approach to influence estimation.

L-Eval($\cdot$), described in Algorithm 6, is my new algorithm for determining the expected influence of the local neighborhood around a given node. LSMI (n, ω_I, ω_M) estimates the marginal gain of n by finding the difference between calling L-Eval($\cdot$) with and without n and replaces the MCEst($\cdot$) function in Algorithm 4. For the mitigator oracle, instead of a call of MCEst($\omega_I, \omega_M \cup n$):

$$\text{LSMI}(\omega_I, \omega_M, n) = \text{L-Eval}(V, \omega_I, \omega_M \cup \{n\}) \text{ - L-Eval}(V, \omega_I, \omega_M),$$
$$\text{s.t. } V = \text{GetVerticesWithin}\theta(n).$$

GetVerticesWithinθ() is a modified Dijkstra's algorithm that measures path-length by hop-distance, tie-breaks with the associated probabilities of the paths, and stores all nodes' shortest hop-distance and associated probability to the given node. It does not add a new node to the search queue if the probability on the path to the node falls below θ. This procedure is outlined in more detail in Algorithm 5. The overall structure remains identical to Dijkstra's algorithm, but distances are now measured with hop-distance instead of summing the weights on edges and a cut-off is implemented when the probability on the path falls below θ.

Algorithm 5: GetVerticesWithin$\theta(n)$

```
for $v \in V$ do
    hopDistanceTo[$v$] := infinity
    probDistanceTo[$v$] := 0
    $Q$.enqueue($v$)
hopDistanceTo[$n$] := 0
probDistanceTo[$n$] := 1

while $Q \neq \emptyset$ do
    $u$ := vertex in $Q$ with smallest distance (by hopDistanceTo)
    remove $u$ from $Q$
    if hopDistanceTo[$u$] == infinity then
        break
    for each neighbor $v$ of $u$ do
        $t_{hop}$ := hopDistanceTo[$u$] + 1
        $t_{prob}$ := probDistanceTo[$u$] $\cdot\ p_{(u,v)}$
        if $t_{hop} \leq$ hopDistanceTo[$v$] AND $t_{prob} > \theta$ then
            hopDistanceTo[$v$] := $t$
            if $t_{prob} >$ probDistanceTo[$v$] then
                probDistanceTo[$v$] := $t_{prob}$
        /* Reorder $v$ in Queue, tie-break with probDistanceTo[]*/
        decrease-key($v$,$Q$)
return all $v$ with hopDistanceTo[] less than infinity
```

In L-Eval($\cdot$), V is the set of n's local nodes and ω_I/ω_M are the influencer/mitigator source sets. Due to the addition of n, we must recalculate the expected influence of each $v \in V$. First, determine all the nearby nodes that impact a given v by calling GetVerticesWithin$\theta(v)$. Since only sources exert influence, we intersect this set with the set of all sources and compile them into a priority queue ordered from lowest hop-distance to greatest. p_I and p_M represent the probability that the influencer/mitigator successfully influences the given node. From the nearest source, we aggregate the conditional probabilities in order. If the next nearest source is an influencer source, then p_I is increased by the probability that the new source succeeds, conditional on the failure of all closer mitigator and influencer sources. The probability that all closer sources failed is exactly $p_I + p_M$. p_M remains unchanged. If the next nearest source is a mitigator source, then

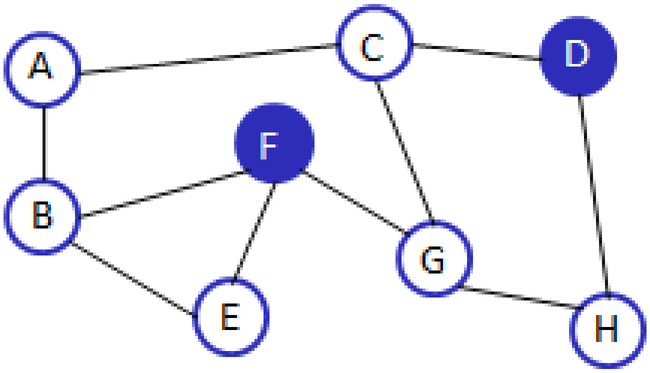


Figure 5.2: Example network

a similar update is performed. The algorithm iterates through all impacted nodes and returns the total expected influence.

To illustrate the aggregation calculation, I reproduce the graph from Figure 5.1 again here in Figure 5.2. Consider node h and assume the influencer has chosen source f and mitigator has chosen source d. Since influence travels along edges in an ordered fashion, the influence of f is only possible if d fails to influence h, since d is closer in terms of hop-distance. Thus, the probability that h is converted into an influencer node is:

$$(1 - p_{(d,h)}) \cdot (p_{(f,g)} \cdot p_{(g,h)}) \tag{5.4}$$

In words, the probability is equal to the joint probability that d fails and the influence from f succeeds in influencing g and then h thereafter. Notice that if the mitigator had a second source further away, it would be completely irrelevant, since only influencer-influenced nodes contribute to the payoff determination.

Although the estimated marginal gain of LSMI can be arbitrarily inaccurate, choosing the best action only requires that the *relative* marginal gain of different nodes be accurate. I show in the experiments in Section 8.3 that LSMI does a very good job of this in practice as evidenced by the high reward achieved by LSMI-based algorithms.

Algorithm 6: L-Eval(V, ω_I, ω_M)

```
InfValue = 0
for $v \in (V - \omega_I - \omega_M)$ do
    N $=$ GetVerticesWithin$\theta(v) \cap (\omega_I \cup \omega_M)$
    /* Prioritize sources by *lowest* hop-distance to $v$*/
    S $=$makePriorityQueue(N)
    $p_I = 0, p_M = 0$
    while S $\neq \emptyset$ do
        $s =$ S.poll()
        if $(s \in \omega_I)$ then
            $p_I = p_I + (1 - p_I - p_M)\cdot$ Prob$(s, v)$, $p_M = p_M$
        else /* $s$ must be in $\omega_M$ */
            $p_M = p_M + (1 - p_I - p_M)\cdot$ Prob$(s, v)$, $p_I = p_I$
        end if
    $InfValue = InfValue + p_I$
return $InfValue$
```

5.3.4 PAGERANK Oracle

PageRank is a popular algorithm to rank webpages [Brin and Page, 1998], which I adapt here due to its frequent use in influence maximization as a benchmark heuristic. The underlying idea is to give each node a rating that captures the power it has for spreading influence that is based on its connectivity. For the purposes of describing PageRank, we will refer to directed edges $e_{u,v}$ and $e_{v,u}$ for every undirected edge between u and v. For each edge $e_{u,v}$, set a weight $w_{u,v} = p_e/p_v$ where $p_v = \sum_e p_e$ over all edges incident to v. The rating or 'rank' of a node u, $\tau_u = \sum_v w_{u,v} \cdot \tau_v$ for all non-source nodes v adjacent to u. The exclusion of source nodes is performed because u cannot spread its influence through a source node.

For my oracles, since the mitigator's goal is to minimize the influencer's influence, the mitigator oracle will focus on nodes incident to influencer sources $N_I = \{n | n \in V \wedge \exists e_{n,m}, m \in \omega_I\}$. Specifically, ordering the nodes of N_I by decreasing rank value, the top r_M nodes will be chosen as the best response. In the influencer's oracle phase, the influencer will simply choose the nodes with the highest ranks. Although PAGERANK is very efficient, we should expect its quality to be low, since the influencer oracle fails to account for the presence of a mitigator and the mitigator oracle only searches through nodes directly incident to the influencer's source nodes. I will refer to oracles based on this heuristic as PAGERANK .

5.4 Experiments

In this section, I show experiments on both synthetic and real-world leadership and social networks. I evaluate the algorithms on scalability and solution quality. One advantage of double oracle algorithms is the ease with which the oracles can be changed to produce new variations of existing algorithms. This allows me to simulate various influencer/mitigator best-response strategies and test my heuristics' performance more thoroughly.

Ideally, I would report the performance of my mixed strategy against an optimal best-response as a worst-case analysis. However, due to scalability issues with the EXACT best-response oracle, rewards for larger graphs can only be calculated against an approximate best-response generated by the APPROX oracle. Unless otherwise stated, each datapoint is an average over 100 trials and the games created used contagion probability on edges of 0.3, 20,000 Monte Carlo simulations per estimation, and an LSMI $\theta = 0.001$. All experiments were run on machines with CPLEX 12.2, 2.8 GHz CPU, and 4GB of RAM.

Algo Label	Def. Oracle	Att. Oracle	Nodes	R
DOEE	EXACT	EXACT	15	3
DOAE	APPROX	EXACT	20	3
DOAA	APPROX	APPROX	100	3
DOLE	LSMI	EXACT	20	3
DOLA	LSMI	APPROX	100-200	3
DOLL	LSMI	LSMI	450	20
DOLP	LSMI	PAGERANK	700	20
DOPE	PAGERANK	EXACT	40	3
DOPA	PAGERANK	APPROX	200-300	3
DOPL	PAGERANK	LSMI	1000+	20
DOPP	PAGERANK	PAGERANK	1000+	20

Table 5.1: Algorithms evaluated

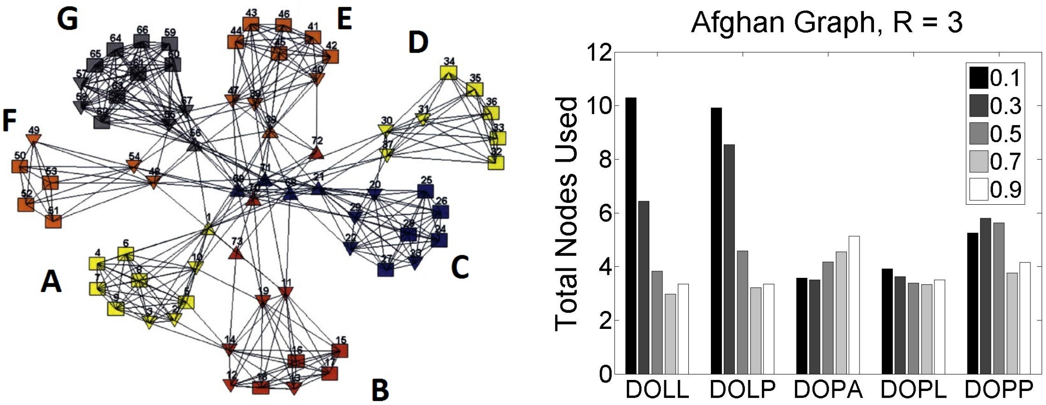
In addition to the optimal Maximin algorithm, I also test the set of double oracle algorithms listed in Table 5.1, where Nodes and R(esources) indicate the approximate problem complexity the algorithm can handle within 20 minutes based on experiments with scale-free graphs.

5.4.1 Leadership Networks

In Hung (2010), a leadership network was created based on real data of a district in Afghanistan with 7 village areas, each with a few ‘village leaders’ with connections outside the village, and a cluster of ‘district leaders’ shown in the middle. I recreate the same network, shown in Figure 5.3a and run my algorithms on it. Although not shown, quality as measured against an APPROX influencer was very similar for all algorithms. Algorithms exceeding 20 minutes are not shown.

Closer examination of mitigator strategies reveals a difference in the oracles’ approach. Since the PAGERANK mitigator oracle considers only influencer-adjacent nodes with the highest rank, most of its strategies focus on two high-degree district leaders (neither are maximal degree nodes) and on a regular member of the highest population Village G. In this graph structure, where sets of nodes are fully connected, this strategy works very well because the influencer’s best response will often be the highest degree district leader and a node in Village G. This approach is more conservative than LSMI , which directly chooses the influencer's source nodes since the 50% chance of wiping out an influencer source provides slightly higher utility. The influencer oracles all select from the same set of four high-degree nodes. Aside from the highest-degree district leader and Village G nodes, an additional high-degree village leader far from Village G is also used. This result suggests that not only connectivity, but also strategic spacing provided by my algorithms is a key point for the influencer's target selection.

Experiments varying contagion probability, shown in Figure 5.3b, show LSMI mitigator oracle algorithms randomizing over many more nodes at low contagion levels. This occurs because the influencer's initial set of nodes accounts for most of his expected utility, encouraging randomization over many nodes. PAGERANK ignores this since a given set of nodes is often adjacent to all sets of influencer-chosen nodes, while LSMI responds by matching the increase node use directly.

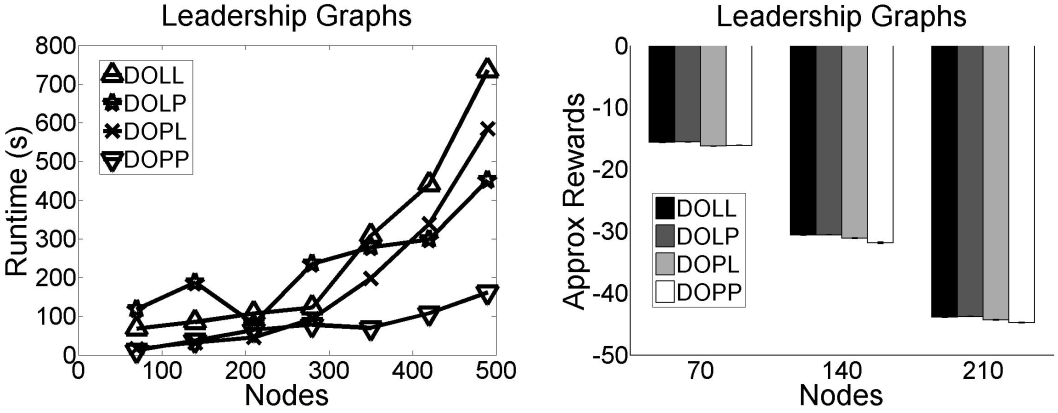


(a) Network from Hung (2010)

(b) Nodes in mitigator strategy

Figure 5.3: Afghanistan leadership network results

As noted previously, a battalion is responsible for 4-7 districts, so I create synthetic graphs with multiple copies of a village structure (70 nodes each) and link all district leaders together to create multi-district graphs. In my experiments, for every district, each player is given 3 resources. Figure 5.4 shows runtime and solution quality against an APPROX influencer best-response. Since I create the graphs one district at a time, the graph sizes increase by 70 nodes at a time. The trend in rewards is once again that LSMI mitigator oracle algorithms very slightly outperform the others. All four algorithms scale to real-world problem sizes.



(a) Runtime

(b) Quality

Figure 5.4: Synthetic leadership network results

5.4.2 Random Scale-Free Graphs

[1]

Scale-free graphs have commonly been used as proxies for real-world social networks because the distribution of node degrees in many real world networks have been observed to follow a power law [Clauset et al., 2009]. I conduct experiments on randomly generated scale-free

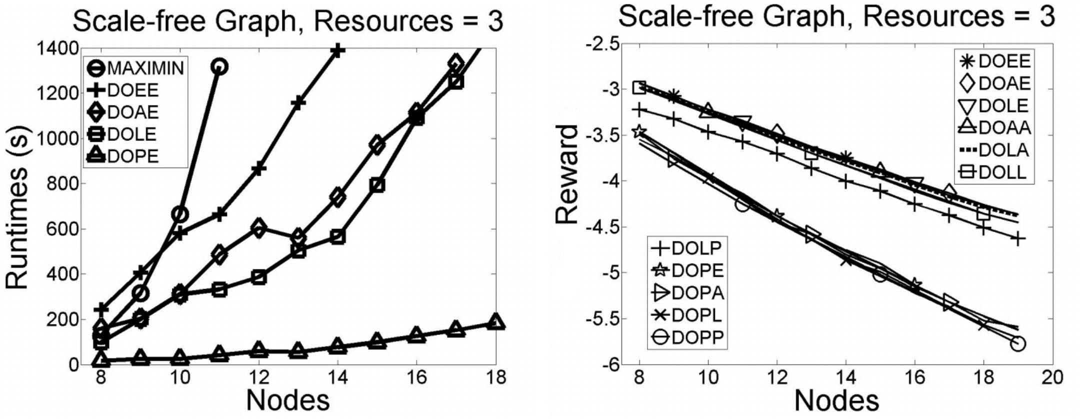
[1] The networks are built using the Barabási-Albert network building algorithm and will be referred to as scale-free networks.

graphs of various sizes to illustrate both the runtime scalability and quality of each algorithm in graphs resembling social networks as opposed to leadership networks.

Figure 5.5 shows the results for small scale-free graphs of 8-20 nodes with 3 resources for each player. The runtime graph, Figure 5.5a shows only the algorithms that exceed 20 minutes for clarity. The remaining heuristic algorithms' results all hug the x-axis because they take minimal time for these graphs. As would be expected, Maximin scales the most poorly and is only able to handle graphs of up to 11-12 nodes. The approximate algorithm, DOAE improves upon DOEE and can handle up to 16-17 nodes, but swapping out the APPROX oracle for the very fast LSMI oracle does not improve runtime scalability very noticeably. This is because although the LSMI oracle is orders of magnitude faster than the APPROX oracle, the EXACT influencer oracle's runtime eclipses both of them, making the improvement irrelevant.

In Figure 5.5b, I show the reward obtained by the mitigator when using the strategies generated against an EXACT influencer best-response as described earlier. The key point is that the majority of rewards are indistinguishable from the optimal algorithms. The DOLL algorithm begins to diverge slightly when the graph nears 100 nodes, but the major exceptions are the algorithms featuring PAGERANK mitigator oracles. Interestingly, DOLP, which uses LSMI for the mitigator and PAGERANK for the influencer still generates high rewards.

Figure 5.6 shows runtime and quality for larger scale-free graphs of 20-100 nodes with 3 resources for each player. As can be seen, the algorithms featuring the APPROX oracle (DOAA, DOLA) begin to exceed the 20-minute cutoff near 100 nodes while the remaining heuristic algorithms continue to hug the x-axis because even these games are completed in minimal time. As discussed previously, due to the inefficiency of the EXACT oracle, I use an APPROX best-response



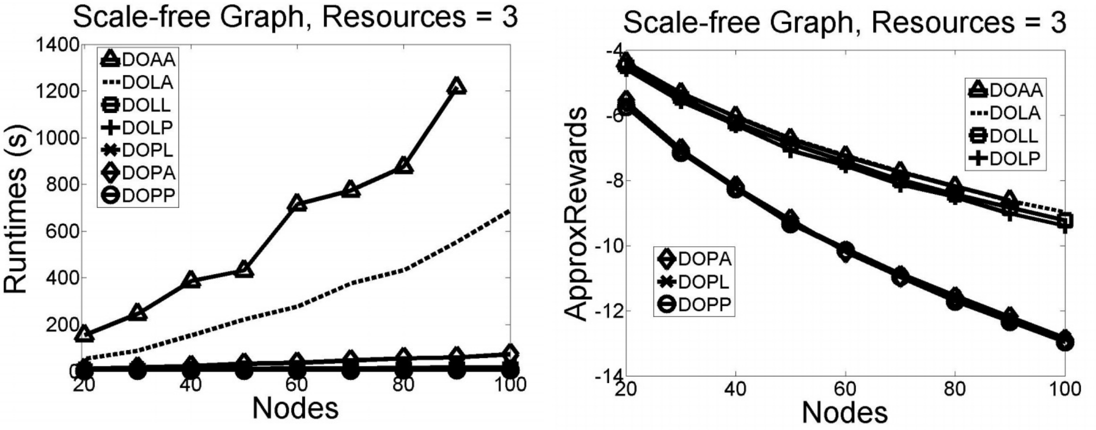
(a) Runtime

(b) Quality

Figure 5.5: Scale-free, 8-20 nodes, 3 resources

to calculate a more conservative reward value. Figure 5.6b again shows algorithms with PageRank mitigator oracles performing noticeably more poorly than the other algorithms. DOLP is again very close to the top performers. Note that while this may be due to the Approx best-response being used instead of an Exact best-response, it is very unlikely than an influencer could perform any better given the hardness of the best-response problem.

Finally, I show very large graph scalability with 100-500 nodes and 20 resources per player in Figure 5.7. These games can only be handled by algorithms using two heuristic oracles, so I try all combinations of LSMI and PageRank oracles. When two LSMI oracles are used, the algorithm begins to exceed 20 minutes around 450-475 nodes. However, when even one of the oracles is replaced with a PageRank oracle, the algorithm scales much better. As noted earlier, DOLP performs very close to DOLL's quality and here we see that it scales much better, suggesting that this combination of oracles provides the best blend of runtime scalability and quality.



(a) Runtime

(b) Quality

Figure 5.6: Scale-free, 20-100 nodes, 3 resources

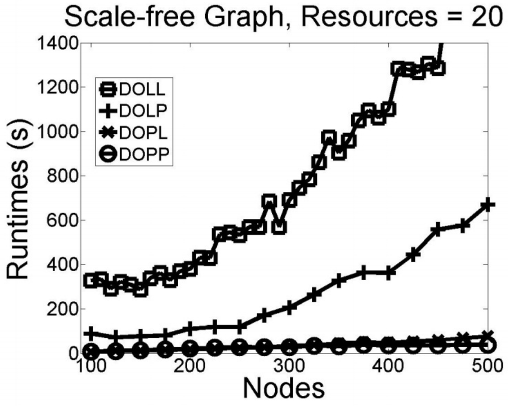
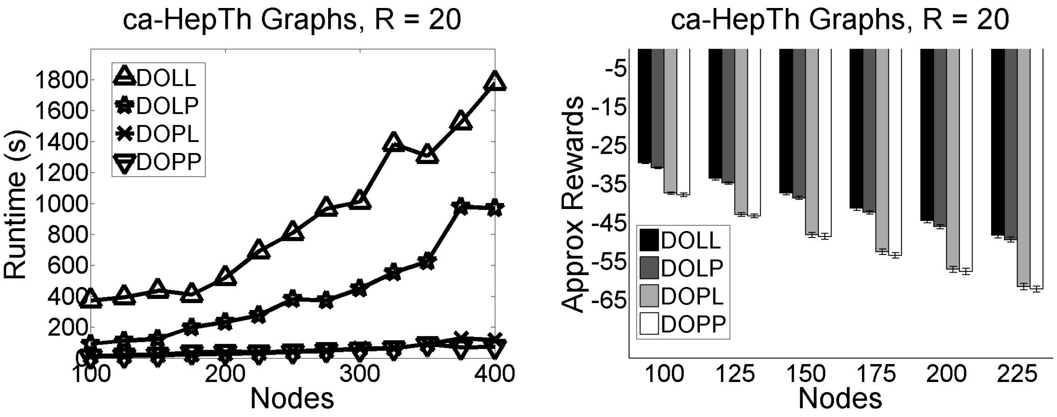


Figure 5.7: Scale-free, 100-500 nodes, 20 resources

5.4.3 Social Networks

To evaluate the performance of my algorithms on social networks, I use the real-world network commonly used to evaluate influence maximization algorithms: High Energy Physics Theory collaboration network (ca-HepTh). I use this graph as an approximation for a general social network as opposed to the leadership network in the previous section which is hierarchical in structure. For the experiments conducted herein, I extract randomly generated subgraphs of varying sizes each of which is generated so that the degree of included nodes are proportional to their degree in the actual dataset.



(a) Runtime

(b) Reward

Figure 5.8: ca-HepTh results

The results shown in Figure 5.8 are very similar to the results from Figure5.6 and 5.7. Unlike in the leadership graphs, the PAGERANK mitigator oracle works poorly in social networks, just as in random scale-free graphs. Simply choosing the highest ranking neighbors may have minimal effect on the influence of an influencer source because many neighbors will *not* be interconnected, which was not the case in leadership networks.

5.5 Strategy Analysis

In addition, three types of variations were explored on scale-free networks in more depth. First, I varied the size of the graph and kept all other parameters constant. Second, I varied the average contagion probability in the graphs at three separate graph sizes. Finally, I varied the standard deviation of the contagion probability in the graphs and again tested these at three separate graph sizes. All experiments featured a randomly generated scale-free graph, 10 resources per player

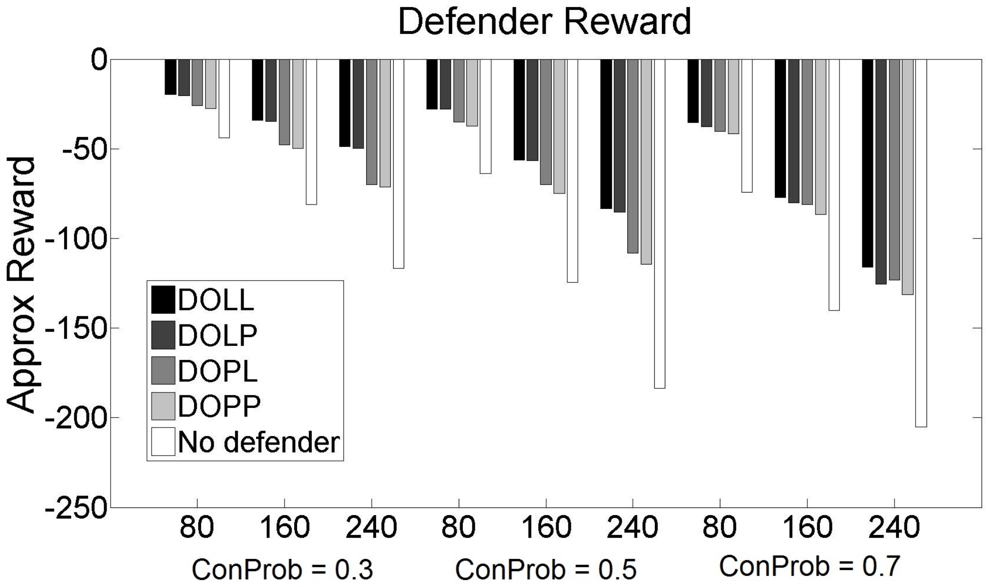
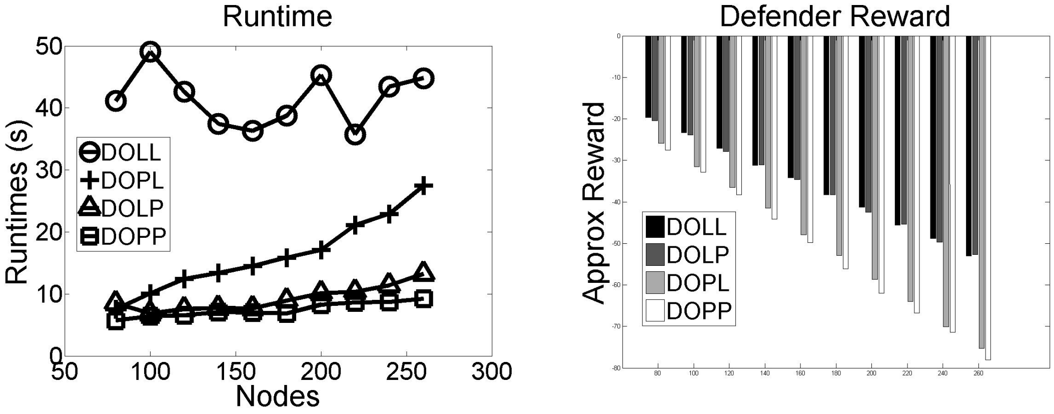


Figure 5.9: Preliminary test, r = 10, avg. = 0.3, s.d. = 0.1

($\omega_M, \omega_I = 10$), and contagion probabilities on edges that were drawn from a normal distribution. Scale-free graphs were chosen due to their widespread use as proxies for general social networks and were generated according to the principle of 'preferential attachment' as introduced by Barabási and Albert 1999. My particular implementation adds edges between existing vertices and newly added vertices with a probability of $p = (\deg(v) + 1) / (|E| + |V|)^2$. 100 trials were run for every data point shown.

Figure 5.9 shows a preliminary test that was conducted to provide a benchmark for the quality results. It shows the reward for the mitigator when each of the four algorithms is used as well as when no mitigator is present as well for graphs of size 80, 160, and 240 and with the average contagion probability set to 0.3, 0.5, and 0.7. Again, the reward reported is the reward achievable by an adversary that best-responds to our algorithm's generated mitigator strategy by calculating the approximate best-response via the algorithm proposed by Budak et al. (2011). As mentioned,

[2] http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/ generators/random/BarabasiAlbertGenerator.html



(a) Runtime

(b) Quality

Figure 5.10: Scale-up results, r = 10, avg. = 0.3, s.d. = 0.1

the graph sizes tested were limited to 260 nodes because for larger graphs even calculating the approximate best-response outlined above begins to take longer than 20 minutes as well.

As can be seen, all of the algorithms provide at least a 30-40% improvement in reward obtained as opposed to having no mitigator present across all of the cases tested. Since this was intended as a preliminary justification for the algorithms, I will provide more in-depth analysis of the solution quality of the algorithms in the following subsections.

5.5.1 Graph size scale-up

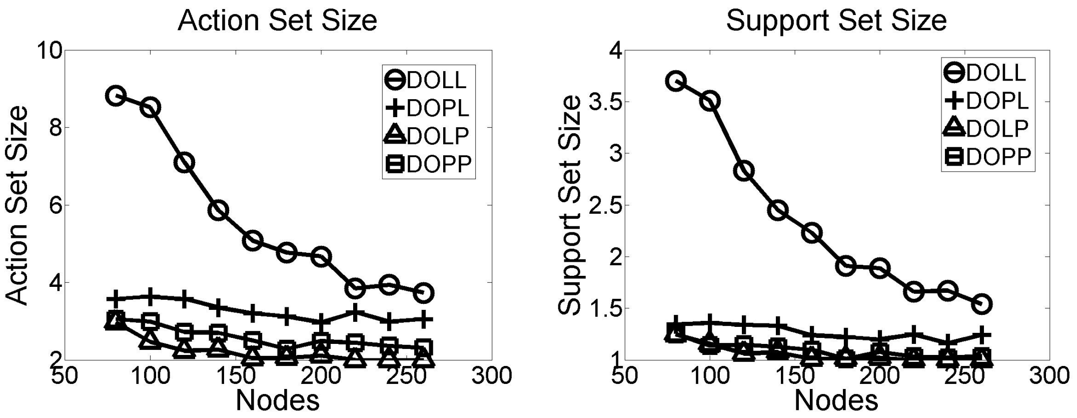
The first set of experiments explored the impact of scaling up the size of the graph alone. Specifically, the more efficient four algorithms (all combinations of the LSMI and PAGERANK oracles) were run on randomly generated scale-free graphs with 80-260 nodes in increments of 20, with 10 resources and contagion probabilities drawn from a normal distribution $\mathcal{N}(0.3, 0.1)$. Graph sizes were limited to 260 nodes because the adversary best-response technique used to determine the mitigator's reward became too cumbersome for larger graphs.

Figure 5.10a shows the impact on runtime as the graph size is scaled up. As can be seen, the solution technique that features two LSMI oracles (DOLL) requires the longest run time at 40-50 seconds for all of the game sizes tested. Interestingly, there did not appear to be a consistent increase in runtime as was observed in the other 3 algorithms (each of which had at least one PAGERANK oracle). This is due to the fact that the runtime depends on the size of the problem but also on the ability of the oracles to find new, higher-quality pure strategies to add to the subgame being solved. DOLL features two highly adaptive LSMI oracles and, as evidenced, tends to generate many more actions for the smaller graph sizes. Thus, although the graphs get larger, fewer iterations are used, causing minimal runtime increase as the graph size is increased.

The other 3 algorithms were much faster across the board, all requiring less than 30 seconds with a consistent trend as the graph size increases. DOPL requires more time than DOLP because of the fact that the mitigator PAGERANK oracle explicitly adapts to the influencer's strategy (only uses nodes adjacent to influencer nodes), while the influencer PAGERANK oracle does not.

Figure 5.10b shows the impact on solution quality as the graph size is scaled up. Unsurprisingly, as the size of the graph increases, it becomes increasingly difficult for the mitigator to block the adversary's influence spread and the mitigator receives a correspondingly lower reward. Again, we also observe a large difference between algorithms that use a LSMI oracle for the mitigator as opposed to a PAGERANK oracle for the mitigator, with the latter providing much lower rewards. This is expected, due to the higher sophistication of the LSMI mitigator oracle as was noted earlier.

Figure 5.11a shows the final number of actions in the mitigator's action set as the size of the graph is increased. The action set is defined as the number of actions available to the mitigator in the CoreLP phase of the double oracle algorithm and is exactly the number of new best-responses



(a) Action set size

(b) Support set size

Figure 5.11: Scale-up results, r = 10, avg. = 0.3, s.d. = 0.1

that have been found by the mitigator oracle. In the worst case, this would include all possible actions in the game, but as can be seen is generally far smaller, making the problem much more tractable. The influencer's action set size was always extremely similar if not identical to the mitigator's action set size.

Figure 5.11b shows a similar metric and features the number of actions in the support set of the final mitigator strategy. The support set is the set of actions that have non-zero probability in the final mixed strategy. Again, the final influencer support set size was always extremely similar if not identical to the mitigator's.

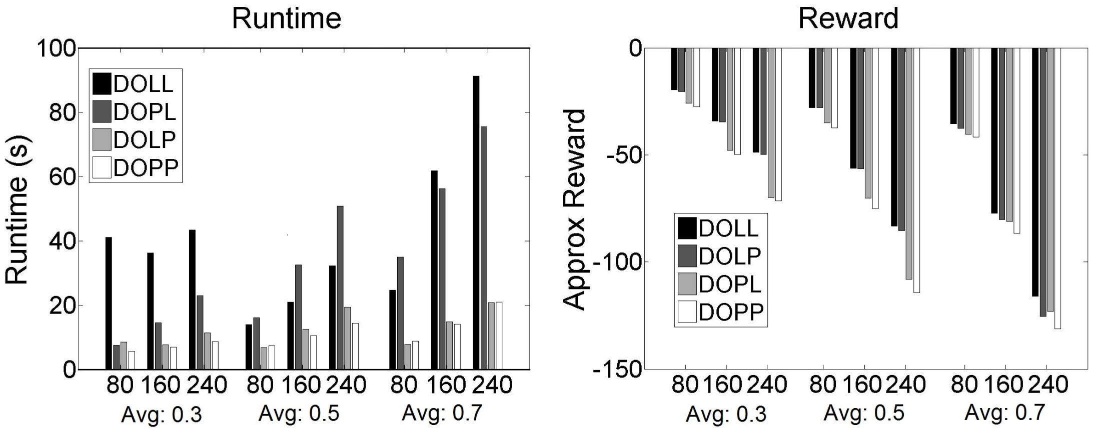
As can be seen, both the action set and the support set sizes are much larger with the DOLL algorithm than for any of the other algorithms. This is due to the sophistication of the LSMI oracles as opposed to the PageRank oracle. The PageRank oracles converge extremely quickly to a small set of actions and often do not generate new actions in response to new adversary strategies. This is especially true for the PageRank influencer oracle, since the mitigator oracle actually chooses nodes directly adjacent to the influencer. Thus, even when only one PageRank oracle is used, the algorithm overall converges quickly. The DOLL algorithm is iterating many more times than algorithms featuring the PAGERANK oracle, leading to the previous runtime result with DOLL being far slower than the other algorithms.

Furthermore, the trends seen in both Figure 5.11a and 5.11b show the size of the final action set and support set decreasing as the graph size is increased. This is due to the fact that as the graph grows larger, very few actions are useful for the mitigator to use to defend against the spread of the influencer's influence. For the influencer, randomization becomes less essential for the same reason. Thus, both players converge to a very small set of actions for the final mixed strategy.

5.5.2 Contagion probability: Average

To explore the impact of changing the contagion probabilities on the four algorithms, I tested three different contagion probability averages for three separate graph sizes. Specifically, I ran all four algorithms with the contagion probabilities drawn from normal distributions $\mathcal{N}(0.3, 0.1)$, $\mathcal{N}(0.5, 0.1)$, and $\mathcal{N}(0.7, 0.1)$. The graph sizes tested were 80, 160, and 240 node random scale-free graphs with 10 resources allowed per player. I measured the same 4 metrics as in the previous section: runtime, solution quality, action set size, and support set size.

Figure 5.12a shows the results pertaining to runtime. The x-axis is divided into three sets of three bars each. Each set represents one setting for the contagion probability average (0.3, 0.5, 0.7) while each bar represents the runtime result for one algorithm. At averages of 0.5 and 0.7, consistent trends can be seen, with larger graphs taking longer and higher probabilities leading to longer runtimes for algorithms with LSMI oracles. This is because LSMI oracles



(a) Runtime

(b) Quality

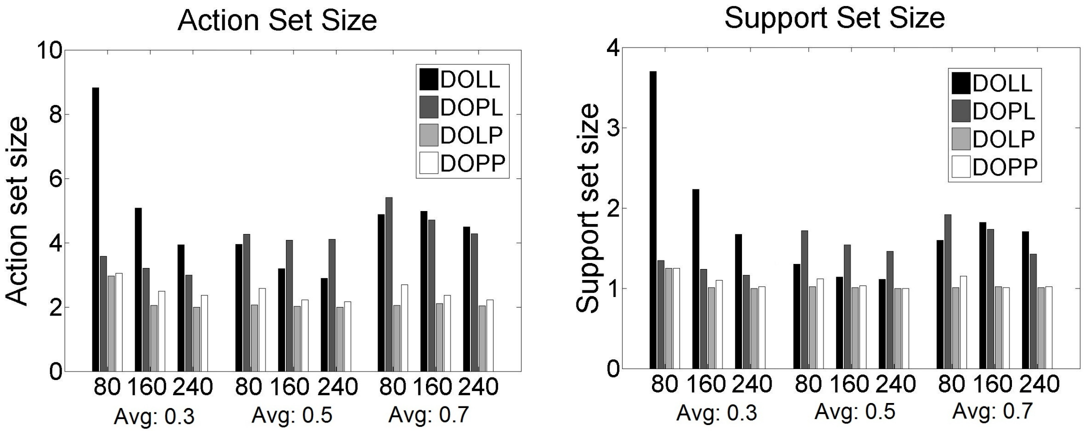
Figure 5.12: Contagion probability average results, s.d. = 0.1

speed up heuristic estimation by calculating only high probability influences, but when contagion probabilities are higher, this leads to many more nodes that must be processed by the algorithm.

For the case of 0.3, however, the trend is not consistent for the DOLL algorithm. Experiments suggest that with low contagion probabilities, two LSMI oracles continually find new best-responses to each other's strategies. This occurs because at low contagion probabilities, different parts of the graph interact minimally and the influencer is able to move to 'new' nodes and entirely avoid the mitigator, resulting in a cat-and-mouse game that requires many more iterations to converge than when a PAGERANK oracle is used.

Figure 5.12b shows the reward for the mitigator using the same approximate best-response technique described previously. Unsurprisingly, larger graphs lead to lower reward for the mitigator because it is harder to defend. Higher contagion probabilities also result in lower mitigator rewards for the same reason.

As we noticed in the scale-up experiments, larger graphs lead to fewer actions in the action set as well as the final support set, as shown in Figures 5.13a and 5.13b. As mentioned, at the



(a) Action set size

(b) Support set size

Figure 5.13: Contagion probability average results, s.d. = 0.1

lowest contagion probability tested (0.3), the action and support set sizes are very large for DOLL, causing very high runtimes due to the many iterations required to generate the observed action sets.

5.5.3 Contagion probability: Standard deviation

Next I tested variations of the standard deviation of the normal distribution that the contagion probabilities on edges are drawn from. Specifically, we ran all four algorithms with the contagion probabilities drawn from normal distributions $\mathcal{N}(0.3, 0.0)$, $\mathcal{N}(0.3, 0.05)$, $\mathcal{N}(0.3, 0.1)I$, and $\mathcal{N}(0.3, 0.15)$. These results, however, did not show statistically significant differences in the results when the standard deviation was changed under the particular parameter settings we tested. We only show the runtime results in Figure 5.14 to support this claim, but the quality, action set size, and support set size results all looked similarly homogenous across the different standard deviations tested.

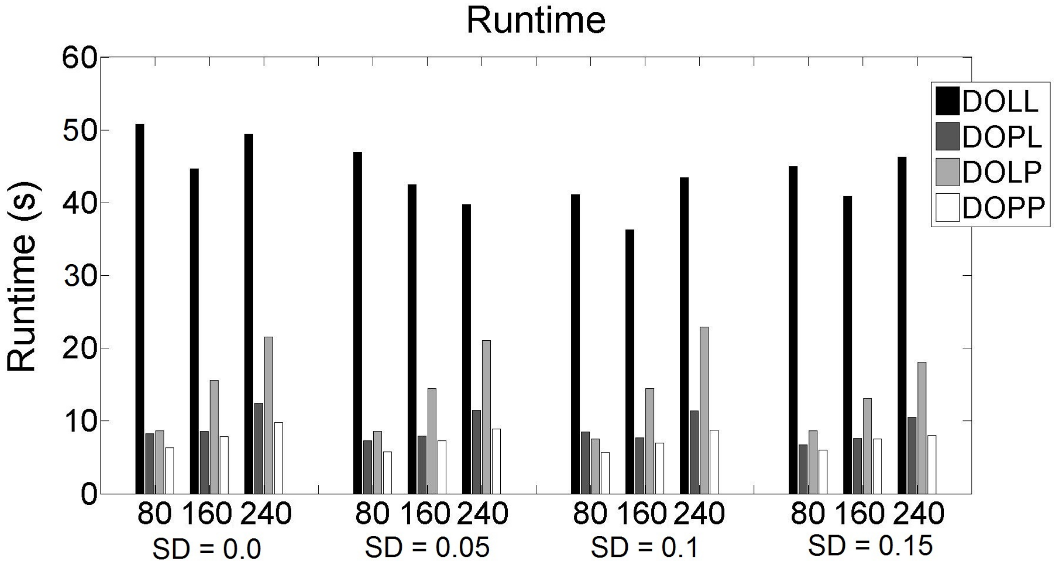


Figure 5.14: Contagion probability s.d. results, avg. = 0.3

Chapter 6: Bayesian Game-Theoretic Contagion Blocking in Networks

This final chapter on game-theoretic resource allocation extends the previous chapter's framework by modeling uncertainty in the mitigator's network knowledge. Specifically, the insurgents in a counterinsurgency operation are generally an indigenous group that have been in the area for many more years than the counterinsurgency operatives that are attempting to subvert the insurgent influence. As such, it is natural to assume that the insurgents have superior knowledge of the social network in the area and that the counterinsurgents are at an informational disadvantage. Thus, I assume that the influencer (an insurgent group) has perfect knowledge of the graph structure, while the mitigator is uncertain about the influence network. In the resulting Bayesian game, a type of the influencer corresponds to a particular instantiation of the influence graph, and the mitigator must reason over the distribution over these graphs (i.e., influencer types) in order to compute an optimal strategy. Given the Bayesian framework, numerous other natural forms of uncertainty can clearly be modeled beyond the existence of edges.

Researchers that have explored the impact of network perturbations on measures of centrality and influence spread in one-player maximizations have noted the extreme impact even slight changes in graph structure can have [Watts and Strogatz, 1998; Kempe et al., 2003; Lahiri et al., 2008; Budak et al., 2011; He et al., 2012]. For example, Watts and Strogatz (1998), in their seminal work on the small-world phenomena noted that adding only a few key edges ('short cuts') was sufficient to cause drastic changes in the infection dynamics. Lahiri et al. (2008), studying the impact of networks that change over time, found that the top influencers in a given time step were drastically and unpredictably altered as the network evolved. In addition, previous work on Bayesian security games has shown that not accounting for even small degrees of payoff uncertainty can lead to large drops in solution quality [Kiekintveld et al., 2011]. Thus, one expects strategies generated without modeling most of the uncertainty about graph structure to do far worse than the optimal solution to the Bayesian game.

After formally specifying the Bayesian game in Section 6.1, I specify an array of models of uncertainty and graph structures that I experiment on in Section 6.2. Supporting the intuition of poor performance when ignoring uncertainty, Section 6.3 shows that there are cases where a mitigator who has incorrect information about a single edge can suffer unbounded loss and that quantifying the impact of changing a single edge in a given graph is #P-Hard. I also empirically show that, indeed, under my models of uncertainty, optimal mitigator strategies for different influencer types are vastly different.

However, while past work has focused on sophisticated algorithms for Bayesian security games [Jain et al., 2011b; Kiekintveld et al., 2011; Yin and Tambe, 2012], I showcase the opposite approach that runs directly counter to what intuition and my initial experiments suggest: ignoring the vast majority of uncertainty has minimal impact. Specifically, I show through extensive experiments that computing a mitigation strategy based on a game with only a few *randomly* sampled influencer types yields near-optimal rewards for widely varied models of uncertainty. I experiment on 3 different synthetic graph models with and without resource imbalances on both sides, 5 models of uncertainty, weighted/unweighted counting of nodes, varied edge weight distributions, varied graph sizes, varied degrees of uncertainty, and varied degrees of sampling. I also conduct experiments on two real-world social networks using two different models graph construction. Results on these over 200 experimental settings shown in Section 8.3 consistently reveal the same result: simple sampling techniques perform near-optimally. Analysis at the end of Section 8.3 suggests that past research does not inform the existence of the phenomenon observed, but that graph structure and the two-player nature of the game play key roles in the robustness of random sampling. This suggests that even in domains as challenging as this, models which ignore uncertainty may nevertheless be robust to it.

6.1 Asymmetric Information Game

I model counterinsurgency as a two-player Bayesian zero-sum game situated on a graph in which two players, the influencer (denoted by I) and the mitigator (denoted by M) compete to maximize influence over the nodes. Formally, let $G = (V, E)$ be a graph with weighted nodes V and edges E, and for each edge $(i, j) \in E$, let p_{ij} be the probability that node i's opinion will influence node j. Pure strategies for the mitigator, ω_M, and the influencer, ω_I, are as in the previous chapter, with each player allowed to choose a subset of nodes. Mixed strategies for each player are denoted ρ_M for the mitigator and ρ_I for the influencer. Influence propagation again occurs via a synchronized independent cascade with the utility of the influencer being $U_I(\omega_M, \omega_I) = \sigma(\omega_M, \omega_I)$, where $\sigma(\omega_M, \omega_I)$ is the expected value of nodes that adopt the influencer's opinion following the independent cascade process.

I now depart from the model in the previous chapter by relaxing the complete/symmetric information assumption. Specifically, I assume that the influencer knows the actual influence graph G exactly, while the mitigator is uncertain about its true structure, and only knows the probability distribution over possible graphs. Let λ be an index identifying a particular graph G_λ, and let us make explicit the dependence of the expected influence on the graph, denoting it by $\sigma(\omega_M, \omega_I, \lambda)$. Finally, denote by P the probability distribution over λ, with P_λ being the probability that the true graph is the one identified by λ. From the mitigator's perspective, the influencer's decision will depend on his type; that is, on the true graph which the influencer observes. Thus, we can view the influencer's strategy ρ_I as a function of λ, with ρ_I^λ denoting the influencer's mixed strategy when his type is λ. The mitigator's utility is then $U_M(\rho_I^\lambda, \rho_M) = -E_{\lambda \sim P}[\sigma(\rho_I^\lambda, \rho_M, \lambda)]$.

6.2 Models of Networks and Uncertainty

Numerous stochastic generative models for graphs have been proposed to generate synthetic instances of graphs that resemble real social networks [Newman, 2010]; some of the best known examples are the preferential attachment process, which generates scale-free graphs [Barabási and Albert, 1999], and the process of generating small-world networks pioneered by Watts and Strogatz (1998). Recently, a new generative model, BTER, has been developed, and the authors convincingly demonstrated that this model matches the important properties of real-world networks, such as the distribution of degrees and clustering coefficients, far better than previously proposed methods [Seshadhri et al., 2012]. BTER graphs feature a scale-free collection of densely clustered community structures (dense Erdös-Rényi subgraphs), which are sparsely interconnected by ‘inter-community’ edges. I conducted experiments on BTER graphs (including variations in community density and interconnectedness), small-world graphs (Watts-Strogatz), preferential attachment graphs, and real-world networks from two villages in India. I show results for BTER graphs and two of the Indian villages here and post the remainder in the Appendix.

I consider several ways to model the mitigator’s uncertainty about the graph:

- *Influential Node* uncertainty models uncertainty about which nodes are most connected, motivated by the fact the identity of the most socially connected and influential individuals is a function of the local culture which is more familiar to the influencer than the mitigator. Specifically, we start with a baseline graph, then, for each type, choose a set of j nodes and add k new randomly chosen edges from each of these nodes to others. It is important to note that in BTER graph, these j nodes are the *only* nodes that can potentially have inter-community edges under this uncertainty. These inter-community edges are particularly important in contagion games because they enable the spread of influence across groups.

- *Random Edge* uncertainty, is the simplest: the mitigator has perfect information about the nodes in the graph, and is uncertain about which edges out of a given set exist.

- *Inter-community Edge* uncertainty, models the mitigator’s uncertainty about a subset of the inter-community edges (i.e., which edges out of a given set of inter-community edges exist).

- *Inter/Intra-Community Edge* uncertainty, models uncertainty about a combination of inter-community *and* intra-community edges and addresses the concern that Inter-community

Edge Uncertainty may provide additional information by being restricted to the critical inter-community edges.

- *Inter-community Edge Set* uncertainty, models uncertainty over which *set* of inter-community edges exists (i.e., which set of 8 edges exists).

Note that in Inter-community Edge, Inter/Intra-Community Edge, and Random Edge uncertainty, we have a type λ for each possible subset of uncertain edges in the graph so the number of types could be as large as $2^{|E|}$. Also, the final three uncertainties, which highlight inter-community edges, apply only to BTER graphs. Here I only show results for Influential Node and Inter-community Edge uncertainty and post the rest in the Appendix. The results omitted are extremely similar to those for Inter-community Edge uncertainty.

The counterinsurgency literature [Hung, 2010] makes clear that military intelligence explicitly performs 'intelligence preparation of the battlefield (IPB)' to ascertain the structure and dynamics of a local population with high fidelity. Therefore, my study focuses on situations with a generally correct social network in which uncertainty is about the details of the network structure.

6.3 The Challenge of Uncertainty

The first question to ask is whether we can bound the impact of a small amount of uncertainty, because that may help us bound the total loss of solution quality given some uncertainty. I show that, in general, ignoring uncertainty can yield a solution that is arbitrarily poor for the mitigator. Consider the graph shown in Figure 6.1 in which the edge from A to B is uncertain, $N > M$, and both players have a single resource. Suppose that the influencer chooses to influence node A with probability 1. If the mitigator mistakenly assumes the edge does not exist, then his best

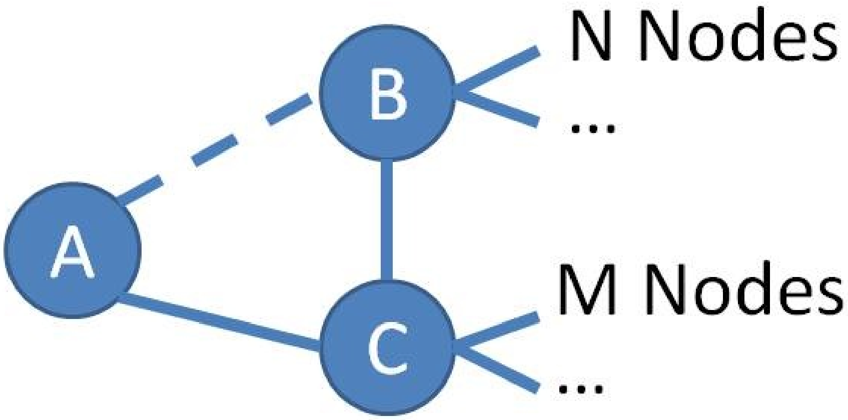


Figure 6.1: Unbounded loss

response is to influence node C with probability 1, but his actual loss amounts to $\frac{N}{2}$ as compared to the true best-response of playing B ($\frac{M}{2}$). A similar situation arises when the mitigator assumes the opposite. Thus, since M is arbitrary, by ignoring the uncertainty of just a single edge the mitigator can suffer unbounded loss.

The network in the above example is rather artificial, so it is natural to wonder what happens under a more realistic model of a network and uncertainty. To this end, I investigate the following empirical question: under our models of uncertainty, if we were to compute an optimal strategy assuming a single influencer type, how much would that strategy vary for different types? To answer this, take a Bayesian game with 40 types and compute an optimal mitigation strategy for each possible influencer type λ under the assumption of complete information. This yields a mixed strategy, Ω_M^λ, for each possible influencer type. Next, select a type b uniformly at random and measure the fraction of pure strategies in the support of ρ_M^b that is different from the pure strategies in the support of each ρ_M^λ for $\lambda \neq b$. In Figure 6.2 I report the *average* fractional difference over 20 independent instances of 40-type Bayesian games on 40-node BTER graphs

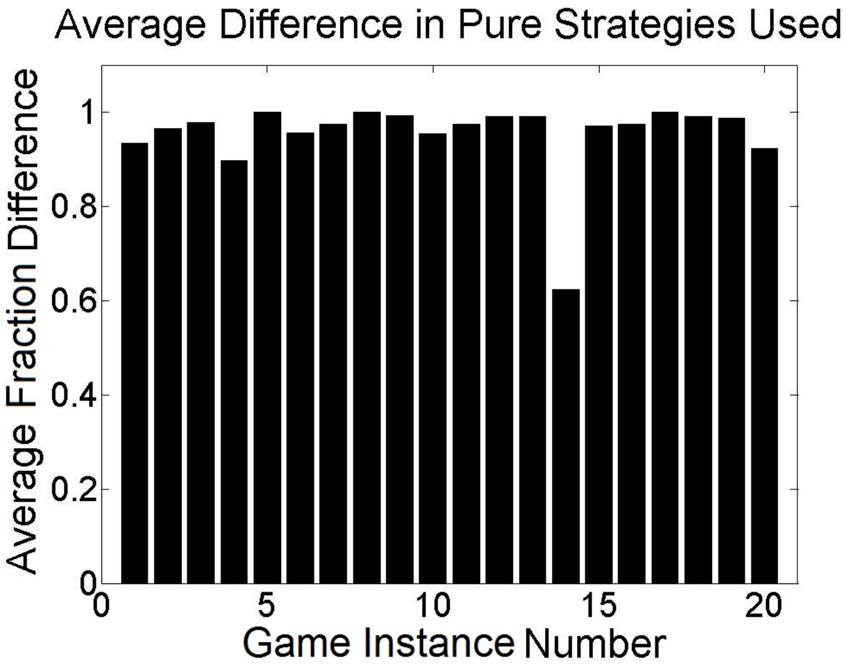


Figure 6.2: Comparison of mixed strategies

(edges vary from 130 to 200) with Influential Node uncertainty (more details on our standard setup will be presented shortly). Note that 1 in this case indicates that the mixed strategy for a randomly chosen type does not share a *single* pure strategy with the mixed strategy computed for *any* other type. As can be seen here for this instance, and is generally true under this uncertainty, nearly all instances show minimal overlap in the pure strategies used by each of the type-specific optimal strategies.

Finally, we turn to the question of complexity, where the result is very clear and very negative. At a high level, the challenge of efficiently reducing the runtime of computing equilibria in our setting lies in quantifying the impact of even small changes in the graph structure. If this could quickly and accurately be determined, then types could be efficiently clustered and bounds could be placed on the quality loss. The fact that computing the expected influence is #P-Hard [Chen et al., 2010b] should already give us pause. Indeed, a simple corollary of this result reveals that such quantification is intractable in general.

Proposition 1. *Computing the difference in expected influence for a given seed set even when a single edge is added to a graph is #P-Hard.*

Proof. Prove this by contradiction. Call the difference function, $d(S, G, e)$, where S is the given seed set, $G = (N, E)$ is the base graph, and e is the edge to be added. Assume $d(\cdot)$ can be calculated in polynomial time. Define a graph $G' = (N, \emptyset)$. $\sigma_{G'}(S)$ can be calculated in polynomial time. Repeatedly add edges from E to G' until G is fully reconstructed, computing $d(S, G, e)$ in each iteration. Since the total influence of G is $\sum_{e \in E} d(S, G, e)$, this implies that we have computed influence in polynomial time, since only $|E|$ iterations were executed, which contradicts that fact that computing the expected influence of a graph is #P-Hard. □

6.4 Double Oracle Algorithm

As in the previous chapter, even though influence blocking is a zero-sum game which can in principle be solved using linear programming, computing an equilibrium of this game in this case remains challenging. As before, payoff estimation is #P-Hard [Chen et al., 2010b] and the strategy sets for both players are exponentially large, making it impractical to store the entire set of payoff matrix. Novel in this model, however, is that because uncertainty is modeled over graph instances, the number of influencer types can be exponentially large.

The #P-Hardness problem was addressed in the previous chapter which introduced the LSMI heuristic which is also used here. While the standard double oracle formulation used in the previous chapter does not immediately carry over, a variation can serve to alleviate the exponentially large action spaces here as well. Specifically, the Bayesian double oracle algorithm introduced by Halvorson et al. (2009) also iteratively creates the payoff matrix. Instead of a single attacker oracle, however, we have an oracle for each attacker type and grow each type's payoff matrix separately.

Algorithm 7: Double-Oracle Algorithm for Bayesian zero-sum games

```
1:  Initialize $\mathbf{M}$ with random mitigator allocations.
2:  Initialize each $\mathbf{I}_\lambda \in \mathcal{I}$ with a random influencer allocation.
3:  **repeat**
4:     $(\rho_M, \rho_\mathcal{I})$ = MaximinLP($\mathbf{M}$,$\mathcal{I}$)
5:     $\mathbf{M}$ = $\mathbf{M}$ $\cup\{$MitigatorOracle($\rho_\mathcal{I}$)$\}$
6:     **for** $\{\lambda \in \Lambda\}$ **do**
7:        r = $\{$InfluencerOracle($\rho_M$, $\lambda$)$\}$
8:        $\mathbf{I}_\lambda$ = $\mathbf{I}_\lambda \cup$ r
9:  **until** convergence
10: **return** $(\rho_d, \rho_a)$
```

Algorithm 7 shows the full double oracle algorithm for Bayesian zero-sum games introduced by Halvorson et al.. The double oracle algorithm begins by initializing the mitigator and each influencer type with random actions. This subgame is solved via the call to MaximinLP with the corresponding mitigator and influencer equilibrium strategies stored in ρ_M and $\rho_\mathcal{I}$ (line 4). Then the mitigator's best-response oracle is called to determine the best action against the current influencer strategy. Then the algorithm iterates across all the influencer type best-response oracles and generates new actions to add to each subgame. The process then repeats until convergence.

This approach runs into the third and final problem: the exponential number of types. Since computing a best response for a given type requires a non-negligible amount of computation, having to do this for every type will simply not scale. To address this, I now show empirically that simple heuristics actually produce near-optimal solutions.

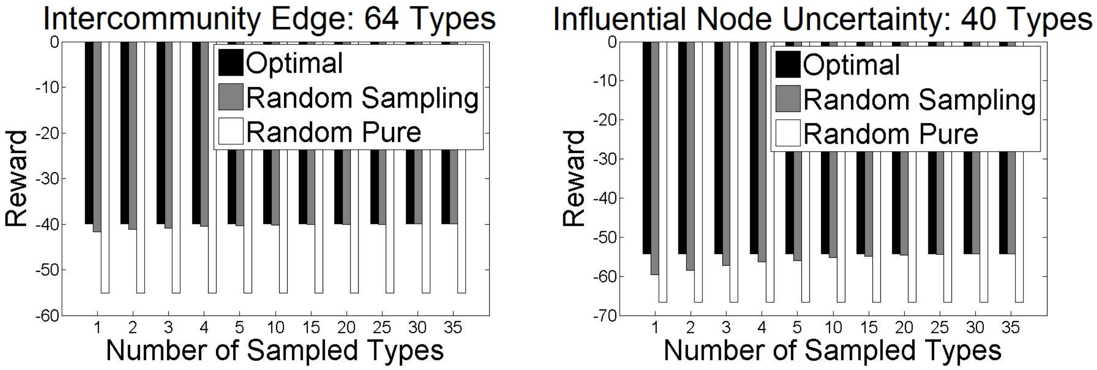
6.5 The Power of Simple

The results presented thus far, as well as the intuition from the vast literature on influence maximization [Leskovec et al., 2007; Chen et al., 2010b; Budak et al., 2011], suggest that carefully accounting for our uncertainty about graph structure is crucial to obtaining high quality solutions. Next, I present a small, representative subset of an extensive collection of experiments, all showing precisely the opposite: we need only to randomly sample a few types from the type distribution and solve the resulting game as if no other types exist, to obtain solutions that are nearly optimal. This is quite surprising, particularly since I have already shown, via the example in Figure 6.1, that ignoring even a single influencer type can yield arbitrarily poor solutions even with only two types.

All the results below are an average of 20 game instances and were run on machines with CPLEX 12.2, 2.8 GHz CPU, and 4GB of RAM. Unless otherwise stated, experiments were run on 40-node graphs (130 to 200 edges), contagion probabilities on edges drawn from a $\mathcal{N}(0.4, 0.2)$ distribution, node values varying uniformly from 1-10, each player having two seed nodes ($|\omega_M| = |\omega_I| = 2$), and payoffs estimated using the LSMI heuristic introduced in Chapter 5. Monte Carlo payoff estimations produced similar results but could not be meaningfully scaled. Since an optimal benchmark is necessary, the best-response oracles iteratively evaluate each available action to determine the best response, rather than using greedy hill-climbing common in the influence maximization literature. Unless otherwise stated, Influential Node uncertainty selects 3 nodes and gives each 4 additional edges. Moreover, *only* these 12 edges could potentially connect communities, making the chosen nodes not only more connected (average degree, excluding uncertain edges, varies from 3-5 with maximums of 9), but also incident to the more consequential edges.

For Inter-Community Edge uncertainty I varied the number of uncertain edges between 1 and 6 (the optimal technique could not scale to more edges). I focus throughout on the mitigator strategy obtained by drawing a random subset of the influencer's types and solving the game assuming no other types exist (referred to as *Random Sampling*).

6.5.1 Experiments

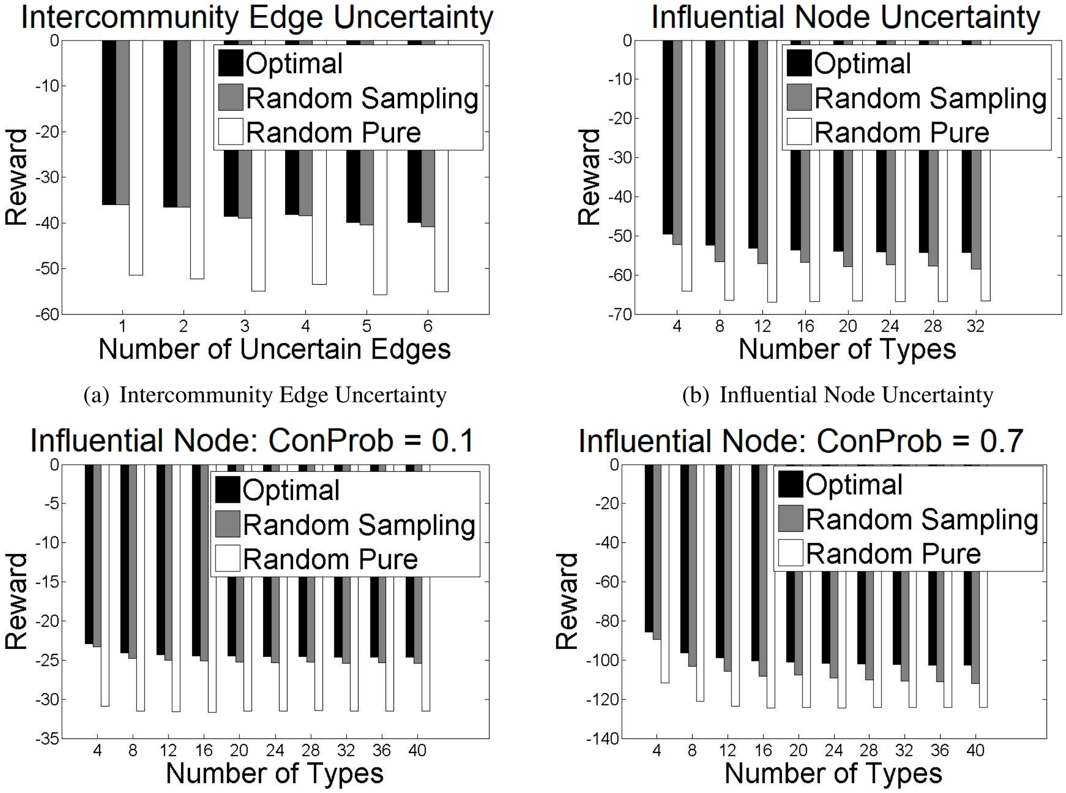


(a) Inter-comm. Edge

(b) Influential Node

Figure 6.3: Reward comparison, BTER graphs

In my first set of results, shown in Figure 6.3, I consider the impact of the number of randomly sampled types on solution quality (only a combination of BTER and two models of uncertainty are shown here, as these results exhibit the *greatest* approximation error from random sampling; extensive other studies, included in the Appendix, offer even more dramatic support of our argument). These experiments use the *same* 40-node games that were featured in Figure 6.2 that showed pure strategies used by individual types have minimal overlap. The x-axis shows the number of sampled types, while the mitigator utility is plotted on the y-axis. The key point is that with only about 2-5 randomly sampled types we obtain a solution that is very nearly optimal, despite the fact that only using a single influencer type yields a relatively poor mitigator reward (Figure 6.3b). While results in the optimization literature such as sample average approximation theory [Shapiro and Homem-de Mello, 2000] show that random sampling can converge exponentially fast to optimal solutions, this “convergence” is uncannily quick.



(a) Intercommunity Edge Uncertainty

(b) Influential Node Uncertainty

(c) Influential Node Uncertainty, ConProb = 0.1

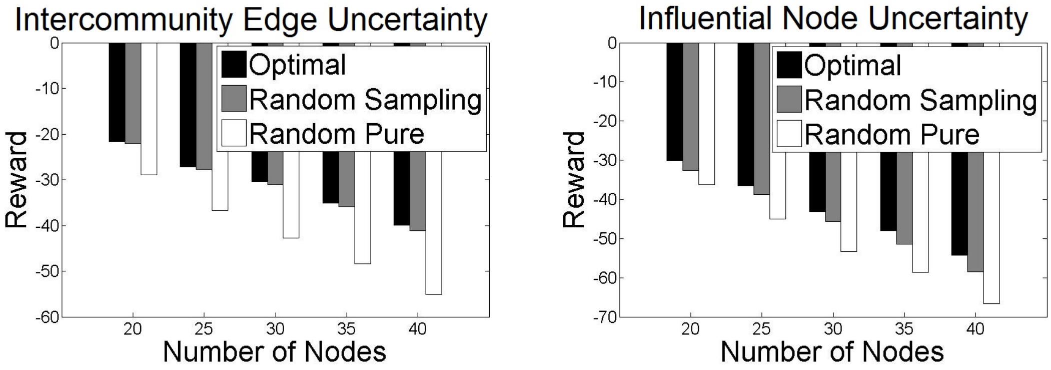
(d) Influential Node Uncertainty, ConProb = 0.7

Figure 6.4: Type scale-ups

Next, I fix the number of randomly sampled types used to generate a solution at 2, and increase the number of *actual* types (increasing the degree of uncertainty). The graph sizes were fixed to 40 nodes. Intuitively, we would expect that the performance of *Random Sampling* should degrade significantly as we increase uncertainty by adding types. Figures 6.4a-d are representative of a broad array of experiments I ran in this space (see Appendix). In addition to considering several types of uncertainty, I also varied the average contagion probability of edges from the 0.4

it is by default to 0.1 and 0.7 as well (denoted ConProb = 0.1, 0.7). Perhaps the most surprising finding in these experiments is that the quality of *Random Sampling* relative to optimal degrades very little as we increase the number of types. While I could not compute optimal solutions for games involving more types, this finding suggests that we may need to sample a decreasing (rather than a constant) fraction of all possible types as the number of total types increases.

In my final set of results using synthetic graphs, I study the impact of the size of the underlying network. The number of edges varied from 28 (20 nodes) to 188 (40 nodes) with up to 6 edges differing between types for Inter-community Edge uncertainty and up to 24 edges for Influential Node uncertainty (12 new edges per type). Here, I keep the number of nodes/edges about which we are uncertain fixed, and increase the network size. Consequently, the expectation is that smaller networks would exhibit significantly greater difference between random sampling and optimal, since uncertainty involves a greater fraction of the graph. Figure 6.5 shows little evidence of this: the quality of simple heuristics relative to optimal is little affected by the fraction of the graph that is uncertain.



(a) Inter-comm. Edge

(b) Influential Node

Figure 6.5: Scale-up of graph size

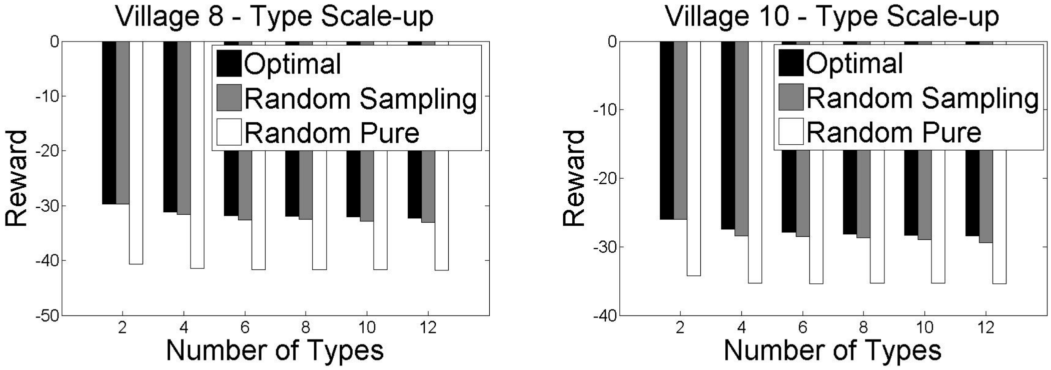
In all, I studied variations involving BTER, preferential attachment, and small-world generative models of networks, all five models of uncertainty described previously, and, moreover, considered numerous variations in the parameter space of all the generative models of graphs and uncertainty about them, the number of resources that players had, etc. For example, I studied games in which the mitigator was allowed to initially impact 3 or 4 nodes, while the influencer was restricted to 2, and vice versa; I varied the density of communities in BTER graphs; varied the degree of perturbation between types. All these results (see Appendix) exhibit essentially the same trends that I show here.

Finally, I conducted a set of experiments on a real-world social network dataset released in 2012 that was obtained via survey data in 75 Indian villages.[1] The survey asked the inhabitants of the villages a series of questions to ascertain their relationship with other people in the village (e.g., would you invite him in for tea, do you go to temple with him, would you loan him money, etc.). From this data, a social network can be constructed by beginning with a complete graph with edge weights of 0.0, increasing the weight of an edge corresponding to a positive answer to a survey question by an amount based on the perceived importance of the question, and then normalizing all weights. More details on the survey questions and the weighting scheme can be found in the Appendix.

For my experiments, I use the household-level data for two of the smaller villages (8 and 10), because even the double-oracle optimization does not scale to larger networks. The results in Figure 6.6 use Influential Node uncertainty, and each type now chooses 8 random nodes and gives each 10 new edges to maintain the same fraction of uncertainty, since the India data sets

[1] Abhijit Banerjee; Arun Chandrasekar; Esther Duflo; Matthew Jackson, 2011-08, ”Social Networks and Microfinance”, `http://hdl.handle.net/1902.1/16559` UNF:5:4EmgOYAQGaoQugFowckNfA== Jameel Poverty Action Lab [Distributor] V5 [Version]

have 77 or 94 nodes and an average degree of 7.7 or 7.4. As the figure testifies, my results are not an artifact of synthetic graph models that I generate, but can be observed on graphs based on actual social network data as well. Furthermore, to alleviate concerns that the weighting scheme may be biased, I also tested alternate weighting schemes with the results virtually identical to what I show here (see Appendix).



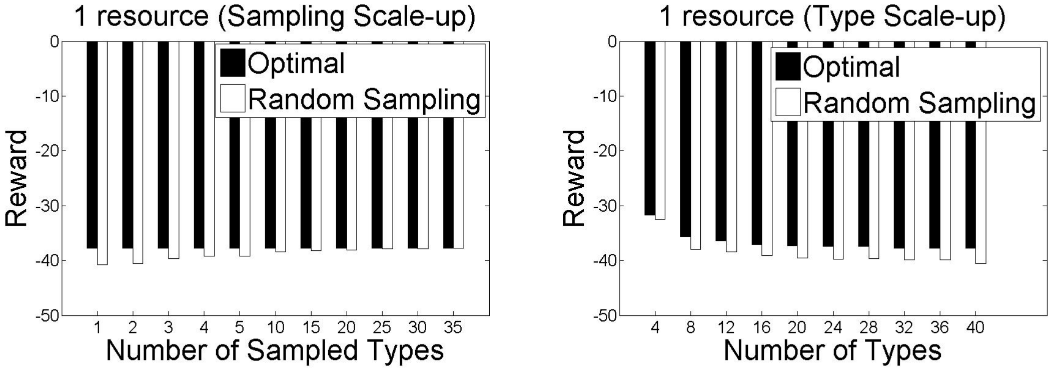
(a) Village 8

(b) Village 10

Figure 6.6: Influential Node uncertainty, Variable-Weight-Scheme

6.5.2 Analysis

The results shown are surprising in their extremity, especially in light of the result presented previously demonstrating minimal overlap of pure strategies in optimal strategies for individual types. As mentioned before, Lahiri et al. (2008) found that nodes that generated high influence spread remained top performers despite graph perturbations in denser networks, but in sparser networks this was less true. This is not the case in my work, since I have results across a range of graph densities including sparse scale-free networks and dense BTER graphs. Fibich and Gibori (2010) note that in their setting, spreading from multiple sources is robust to network structure perturbations but that spreading from a single resource is highly sensitive to them. In results previously shown, both players were given two resources. To examine the validity of Fibich and Gibori's claims in my scenarios, I augment my results with scenarios where each player is given only one resource. For these games, a 40-node BTER graph was used with Influential Node Uncertainty and the remaining parameters were fixed to the defaults unless otherwise specified (40 types, 2 sampled types). Figure 6.7a shows scale-up of the number of types sampled and Figure 6.7b shows scale-up of the number of types in the game. As can be seen, both of these figures show that even with only one resource, random sampling still performs near-optimally.



(a) Sampling scale-up

(b) Type scale-up

Figure 6.7: Results for games with one resource

While past researchers have found the number of resources and the density of graphs to be key features, evidence shows that these attributes do not explain the phenomenon exhibited in my work. Examining my results more closely, two other features appear critical. First, graphs with a small set of nodes with very high *influence spread* and a vast majority of low influence spread nodes are particularly robust to the uncertainties tested (e.g., a sharp power-law distribution or even a steep linear distribution). Second, the two-player game itself is more resilient to the uncertainties modeled in my work as compared to the one-player variant (i.e., influence maximization).

With respect to the influence spread distribution, consider a scale-free graph with 40 nodes. Figure 6.8 shows the expected influence spread achieved by selecting the given node, sorted left-to-right from lowest to highest. The black bar shows the expected influence spread of the node in the base network and the white bar shows one example of expected influence spread in a perturbed network. As can be seen in the black bar, there is a steep increase in influence spread with the rightmost nodes achieving rewards of 10-12. Although there is a steady increase in influence spread of nodes as opposed to a flat majority with a sudden spike, the sheer steepness of the increase means that introducing uncertainty by adding edges will, with high probability, have no impact on the identity of the top influencers in the graph. The white bar shows results for a perturbed network in which three random nodes are given four new random edges (i.e., influential node uncertainty). As can be seen, the top nodes have hardly changed despite the addition of these edges. Thus, despite the existence of numerous Bayesian types, the top spreaders remain nearly constant across all types, implying that the best actions for each player in a randomly sampled type will also be the best actions for most other types. Thus, with high probability, optimally solving a random sampling of the types will produce aa near-optimal solution.

Finally, when there is significant change in the top spreaders, the two-player game appears to be more resilient to the uncertainties modeled as compared to the one-player variant explored in past works. Consider, for example, a 40-node random scale-free network with 10 isolated nodes with no incident edges and extremely low node value. To achieve a change in the top spreaders, we use a variation of influential node uncertainty where we grant a single node 15-20 edges instead of granting three nodes four additional edges each (maximum degree in networks tested varied from 7-15). We can then create a game in which we have 10 Bayesian types, each one of which grants one of these isolated nodes a set of new edges.

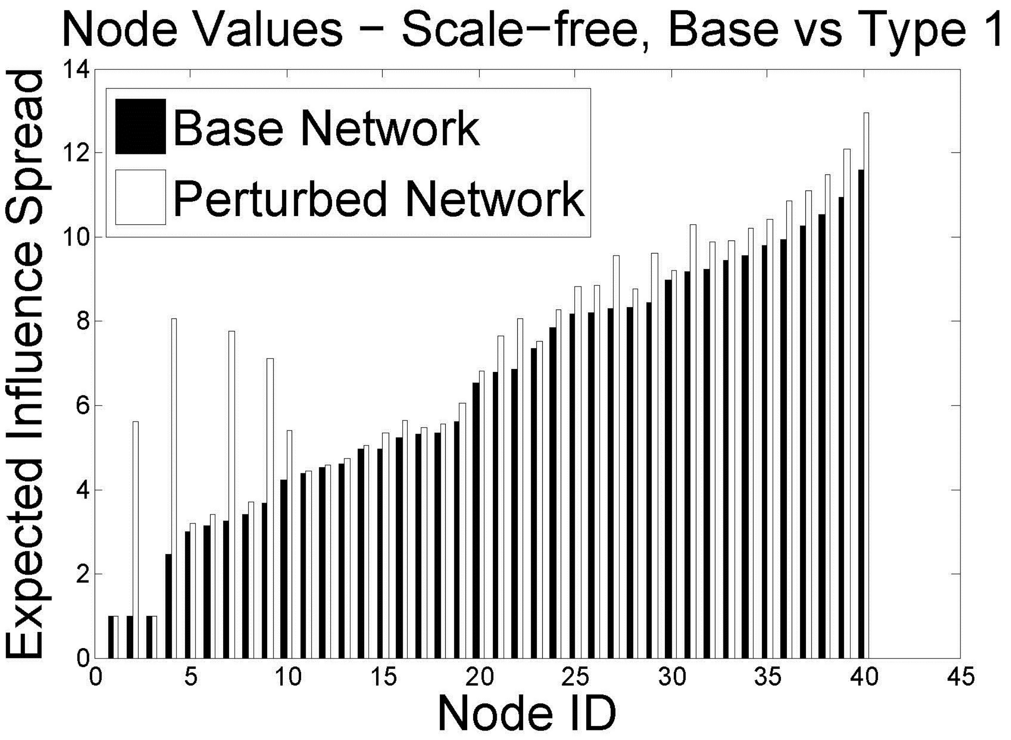
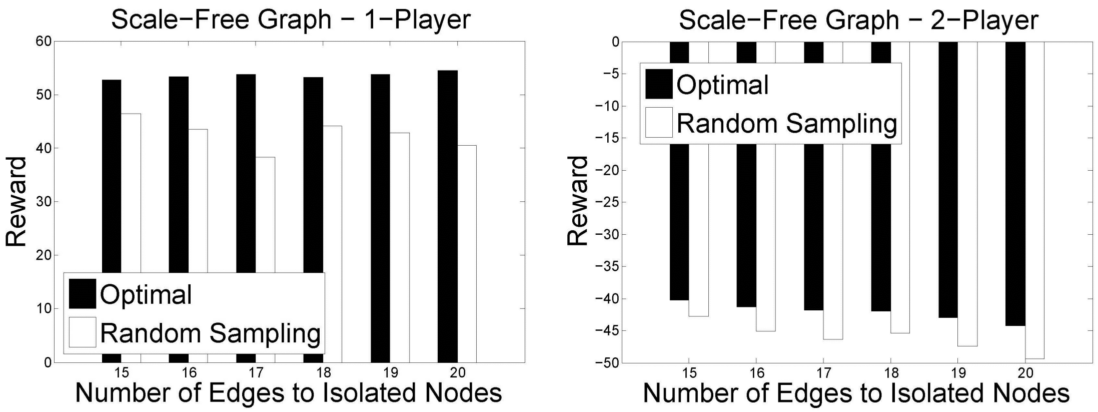


Figure 6.8: Expected influence spread by node, scale-free graphs

In a one-player influence maximization problem with one resource allowed, the optimal answer will be generally be one of the nodes within the connected scale-free network[2]. However, when a single type is randomly sampled, the optimal solution to the subgame will often be the isolated node that was granted new edges in that type. Such a strategy will perform phenomenally in the single type that it was optimized for, but will perform abysmally in all other cases, yielding a very low reward overall. This is evidenced in Figure 6.9a, which shows the results for the one-player influence maximization problem for this game (averaged over 20 trials). Here, random sampling performs more than 20% worse than optimal. When the same game setup is used for the two-player game, however, random sampling of even just a single type performs within 10% of optimal. These results are shown in Figure 6.9b.

[2] Assuming similar transmission probabilities on newly added edges and sufficient edge density within communities.



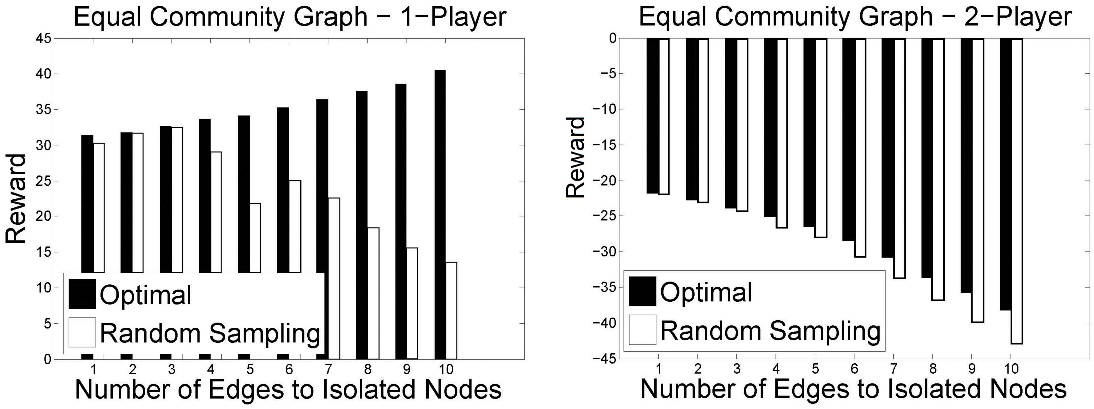
(a) One-player game

(b) Two-player game

Figure 6.9: Results for scale-free networks

This is even more pronounced when we consider a BTER-style graph with 5 densely connected communities, each with 8 nodes, each unconnected to the other communities. Again, we add 10 isolated nodes. gain, we can create a game in which we have 10 Bayesian types, each one of which grants one of these isolated nodes 1-10 new edges (the maximum degree in the networks tested was 5). As can be seen in Figure 6.10a, as we scale up the number of edges granted there is a dramatic drop in performance of random sampling in the one-player game as the number of edges exceeds 4 (nearly 70% worse than optimal at 10 edges). In the two-player variant, however, we only see a gradual degradation in random sampling's performance as the degree of network perturbation is increased, with the worst case being within 11% of optimal.

Thus, the extremely strong performance of random sampling in this setting appears to arise from a combination of features. In some cases the graph structure results in a large spread between top influence spreaders and other nodes that is naturally robust to mild uncertainties. In other cases, the two-player aspect of the game generates additional robustness beyond that of



(a) One-player game

(b) Two-player game

Figure 6.10: Results for one- versus two-player game

the one-player game where past researchers have found dramatic changes due to random graph perturbations.

Chapter 7: ESCAPES Evacuation Simulator

The previous three chapters introduced game-theoretic resource allocation techniques for optimizing the use of resources in games with networks and contagion. This chapter is the first of three that discuss my studies into contagion phenomena and begins with my analysis, in simulation, of the interaction between the contagion of fear in an evacuating crowd and the fear mitigation effects expected of an authority figure. To accomplish this, I develop an evacuation simulator, ESCAPES, which incorporates fear contagion and authority figures among other features that provide a novel combination of attributes yielding emergent behaviors observed in evacuation literature. Using this simulation, I provide analysis of fear and escape rates as well as recommendations for authority figure policies.

This chapter is divided into four sections. First, since the creation of a simulation touches upon a vast literature unrelated to either game theory or contagion specifically, I include a discussion of evacuation and pedestrian simulation research in Section 7.1. Second, in Section 7.2 I describe the individual agent types that are included in the simulation including authority figures, individual travelers, and family agents. Third, in Section 7.3 I describe the interactive elements of the simulation including emotional contagion and fear mitigation, the spread of knowledge about exits and events, and social comparison theory. Finally, Section 7.4 features a series of simulation runs exhibiting numerous emergent trends found in evacuation literature as well as simulations of a terminal at Los Angeles International Airport. These simulations demonstrate not only the fidelity of the features modeled but also the use of simulation for authority patrolling policy recommendations.

7.1 Related Work

Early work in pedestrian dynamics noted the similarity between crowd behavior and well-understood phenomena observed in physics. These observations led to the development of models based on fluid-dynamics [Henderson, 1974]. as well as gas-kinetics [Hoogendoorn and Bovy, 2000]. In these models, pedestrians are represented by particles whose motion is governed externally by the physical equations for the given medium. Another approach to force-based crowd simulation is built off the idea of social forces [Helbing and Molnar, 1995]. Instead of being based on the physical properties of water or gas, social forces represent the attractive and repulsive forces felt by a pedestrian toward various aspects of its environment. These forces can then be combined to form an equation for describing the motion of a pedestrian. Yet another approach involves the use of cellular automata (CA). In CA-based models [Burstedde et al., 2002], the environment is divided into a grid consisting of cells. At each time step, a cell transitions to a new state based upon its current state and the states of the neighboring cells. Due to their simplicity, in terms of both computation and representation, CA-based models have been used extensively to simulate crowd evacuations[Burstedde et al., 2001; Song et al., 2005; Tissera et al., 2007]. However, in both forced-based and CA-based models, it is difficult to simulate goal-driven and heterogeneous behavior. Thus, the specific crowd phenomenon we are looking at are not typically modeled with these approaches.

Agent-based models allow for each pedestrian to be modeled as an autonomous entity. Under this model, pedestrians are represented as agents capable of perceiving and interacting with their environment as well as other agents. While being the most computationally expensive modeling technique, agent-based models are capable of a higher degree of expressivity and fidelity. The ability to represent cognitive information and model complex and heterogeneous behaviors has opened the possibility for new avenues of research that had not been attempted with previous methods.

As a result, there has been a shift toward the use of agent-based models for evacuation simulations. However, much of this research has been focus solely on modeling the physical interactions between agents[Lin et al., 2010]. The EXODUS[1] system represents the state-of-the-art for these systems with versions specifically for various types of large-scale scenarios and additional modules that can model phenomena such as toxic gas and fire spread. The system does move slightly beyond physical interactions to include informational aspects such as signage and exit familiarity, but still does not attempt to use psychologically-based decision-making in their agents.

Despite this trend, there has been some interest in incorporating emotional as well as the informational interactions into agent-based models. The complex relationship between the spread of information and the spread of emotion was explored from a theoretical modeling perspective in [Hoogendoorn et al., 2010]. In [Chao and Li, 2010], the effect of communication on collective behavior was analyzed by simulating riots. The reflection of individual personality traits in the personality of crowds was studied in [Durupinar et al., 2008]. Work of this nature tends to focus

[1] http://fseg.gre.ac.uk/exodus

on modeling a small subset of emotional and information interactions with a high level of fidelity. A comprehensive approach is presented in [Pelechano, 2005] focuses on creating agents with sophisticated psychological models. My research is less concentrated on individual agents and more concerned with the interactions between agents and the resulting group dynamics. Additionally, ESCAPES is focused on a different set of domains including airports, malls, and museums. To accurately represent these types of environments, it is particularly important to model the influence of families, emotional contagion, social comparison, and spread of knowledge, which past work has not cohesively addressed.

7.2 Agent Design

The ESCAPES system is a two-part system comprised of a 2D, OpenGL environment based in the open-source project OpenSteer[2] and a 3D visualization component using Massive Software[3]. The 2D module consists of agents as described below, outputting their physical and behaviorial information into files that are then imported into customized Massive extensions to generate 3D movies of the scenarios. The 2D module can be used for efficient statistical analysis of different security policies. As mentioned previously, the 3D visualization is a key component for airport security officials, as it provides a superior training medium to their current tools. Screenshots in Figure 7.1 show the children models as well as some people running in different directions (denoted in the white circle) when an evacuation begins. Here we describe the architecture of the 2D module, first introducing the individual traveler agent, then detailing two special agent

[2] http://opensteer.sourceforge.net

[3] http://www.massivesoftware.com

categories (families, authorities), and finally discussing interaction level dynamics (spread of knowledge, emotional contagion, social comparison).

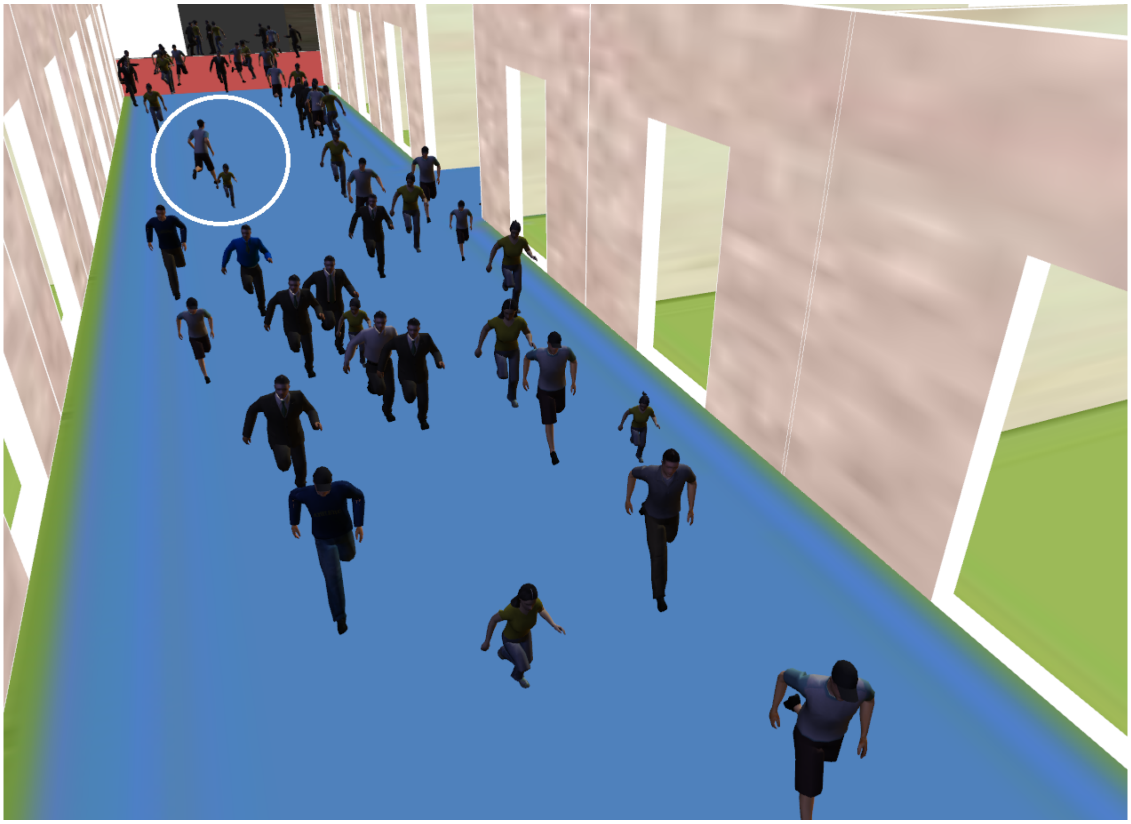


Figure 7.1: ESCAPES 3D visualization

7.2.1 Individual Travelers

All agents share a common architecture based in a BDI framework, possessing varying degrees of knowledge about the world and their neighbors. Each agent has access to a subset of the 14 available behaviors, any one of which may be active at a given time, where the behavior is selected via a probabilistic weighting scheme. The weighting scheme is a combination of 6 'Cognitive Mechanisms,' each of which prioritize some of the agent's desires. For example, there is a Cognitive Mechanism that prioritizes the basic desire of an agent to 'Wander' through his environment or ‘Shop’ in the stores. On the other hand, there is another Cognitive Mechanism that prioritizes an agent’s desire to survive by evacuating through an exit once an event has occurred via one of the escape behaviors (‘Run to Nearest Exit’, ‘Run to My Exit’, and ‘Search for Exit’). During execution of these behaviors, individual travelers may move at integer speeds from 0 to 3.

Each agent also has specific levels of emotions and information about the environment. Studies have shown that emotional stress causes changes in decision-making and may even cause someone to forget where he/she entered a building from [Chertkoff and Kushigian, 1999]. Combined with the incomplete knowledge of a person that is in a place for the first time, which occurs extremely frequently in the airport scenario that I model, an evacuation suddenly becomes much more difficult to manage. Thus every agent has a fear level, an event certainty level, as well as a list of known exits. A more extended discussion of these attributes will take place in Section 7.3, but we briefly mention their implementation here first.

Fear is modeled as an integer value between 0 and 2 (*FearFactor*), 0 indicating that the agent has no fear. Higher levels of fear lead to higher movement speeds to get out of the area as soon as possible. Each agent’s fear is a result of a number of factors such as their proximity to the event, the presence of authority figures nearby (as a result of documented impact of authority figures on evacuees [Diamond et al., 2010; Smith and Ellsworth, 1985]) and the level of fear in neighbors and family members (as a result of contagion [Hatfield et al., 1994]).

Event certainty is modeled as an integer value between 0 and 2 (*EventCertainty*), designating how aware the agent is that an event has occurred and that, therefore, an evacuation is necessary. An event certainty level of 2 is generated only by people close to the event, who immediately run directly away from the event before beginning active exiting behavior. Further away agents may have 1, which immediately triggers exiting behavior. Agents furthest away have an EventCertainty of 0 and continue their normal behavior, as they are unaware of any need to evacuate. Each agent's EventCertainty level is dictated by their proximity to the event, the presence of authority figures nearby that would inform them of the event, and the event certainty of neighbors via the Spread of Knowledge mechanism discussed in Section 7.3.1. The importance of uncertainty about an event has been noted in evacuation literature as a major cause of delay and, therefore, casualties [D.S.Mileti and J.L.Sorensen, 1990].

Exit knowledge is modeled as a binary value indicating whether or not an agent knows about a given exit. Given a list of known exits, if an agent decides to evacuate, he will choose the nearest one. Exit knowledge is dictated by where they entered from, a random chance to forget that exit, and the presence of authority figures nearby that would inform them of exits. A person's knowledge of exits are clearly of paramount importance in any evacuation situation, especially in airport scenarios where many people are first-time visitors and are unaware of the environment layout.

7.2.2 Family Agents

Evacuations in some environments pose additional challenges as a result of the population present. In the airport scenario that we focus on, families have been identified as an important facet of the environment that must be modeled to more realistically portray the situation [Diamond et al., 2010]. One can see how this might differ from the evacuation of an office building where only knowledgeable adults are present. For instance, children often rely on their parents to lead them and parents will undoubtedly seek out each other and their children before exiting, oftentimes disobeying authority instructions [Proulx and Fahy, 2008].

I model the presence of family units composed of 2 parents and 2 children with behaviors and cognitive mechanisms not applicable to general agents. Prior to an evacuation, children usually execute the ‘Follow Parent’ behavior, except occasionally executing the ‘Drag into Shop’ behavior which leads their parents into nearby stores that they find interesting. To enhance realism, we also restrict children to slower movement speeds (maximum of 2), which parents leading them will inevitably match. Parents that are not with their children heavily prioritize finding them via the ‘Find Child’ behavior, and put some emphasis on the ‘Find Other Parent’ behavior (they may also Wander or Shop). When an evacuation occurs, however, parents immediately seek each other out to gather the family together before proceeding to an exit, as has been shown to occur in real evacuations [Proulx and Fahy, 2008]. After an evacuation is underway, children will no longer execute the ‘Drag into Shop’ behavior, resorting exclusively to ‘Follow Parent’.

7.2.3 Authority and Security Agents

Studies have shown that some authority figures have a very strong calming effect on people in an evacuation situation [Smith and Ellsworth, 1985]. This can come through implicit calm at the sight of other people that appear calm via emotional contagion and may be enhanced due to the uniformed authorities having a stronger contagion effect due to their leadership role [Hatfield et al., 1994]. Also, by simply being there everyday, authorities know the environment and are trained to properly direct people to the nearest exits in the event of an emergency.

In my simulator, under normal conditions, authority agents ‘Wander’ or ‘Patrol’ the environment. After an event occurs that necessitates an evacuation, all authority figures switch to ‘Patrol’ in an attempt to inform everyone of the event and where nearby exits are located. I also set the

FearFactor of authority figures very low and keep it constant to mimic well-trained security personnel that can maintain a level head in volatile situations. The calming effect they have on other agents is modeled by overriding nearby agents' FearFactor with the authority figure's FearFactor. The practical effect of this is to slow agents down (since FearFactor directly impacts travel speed), which may increase the total evacuation time, but also reduces the severity of colliding and the level of chaos. Also, authorities know all exit and event locations and pass this information to agents that are nearby.

7.3 Agent Interactions

With the existence of crowds, agent interactions are a fundamental aspect of the ESCAPES evacuation simulation. Thus, I base the agent interactions on existing evacuation and social psychology research. I incorporate a realistic 'Spread of Knowledge' of events and exits, an Emotional Contagion module to model the infectious nature of emotions, as well as a social comparison component to capture people's mimicry of others.

7.3.1 Spread of Knowledge

As mentioned, while unimportant for office building or railway station simulations, realistic knowledge spread to model the behavior of first-time visitors is a necessary component in an airport simulation. Thus, I model the spread of two types of knowledge in our system: Exit Knowledge and Event Knowledge.

7.3.1.1 Exit Knowledge

People entering an environment for the first time will possess incomplete knowledge of exit locations. Thus, they must rely on authorities, signs, and following the crowd to make their way towards the nearest exit if there is one closer than the one they entered from. It has been shown that in times of high emotional stress, people even forget where they entered [Chertkoff and Kushigian, 1999].

ESCAPES includes this level of realism, giving agents knowledge of their entry location and a random chance that they forget this knowledge. In contrast, authority figures begin with and maintain full knowledge of all exit locations and pass a limited subset of this to nearby agents to simulate their redirection of passersby to the nearest exits. Also, family members will inform each other of exits they find out about, but otherwise, agents do not communicate exit knowledge to each other. Agents are also able to use the 'Search for Exit' behavior to find a way out on their own or some may choose to simply follow nearby, similar agents via the SCT module's 'Follow Most Similar Agent' behavior.

7.3.1.2 Event Knowledge

In real emergency situations, pre-evacuation delay has been cited as a major cause of slower evacuations and, therefore, deaths [D.S.Mileti and J.L.Sorensen, 1990; J.L.Bryan, 2002]. This delay is largely due to a lack of knowledge about the emergency, both in disbelief of the severity of the situation as well as a desire to find out more about what has occurred. Pre-evacuation delay has been noted to persist despite verbal warnings and physical cues in the environment [J.L.Bryan, 2002].

In ESCAPES, agents that are near the event as it occurs will have full knowledge of what has occurred, whereas agents far away have no idea are unaware that anything is wrong. As civilians pass each other, they communicate their level of certainty to each other, raising awareness of the situation. As civilians become more aware, they are more likely to run towards the exit as their self-preservation desires take precedent over all other desires.

Authority figures are assumed to instantly know when something has occurred, simulating an immediate radio notification from central security personnel. This does not necessarily translate into an immediate announcement to the general public, since oftentimes the appropriate response is not immediately obvious. Authority figures also communicate their certainty of the event to nearby agents, mimicking an actual authority figure telling people to evacuate.

7.3.2 Emotional Contagion

Emotional contagion is the effect of one person's emotional state on the emotional state of people around him/her both explicitly and implicitly [Hatfield et al., 1994]. It has been observed in families, small-scale interactions as well as large crowds [Forgas, 1990; Hatfield et al., 1994]. Researchers continue to develop theories on the phenomenon and are still exploring the various factors that are believed to influence the level of contagion as well as its effect on decision-making.

In an evacuation scenario, fear abounds, due both to uncertainty of the situation as well as concern for one's own safety [Smith and Ellsworth, 1985]. As a result of emotional contagion, bystanders that are unaware of the event may develop otherwise inexplicably high levels of fear as well. Their subsequent decisions and behaviors as a result of this 'inherited' fear have not been explored in the context of a crowd or evacuation simulation. We therefore propose a baseline implementation and analysis of a model of emotional contagion.

Specifically, there are two components that spread emotions amongst agents. First, as agents pass by each other, they inherit the highest level of fear of neighboring agents. This is the baseline emotional contagion model that conforms with a theory of emotional contagion in which the highest level of emotion is transferred to all surrounding agents and inherited at full effect [Hatfield et al., 1994]. Second, as agents pass by authority figures, their level of fear is reduced to the authority figure's fear level. This simulates the implicit and explicit calming effect of authorities and conforms with a theory of emotional contagion that allows for specific agent types to reduce the level of emotion of surrounding agents (e.g., an agent that is greatly respected by all surrounding agents [Hatfield et al., 1994]).

7.3.3 Social Comparison (SCT)

Social Comparison Theory [Festinger, 1954] is a social psychology theory, initially presented by Festinger. It states that humans, when facing uncertainty, compare themselves to others that are similar to them, and act towards reducing the differences found. Social comparison is considered a general cognitive process, which underlies human social behavior. During emergencies, individuals face greater uncertainty, and thus the weight of social comparison in human decision-making is increased [Kulik and Mahler, 2000].

I find the utilization of the computational model of social comparison [Fridman and Kaminka, 2009] helpful in developing agents with the social skills that are crucial to the accurate simulation of different crowd behaviors. The SCT computational model can be used, for instance, by agents who wish to urgently exit an area. If they do not know the location of a close exit, they may turn to mimicking others hoping that they will lead them to safety.

For the simulation, SCT was implemented as follows. First, the agent compares itself to others around it by measuring the similarity in a set of features, including speed, emotional state, distance, etc.. The similarity values are combined, and the agent that is most similar (within bounds) is selected. The agent executing SCT takes actions to reduce dissimilarities to the selected agent. In this simulation, SCT increases the tendency to mimic someone else's behavior, whereas emotional contagion transfers emotions regardless of what different behavior will be chosen based on it.

7.4 Evaluation

I conducted extensive testing using a generic scenario to evaluate the impact of the emotional and informational phenomena modeled in ESCAPES. The scenario takes place in a generic airport setting consisting of 2 gates, 3 hallways, and 14 shops. There is an exit in each gate as well as the end of one of the hallways. Unless otherwise noted, the experiments for the generic scenario feature the following: 100 travelers which includes 10 families, 10 authority figures, emotional contagion, spread of knowledge, and social comparison. Simulated evacuations are typically evaluated by examining the rate at which people evacuate. While, evacuation rate is obviously important there are other metrics which can also provide insight as to how an evacuation proceeded. In Sections 5.1-6, we analyze the results from these experiments using the metrics which best highlight the effect of the various phenomena. Additionally, I modeled Tom Bradley International Terminal at Los Angeles International Airport and ran proof-of-concept tests on this to evaluate performance on a real domain. A description of the scenario and accompanying results is provided in Section 7.4.7

In all of my experiments, an event occurs during the 14th time step and travelers have until the 300th time step to evacuate. It is assumed that by this time, airport officials will have managed to coordinate in response and issue a general order to evacuate through their emergency broadcast system. All the results in this section have been averaged over 30 independent simulations.

7.4.1 General Testing

As mentioned in previous sections, current evacuation simulators tend to focus on the physical interactions of agents. The agents in these simulations are typically homogeneous, rational, and omniscient. In contrast, ESCAPES agents are heterogeneous, emotional, and limited in both knowledge and perception. In Figure 7.2, I compare the evacuation rates from simulations in which the population of travelers is modeled as homogeneous, omniscient agents to those in which the population is modeled as ESCAPES agents including authority figures and families. The y-axis represents the percentage of travelers who have yet to evacuate. This percentage will decrease over time and the slope of the line signifies the current rate at which travelers reached safety. For example, after 85 time steps we can see all travelers have evacuated in the physical interaction model whereas 25% of travelers have yet to evacuate in the physical, emotional, and informational model.

When modeling omniscient agents, simulations consist of travelers with complete knowledge who are not influenced by their emotions. The only relevant interaction between travelers occurs when there is congestion due to an area becoming overcrowded. When the event occurs, all travelers are able to perceive it instantaneously and begin to head for an exit. We see a steep decline in the number of unevacuated travelers, as those close to an exit evacuate rapidly. There is then a temporary decrease in the rate of evacuation as those travelers who were far away from an exit rush towards it. Once those travelers start reaching the exits, the rate of evacuation picks up again until everyone has evacuated. While these models can provide a good first order approximation, they fail to capture much of the underlying complexity present in evacuations.

With travelers who are more realistic, the evacuation rate is slower. This is due to a multitude of factors such as families taking time to find their loved ones, travelers never learning about the event, or travelers having limited knowledge about exits. Unlike when travelers are modeled as omniscient agents, situations arise with ESCAPES agents where there are travelers who are unable to evacuate in time. However, it is important to examine these situations because it is exactly these scenarios where the potential for danger is greatest were they to occur in real life. Models using omniscient agents provide best-case scenarios and a lower bound on evacuation times. While this information is useful, a system that is capable of modeling unforeseen worst-case scenarios, such as ESCAPES, will be more effective as a training and policy-making tool.

7.4.2 Families

Studies have shown that the presence of the families results in slower evacuation times [Proulx and Fahy, 2008]. I tested the effect of families on evacuation rate by comparing the results from simulations with varying numbers of families. Figure 7.3 shows that increasing the number of families slows the overall rate of evacuation. After 85 time steps, simulations starting with 10 families had 30% of travelers remaining, whereas the simulations with 5 families had 15% remaining, and simulations with no families had only 5%. This slow down is a consequence of two main factors. First, instead of heading towards a known exit immediately upon learning

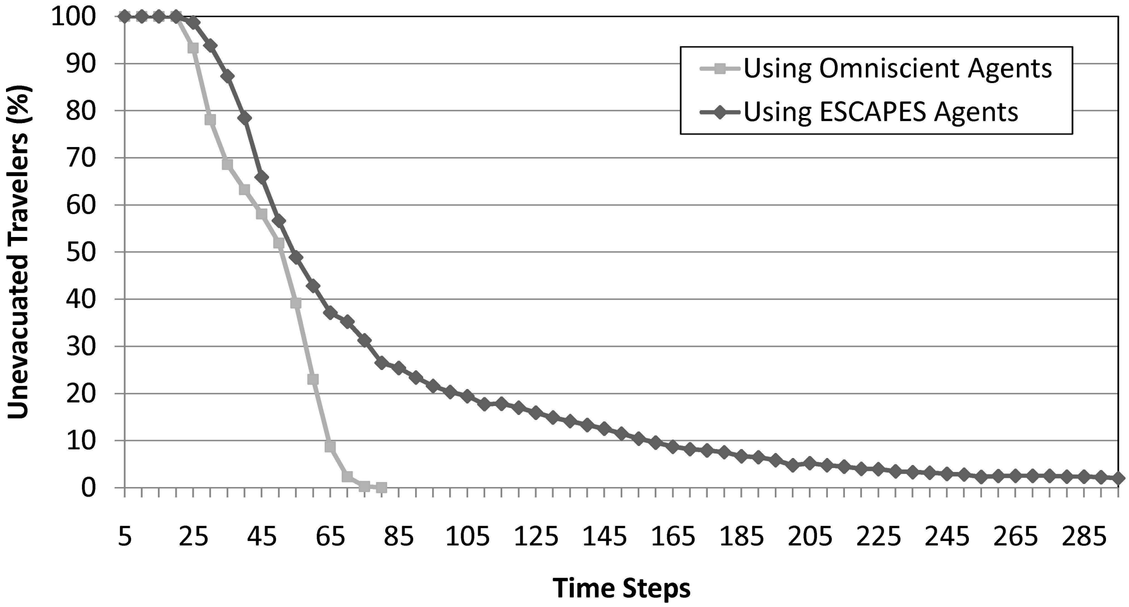


Figure 7.2: Effect of Modeling Physical, Emotional, and Informational Interactions on Evacuation Rate

of the event, parents first seek out the other members of their family. As a result, parents will often ignore known information and perform actions which are suboptimal from an individual perspective. Second, once family members have found each other, they stay grouped together. Due to children moving more slowly, as mentioned in Section 7.2.2, family units move slower than typical travelers.

7.4.3 Emotional Contagion

The spread of emotions through crowds as a result of emotional contagion has been well-documented [Hatfield et al., 1994]. In the simulations, emotional contagion is used to propagate fear. Travelers with high levels of fear pass on their FearFactor to travelers with lower levels of fear. Higher values of FearFactor activate a flight response in travelers. At the crowd level, this phenomenon causes travelers to collide into each other. The overall number of collisions can then be view be

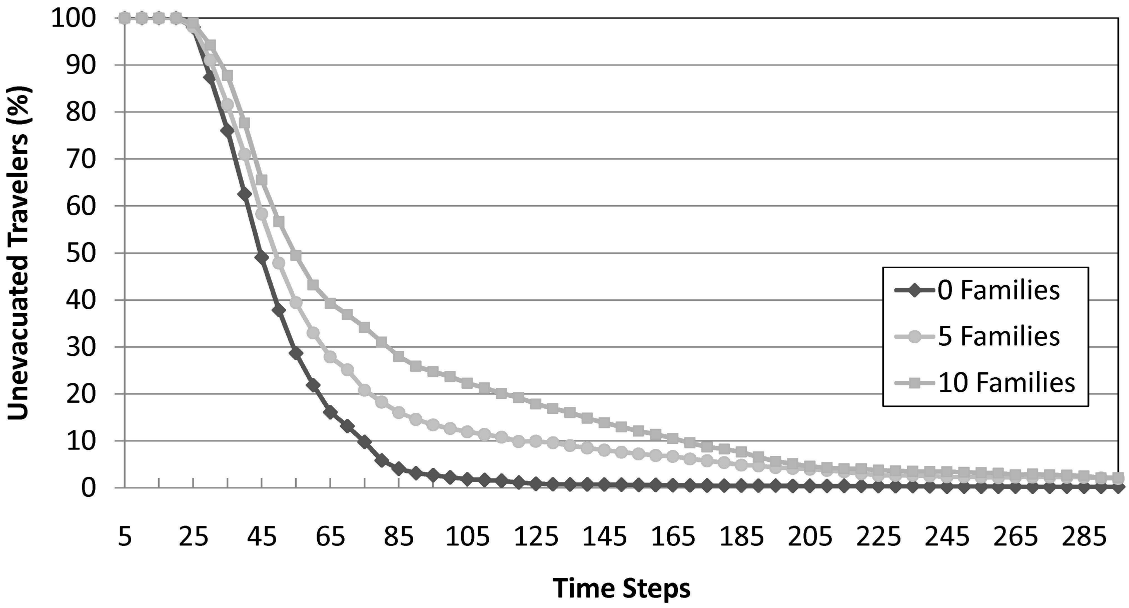


Figure 7.3: Effect of Families on Evacuation Rate

as a measure of the level of chaos in an evacuation. By modeling emotional contagion, we would expect to see an increased levels of fear which in turn will produce a higher number of collisions between travelers.

To isolate the impact of emotional contagion we ran experiments without authority figures. Without the calming influence of authority figures, there is nothing to impede the dissemination of fear through emotional contagion. Specifically, I compared the number of high-speed collisions that occurred over the course of an evacuation both with and without emotional contagion. High-speed collisions are defined as collisions that occur while a traveler has a speed of 2 or greater. Focus is placed on these collisions as they are more likely to cause injury or falls in real evacuations. When emotional contagion is modeled, evacuations average 6932 high-speed collisions, whereas evacuations without emotional contagion average 2701 high-speed collisions. From these results, we can see that modeling emotional contagion results in more chaotic evacuations with an increased number of high-speed collisions.

7.4.4 Spread of Knowledge

Agent-based evacuation simulations often start after an incident has occurred and assume that all agents are instantaneously aware of the need to evacuate. ESCAPES is geared towards domains where this is likely not the case. It is then important to model how knowledge of an event would spread throughout a crowd. In the simulations, EventCertainty represents the level of a traveler's knowledge of the event. Higher values of EventCertainty reflect greater knowledge about the event. The average EventCertainty over all unevacuated travelers is a good way to measure the level of knowledge of those who are still in danger.

In Figure 7.4, I contrast my model for the spread of knowledge against a model in which instantaneous knowledge is assumed. The y-axis represents the average EventCertainty for all unevacuated travelers, while the x-axis represents the time step. With instantaneous knowledge, travelers are able to fully perceive the event immediately after it occurs regardless of where they are situated in the environment. Accordingly, the average EventCertainty jumps from 0 (no knowledge) to 2 (full knowledge) and remains at this level for the duration of the simulation. When knowledge is spread, the situation is much different. Immediately after the event, EventCertainty is low as only the travelers close by know that is has occurred. As time passes, knowledge of the event propagates through the crowd as travelers with information disseminate it to their neighbors. As a result, EventCertainty rises until it reaches a point where almost all travelers are fully aware of the event. From this point, EventCertainty decreases as travelers with knowledge of the event are able to evacuate leaving an increasingly higher proportion of travelers who are unaware of the event.

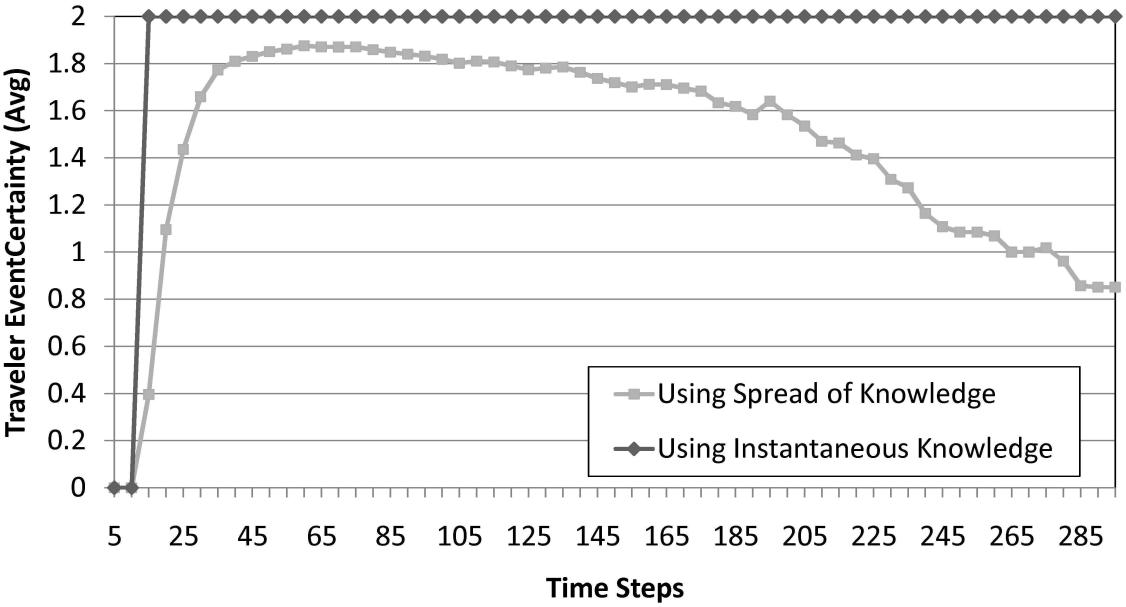


Figure 7.4: Effect of Knowledge Tranfer on EventCertainty

Throughout the evacuation, authority figures are patrolling for travelers to inform. However, if a traveler is particularly isolated they may never come into contact with an authority figure. Instantaneous knowledge is a common assumption in agent-based evacuation models, but humans are not omniscient. In comparison, my model for the spreading of knowledge provides a more realistic approximation of knowledge diffusion through crowds.

7.4.5 Authorities

Authority figures have been shown to exhibit a calming effect over crowds [Smith and Ellsworth, 1985]. In the simulations, authority figures always have a low level of fear (FearFactor=1) and the highest level of knowledge about the event (EventCertainty=2). They then help to calm the crowd by passing these values onto all travelers they come into contact with. Thus, the presence of authority figures in the simulations should result in a lower level of fear among travelers. We can use the percentage of unevacuated travelers with the highest level of fear (FearFactor=2) as an inverse measure on the ability of authority figures to calm the crowd.

Figure 7.5 shows the effect of varying the number of authority figures on the FearFactor of travelers over the course of the evacuation. The y-axis represents the percentage of unevacuated travelers with FearFactor=2. Initially, there are no travelers with FearFactor=2. At the 15th time step, the percentage increases to include all travelers close to the event. This percentage continues to climb as a result of the contagion effect until it reaches a maximum between the 35th and 50th time steps. As time progresses, the effect of emotional contagion is balanced out by the influence of authority figures and the successful evacuation of travelers with FearFactor=2. From the results, we can see that increasing the number of the authority figures results in a lower percentage of travelers with FearFactor=2. With 6 authority figures, the percentage of travelers with FearFactor=2 reaches a maximum of 47%, whereas simulations with 8 and 10 authority figures reach maximums of 36% and 27%, respectively. Given that authority figures are distributed evenly, this is a logical result, as more authority figures provide for better spacial coverage. This in turn, increases both the likelihood and speed in which authority figures will inform travelers about the event. Thus, I have shown that authority figures in the simulations display a calming effect on travelers and increasing the number of authority figures only strengthens this effect.

7.4.6 SCT

It has been observed that Social Comparison leads people in close proximity to mimic the actions of the those around them [Festinger, 1954]. In a crowd setting this would logically result in a grouping effect. The phenomenon of grouping within crowds has been well documented in research on pedestrian dynamics [Helbing et al., 2000]. To measure the prevalence of localized

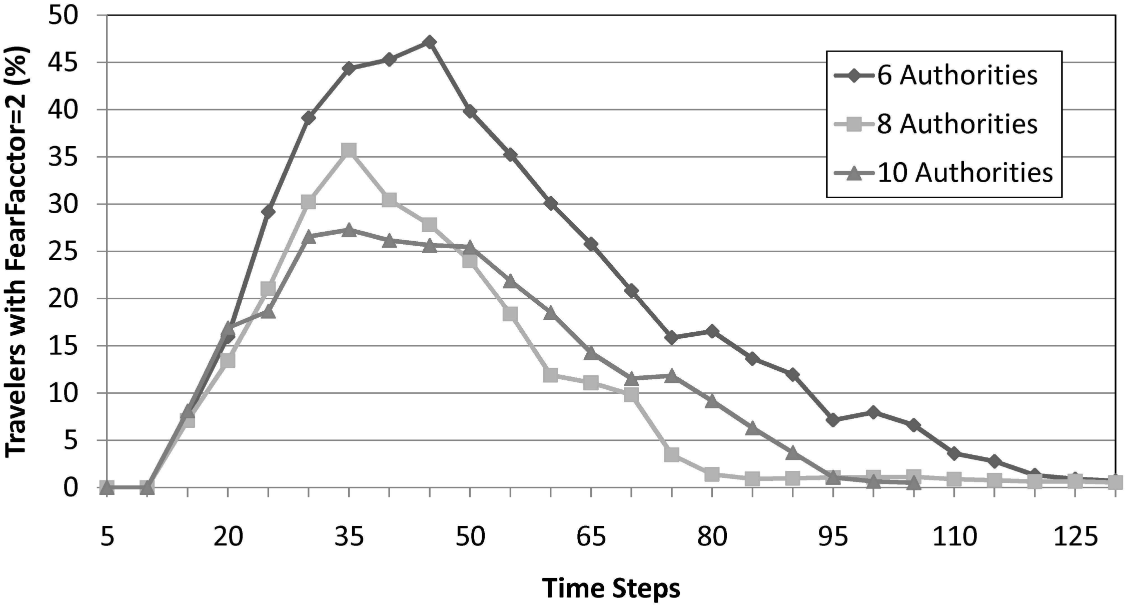


Figure 7.5: Effect of Authority Figures on FearFactor

grouping in the simulations, I introduce the notion of connectivity. A traveler's connectivity is equal to the number of neighboring travelers plus one. Travelers are considered to be neighbors if they are within a specified distance of each other. Thus, a traveler with a connectivity of 1 is considered to be isolated. As connectivity is a measure of grouping, we would expect to see an increase in the overall level of traveler connectivity by modeling Social Comparison. The impact of Social Comparison on the average connectivity of all unevacuated travelers can be seen in Figure 7.6. Connectivity, both with and without Social Comparison, rises in the moments leading up to and following the event. Without Social Comparison, the level of connectivity then steadily drops as travelers begin to disperse and exit the terminal. This continues until the average level of connectivity reaches 1, which represents travelers being isolated. With Social Comparison, the level of connectivity declines at a much slower rate before also reaching 1. These results indicate that Social Comparison increases the level of connectivity and thus the amount of grouping displayed by travelers.

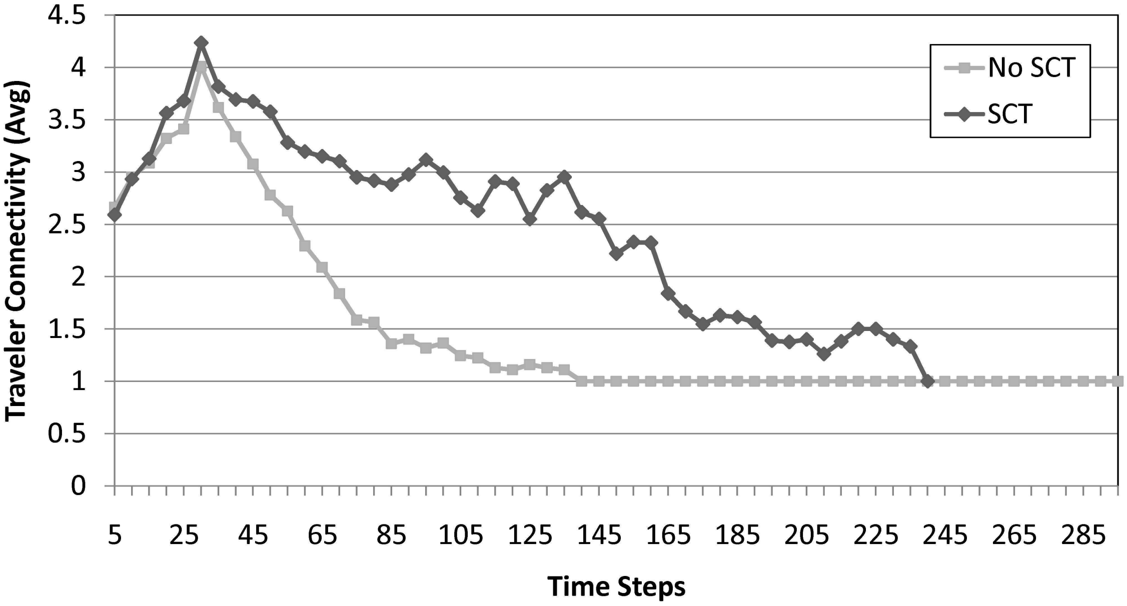


Figure 7.6: Effect of SCT on Connectivity

7.4.7 Los Angeles International Airport

Finally, I modeled the Tom Bradley International Terminal (TBIT) at Los Angeles International Airport as a realistic test scenario for ESCAPES. The scenario is approximately 55 times larger than the test case used in Section 7.4. Ideally, I would have liked to experiment on the full scenario and compare results with data from LAX, however, such data is not available. While lack of data is a major issue for most simulations in academia, the security domain presents an added level of difficulty due to confidentiality and national security concerns surrounding such data. Thus, for the tests in this section, I focused on one end of the terminal (the hallway and two gates, with one exit in each gate) and examined the impact of various authority policies with the aim of generating policy recommendations. I used 200 pedestrians, including 20 families of four, variable number of authorities, and two exits as the default case.

As a baseline test, I first ran experiments to examine the impact of increasing the number of authority figures as well as removing one exit from the scenario. We would expect that increasing

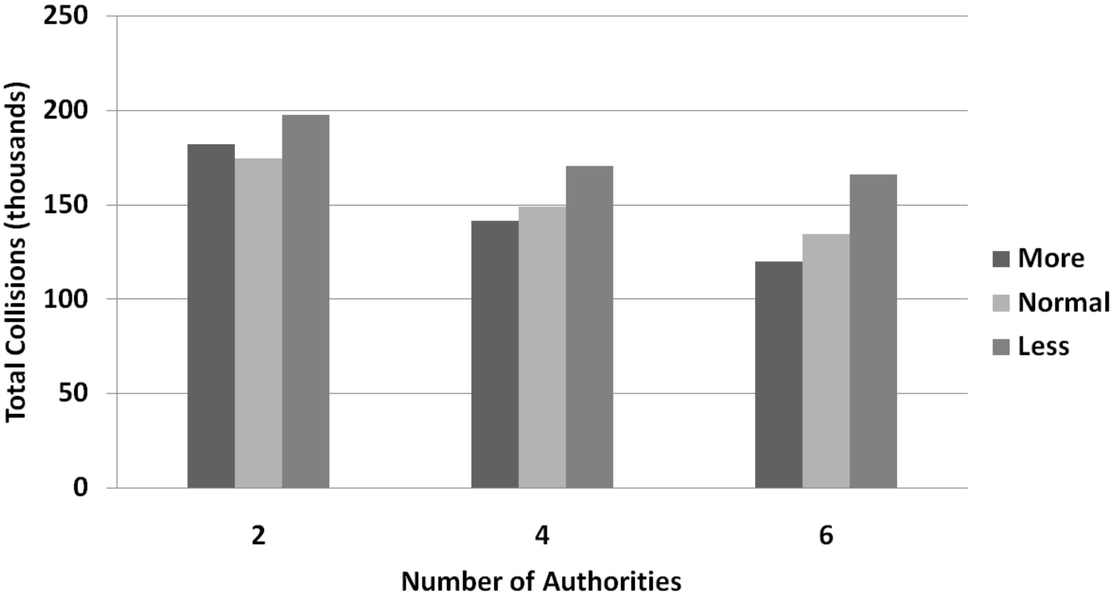


Figure 7.7: Effect of adding exits and authorities

the number of authority figures creates a calmer evacuation and removing an exit creates a more chaotic evacuation as more people squeeze towards fewer exits. Figure 7.7 shows the number of collisions (in thousands) under different parameter settings, where the number indicates the number of authorities in the setup and More/Less indicates whether an exit was added or removed from the base scenario. Higher bars indicate a more chaotic evacuation. All differences within a single authority setting, with the exception of 2-authority More vs 2-authority Normal, were statistically significant. As can be seen by the fact that the results are higher as we move to the right within a single authority setting, fewer exits lead to more chaotic evacuations. Comparing across authority settings, all differences within a single exit setting were statistically significant, with the exception of 4-authority vs 6-authority Less. As can be seen, fewer authorities leads to more chaotic evacuations as well. Both of these results are in line with expectation.

Next, as per security officials' interest, I examined the impact of having more authority figures to aid in recommending how many are needed to safely evacuate this space. Figure 7.8 shows

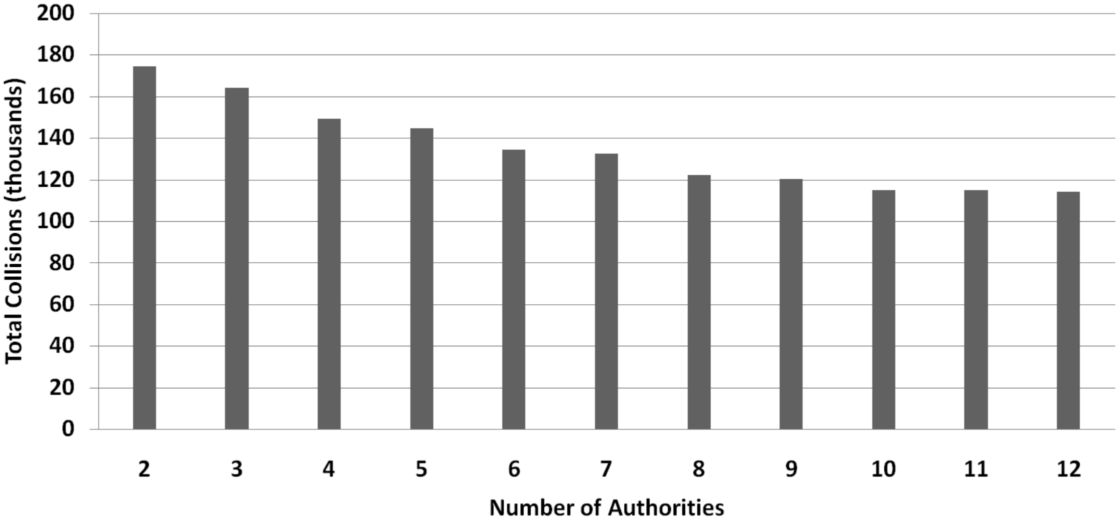


Figure 7.8: Effect of more authorities

the number of collisions over the course of the evacuation (in thousands), with the number of authorities listed on the x-axis. t-tests revealed that settings of more than 8 authority figures did not produce statistically significantly different results from the 8-authority case. This result implies that for this particular space, using more than 8 authorities would not produce better results.

I also ran tests with an alternate patrolling strategy. The default strategy is to proceed to a randomly chosen 'patrol point', the list of which is predefined to be the corners of each area in the scenario. The alternate strategy I tested was to have authority figures patrol the perimeters of the waiting areas and hallways. Results pertaining to the number of collisions were not statistically significantly different, implying no benefit to either strategy. However, further analysis revealed another trend.

Specifically, I looked at what percentage of the population would be reached by patrolling authorities on average within the first 300 time steps of the simulation. Figure 7.9 shows the percentage of people that were reached by authorities within 300 time steps. I show only the

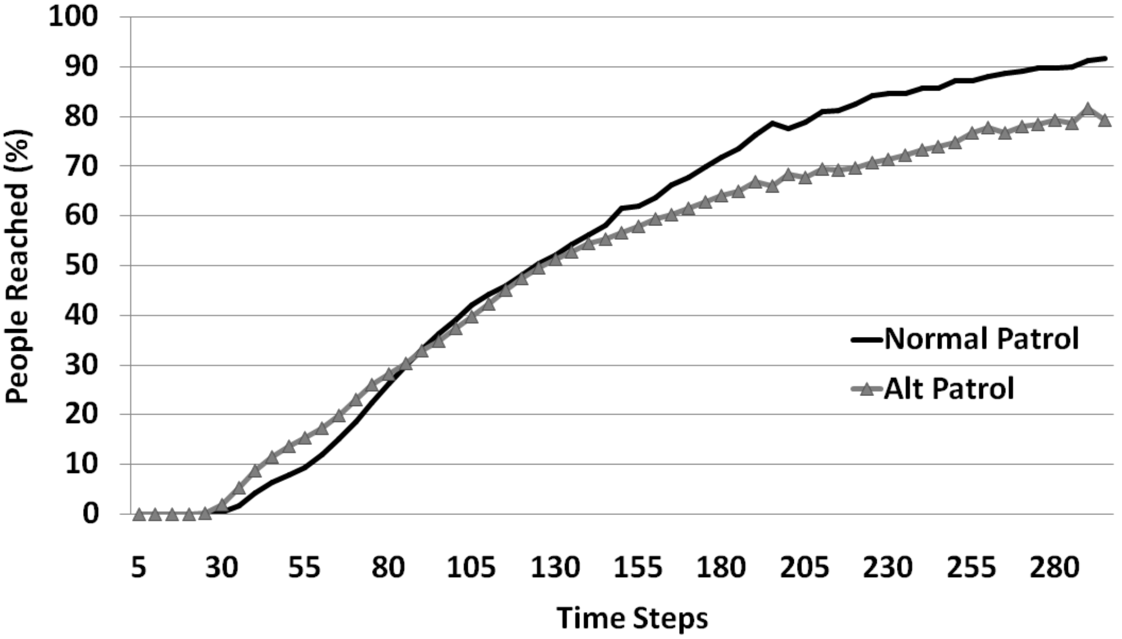


Figure 7.9: Effect of alternate patrol

case of 6 authority figures, but all like comparisons showed the same results (although varying in degree of the difference). Namely, the alternate strategy lines were always steeper at the beginning of the evacuation, but flattened out, implying that initially the alternate strategy was superior, but as fewer and fewer people remained, the point-to-point strategy was superior. Patrolling the edge of the room is effective to reach agents on the outskirts and more evenly distributes authority figures, but due to the large size of the waiting areas, crossing the room to reach different corners ultimately covers more ground. These results imply that a coordinated authority policy that intelligently covers the ground would be superior to both.

Chapter 8: Empirical Study of Emotional Contagion Models

This chapter describes my work to empirically evaluate the fidelity of computational models of emotional contagion, which is a critical step prior to any optimization of resources. Recent work in contagion simulation has sought to quantify the qualitative findings of social psychology into useable models, primarily drawing from two bodies of research on similar phenomena. Researchers at VU University introduced one of these in [Bosse et al., 2009a] (ASCRIBE) that used a deterministic, interaction-based model derived directly from a social psychology theory of emotional contagion [Barsade and Gibson, 1998]. This model is a prototypical example of the heat dissipation phenomena studied in thermodynamics wherein neighboring substances will transfer energy to each other at rates unique to each substance (i.e., specific heat) and is discussed in more detail in Section 8.1. In contrast, Durupinar (2010), presented in Section 8.2, used a probabilistic threshold model wherein successive interactions with emotionally 'infected' people raises the chance of infection with an emotion. This model is a standard one from the extensive epidemiology literature that models the spread of diseases [Dodds and Watts, 2005; Kermack and McKendrick, 1927; Murray, 2002], the research in diffusion of innovations [Rogers, 1962], and social contagion work [Schelling, 1973].

Feature	ASCRIBE	Durupinar
Emotion Level	Continuous	Binary
Fear Level Impacts Contagion	Yes	No
Emotional Decay	No	Yes
Interaction Type	Individual	Threshold
Interaction Determinism	Deterministic	Probabilistic
Proximity	Yes	No

Table 8.1: Key model differences

As outlined in Table 8.1, there are 6 primary differences between the Durupinar and ASCRIBE models. First, the Durupinar model follows in the tradition of epidemiology where it is nonsensical to discuss a ‘degree’ of infection and uses a binary specification for emotional level, whereas the ASCRIBE model uses a continuous description. Second, with no degree of emotion, the Durupinar model cannot specify differences in contagion that may result from differing levels of emotion (e.g., a high-fear agent may cause nearby agents to become more fearful than a low-fear agent would). Third, the Durupinar model includes a decay factor whereas the ASCRIBE model does not. Fourth, the ASCRIBE model uses an individual interaction model where each agent encountered causes some contagion, whereas the Durupinar model uses a threshold model in which each encounter causes an increased chance of contagion. Fifth, the ASCRIBE model uses a deterministic interaction scheme, whereas the Durupinar model uses a probabilistic one. Finally, the latest ASCRIBE model [Bosse et al., 2011] incorporates proximity’s effect on contagion, whereas the Durupinar model does not.

Using the ESCAPES evacuation simulation [Tsai et al., 2011b], the original ESCAPES model is replaced with these two models, showing substantial differences in their predictions, motivating the need for an accurate model of emotional contagion in this context. In simulation results presented in Section 8.3, I am already able to identify key differences that indicate epidemiological / social contagion models are less suited to modeling emotional contagion. Next, in Section

8.4, I attempt to reproduce a subset of 35 people from real video footage of a panic situation using each of the models, showing the ASCRIBE model to indeed be superior to both the Durupinar model and the original ESCAPES model, beating out the Durupinar model by 14% *per agent per frame* during the 15s scene. To identify which of the key features causes the differences in the results, I test hybrid models to conclude that while adding a 'decay' feature (as found in the Durupinar model) to the ASCRIBE model does not improve it, removing proximity effects and fear's graduated effect on speed substantially worsen the model. Finally, I perform the same evaluation on a second video, extracting 10 people, and show the ASCRIBE model to again be superior, outperforming the Durupinar model by 12% per agent per frame during the *four-second* scene.

8.1 ASCRIBE model

Introduced in 2009 by researchers at VU University [Bosse et al., 2009a] and built upon in multiple works including [Bosse et al., 2009b, 2011], the ASCRIBE model iterates through all agents and deterministically calculates new emotional levels based on a set of individual and pairwise parameters that I describe here. The mechanism used resembles heat dissipation modeling in physics, wherein each material has a specific heat capacity, which can be likened to a person's susceptibility to other people's emotions in emotional contagion. As such, the model moves a crowd towards a weighted-average of the group's emotional levels, just as heat will dissipate until adjacent temperatures are the same, barring generative heat sources.

The model defines 5 parameters (shown in Table 8.2) for every pairwise interaction based on theory put forth in [Barsade and Gibson, 1998]: level of sender's emotion q_S, level of receiver's

level of the sender's emotion	q_S
level of the receiver's emotion	q_R
sender's emotion expression	ϵ_S
openness for received emotion	δ_R
strength of the channel from sender to receiver	α_{SR}

Table 8.2: Aspects related to a sender *S*, receiver *R*, or both

emotion q_R, sender's expressiveness ϵ_S, receiver's openness δ_R, and the channel strength between S and R α_{SR}. All values are numbers in the interval $[0, 1]$. At each time step, each agent calculates the average emotional transfer from all relevant agents. Specifically, the differential equations for emotional contagion in a group G of agents is:

$$dq_R/dt = \gamma_R(q_R^* - q_R)$$

for all $R \in G$, where γ_R is the overall strength at which emotions from all other group members are received, defined by $\gamma_R = \Sigma_{S \in G\setminus\{R\}}\gamma_{SR}$. q_R^* is the weighted combination of emotions from the other agents, defined with a weight factor:

$$w_{SR} = \epsilon_S \alpha_{SR} / \Sigma_{C \in G\setminus\{R\}} \epsilon_C \alpha_{CR}$$

$$q_R^* = \Sigma_{S \in G\setminus\{R\}} w_{SR} q_S$$

Specifically, from a sender S to a receiver R, the strength of the emotion q_S received would be $\gamma_{SR} = \epsilon_S \cdot \alpha_{SR} \cdot \delta_R$. [Bosse et al., 2009a] details the mathematical formulation, but the emotional level of an agent converges towards a weighted average of the group's emotional level. The speed at which this convergence occurs as well as the weighting depend on the parameter settings for the channel strength, expressiveness, and openness for each agent as well as, of course, their individual emotional levels.

The latest version of the model [Bosse et al., 2011], extends the original emotional contagion model and includes beliefs and intentions and belief/intention contagion as well. However, as my goal is to empirically evaluate emotional contagion models and the latest work extends far beyond simply emotional contagion, I leave its validation to future work. Thus, I do not use the extended model but instead modify the initial model by incorporating a proximity effect as done in [Bosse et al., 2011].

8.2 Durupinar Model

Durupinar [Durupinar, 2010] uses a probabilistic threshold model based on epidemiological models of disease contagion. While many types of epidemiological models exist [Dodds and Watts, 2005; Kermack and McKendrick, 1927; Murray, 2002; Schelling, 1973], Durupinar opts for a baseline model from [Dodds and Watts, 2005]. In this model, individuals can be in either *susceptible* or *infected* states. Other models incorporate additional states such as inoculated, recovered, etc. which could be incorporated in extensions to Durupinar's basic model, but have not been explored in the context of emotional contagion. The epidemiological model's applicability to emotional contagion was not discussed in [Dodds and Watts, 2005], from which Durupinar drew, but its use by Durupinar assumes similarity between disease spread and emotion spread that we criticize in this work.

Each agent begins with a randomized threshold drawn from a pre-determined log-normal distribution. At each time step, T, for each agent, a random agent is chosen from the relevant population group. If the agent is infected, it generates a random dose, d_j, with size drawn from a pre-determined log-normal distribution and passes it to the original agent. If the agent is not infected, then a dose of 0.0 is generated. Each agent maintains a running history of the last K doses received. If the cumulative total of all doses in the agent's history exceeds his threshold, the agent enters the infected state (Equation (1)). This causes the emotion level to be set to 1.0 with an exponential decay towards 0.0 at a rate β (Equation (4)), at which point the agent re-enters the susceptible state. A non-zero emotion level indicates that the agent has the emotion, but the actual value does not hold meaning other than to track the decay. The random dose and threshold are generated from log-normal distributions (Equations (2),(3)) with user-specified averages and standard deviations and K is a static global variable.

$$D_j(t) = \sum_{t'=t-K+1}^{t} d_i(t') \tag{8.1}$$

$$d_j = \text{log-}\mathcal{N}(\mu_{d_j}, \sigma^2_{d_j}) \tag{8.2}$$

$$T_j = \text{log-}\mathcal{N}(\mu_{T_j}, \sigma^2_{T_j}) \tag{8.3}$$

$$e_t = e_{t-1} - \beta \cdot e_{t-1} \tag{8.4}$$

Durupinar also provides a psychological basis for setting the dose and threshold distribution values by incorporating findings from Jolliffe and Farrington (2006) on the correlation between the basic empathy scale and the OCEAN personality factors. A much richer emotional model is also described, but for the purposes of this study, we only use the emotional contagion model. The particular model introduced here is but one example from the range of similar contagion models [Kermack and McKendrick, 1927; Murray, 2002; Schelling, 1973], but they all share a binary, probabilistic treatment of effect. While it may seem trivial to interpret the decaying emotional indicator as a continuous variable, this alteration proves unhelpful in our experiments. As we show in the following sections, this fundamental difference between the heat dissipation-style

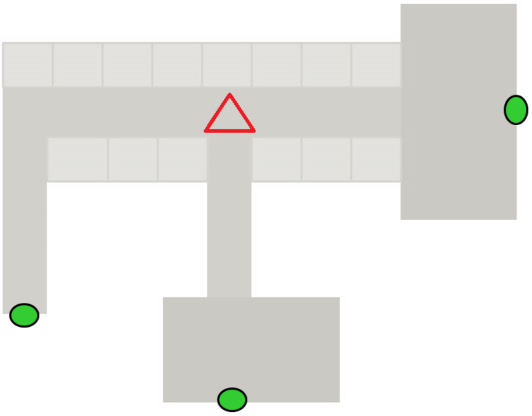


Figure 8.1: Evacuation scenario.

models (such as the ASCRIBE model) and epidemiological models leads to inaccuracies in the Durupinar's modeling of emotional contagion.

8.3 Simulation Experiments

Although they are similar from a computational performance perspective, the ASCRIBE model and the Durupinar model use very different mechanisms to recreate emotional contagion. Thus, I evaluate the impact of these differences in two ways, beginning first with simulation. I ran the evacuation simulation, ESCAPES, using each model to perform sensitivity analysis as well as identify any qualitative trends that might support or discredit either one of the models. We can also evaluate the model's robustness to errors in parameter estimation, which is extremely important in emotional and crowd modeling which usually lack high fidelity, fine-grain data.

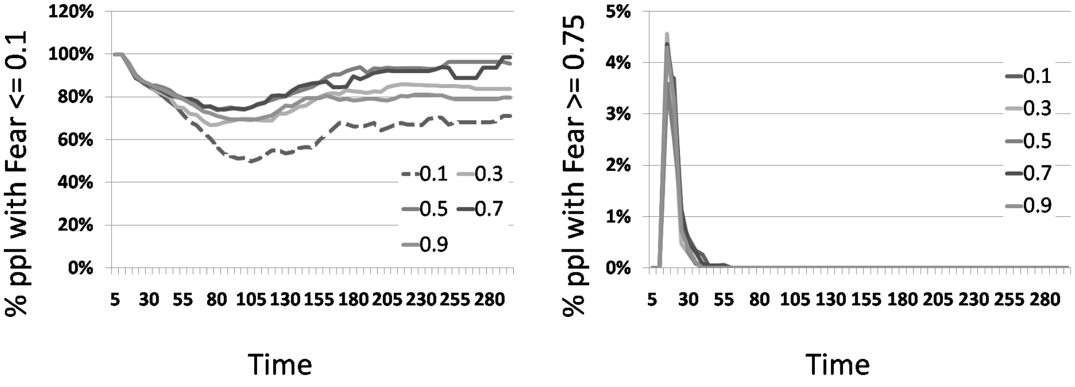
For all the experiments discussed in this section, the same map was used (spatial layout can be seen in Figure 8.1) and 30 trials were run for each setting. It features 2 large spaces that represent airport boarding areas, each with an exit (marked with dots), connected by hallways which are lined with smaller spaces that represent shops. 15 seconds into the simulation, an event occurs at the center of the scenario (marked by the triangle), inciting fear and a need to evacuate that is communicated by authority figures to pedestrians. For initial fear levels, we define a 'seeing distance', σ_d. Agents within this distance of an event will immediately have a fear level of 0.75 in the ASCRIBE model and 1.0 in the Durupinar model, since the Durupinar model does not feature a continuous measure of fear. I also define a 'hearing distance', ω_d, within which the agent will receive 0.1 in the ASCRIBE model and 1.0 in the Durupinar model. The scenario features 100 normal pedestrians, including 10 families of 4 each, as well as 10 authority figures that patrol the scenario. In Sections 8.3.1 and 8.3.2 we evaluate model robustness and then identify qualitative differences in Section 8.3.3.

8.3.1 ASCRIBE model

In examining the contagion effect, the parameters of interest in the ASCRIBE model were the individual expressiveness settings and individual openness settings. The channel strength is set to 1 if an agent is nearby and 0 otherwise, as done in [Bosse et al., 2011]. Given that there was a whole population of agents, I elected to use randomly drawn values for expressiveness and openness based on a normal distribution. It may be useful to relax this assumption of a normal distribution in future work, perhaps when more quantitative analysis of human contagion parameters becomes available. I explored variations of the averages and standard deviations (SD) used, but surprisingly, none yielded qualitative changes in the trends observed in the simulation from both a contagion perspective (i.e., how the fear spread) and a safety analysis. The only exceptions were, unsurprisingly, when the receiver openness or sender expressiveness parameters varied tightly around a very low mean, often leaving many agents with 0.0 openness or expressiveness. Very low receiver openness values created perpetual high-fear sources that constantly raised the fear levels of surrounding agents without ever dissipating their own fear. At very low sender expressiveness values, the majority of agents remain at their initial fear level. Both cases result in vastly different trends from the mean convergence behavior seen in the other settings.

To illustrate the contagion effect of variations in the parameter settings, Figure 8.2a plots the percentage of people with low fear (≤ 0.1) on the y-axis and the time step on the x-axis, while Figure 8.2b shows the same results for high-fear people. In both figures, openness varied from 0.1 to 0.9 in increments of 0.2 while keeping a SD of 0.1 and sender expressiveness was fixed with an average of 0.5 with a SD of 0.1. In Figure 8.2a, when an event first occurs, those near it become fearful and slowly raise nearby peoples' fear as they move towards exits, causing a steady decline in the percentage of people with fear less than 0.1 that only rises again as fearful agents make their way out of the simulation. Note how the dotted line (0.1) dips much lower than the other lines, showing the exception mentioned above. In Figure 8.2b, a few agents near the event have their fear raised very high, but as they encounter zero-fear agents, their fear levels are brought down below 0.75 and never again rise higher since no new events occur. The tightness of the lines implies that the trend is robust to variations in the average receiver openness except at very low settings. Similar tightness of lines was observed in variations of sender expressiveness, with the same exception.

Figure 8.3 focuses only on variations of openness to illustrate the trends observed in evacuation safety. Figure 8.3a plots the percentage of people that have not escaped on the y-axis and time steps on the x-axis, whereas Figure 8.3b shows the average number of agent-agent collisions accumulated by each person remaining in the simulation on the y-axis and time steps again



(a) Percent low-fear agents

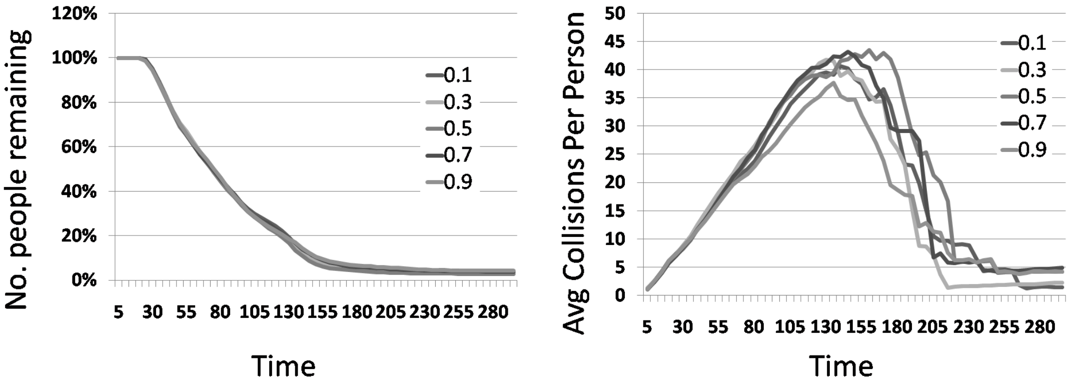
(b) Percent high-fear agents

Figure 8.2: ASCRIBE model: Variations of Openness on contagion

on the x-axis. Since the simulator does not explicitly handle agents colliding, we define 'collisions' as anytime two agents touch each other. Figure 8.3a shows all parameter settings for openness leading to almost identical escape times for people during the simulation. Figure 8.3b shows extremely similar collision counts for people across the parameter space as well. Thus, when measuring the second-order metrics of pedestrian escape times and number of collisions, the model remains robust to parameter variations of the type tested. Variations of the other parameters' averages and standard deviations all resulted in the same extremely tightly clustered lines as seen in Figures 8.2 and 8.3 (with the previously noted exception).

8.3.2 Durupinar Model

Sensitivity analysis of the Durupinar model is considerably more delicate than the ASCRIBE model, because although there are only 5 key parameters for the whole population (as compared to 2 per individual plus 1 for each pair for the ASCRIBE model) and even a small change in a single parameter can result in qualitatively different trends as we will show. Thus, I begin with experimentally chosen default values and vary each parameter to identify key sensitivities. In



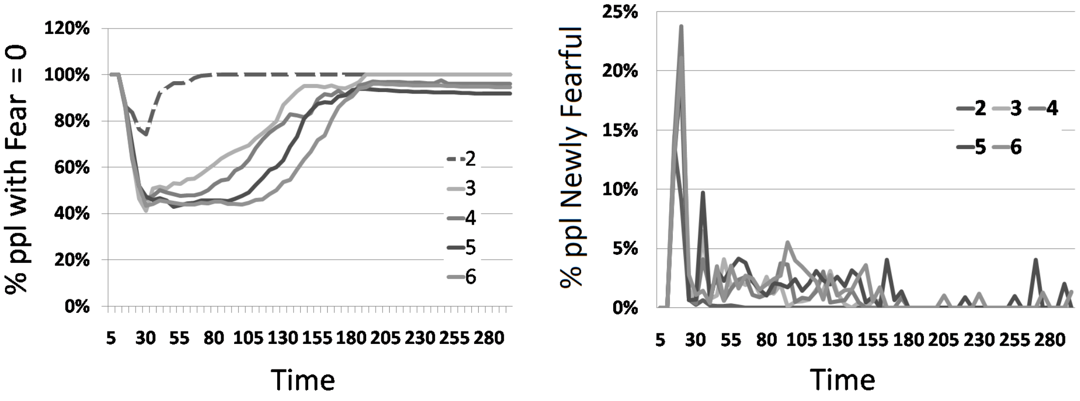
(a) Percent pedestrians unevacuated

(b) Average collisions per person

Figure 8.3: ASCRIBE model: Variations of openness on safety

particular, I begin with a baseline of K of 4, dose average of 2, dose standard deviation of 0.5, threshold average of 7, and threshold standard deviation of 2.

Figure 8.4a shows the percentage of no-fear pedestrians ($= 0$) on the y-axis and time steps on the x-axis, with each line representing a different setting of K. Figure 8.4b shows the percentage of newly fearful pedestrians (defined as ≥ 0.75) during the same variations of K. Unsurprisingly, altering any one of the parameters' averages *or* standard deviations individually alters the magnitude of the contagion effect, but not the overall trends. The exceptions are at values far from the baseline. For example, at extremely low values for K or dose distribution average and at extremely high values for threshold distribution average, when very few agents become fearful at all, as seen in the dotted $K = 2$ line in Figure 8.4a. This implies that the model remains robust to parameter changes with respect to the contagion trends that emerge as long as parameter values are chosen within a tolerance of the baseline. Similar results were found for variations of threshold and dose strength averages and standard deviations.



(a) Percent no-fear pedestrians

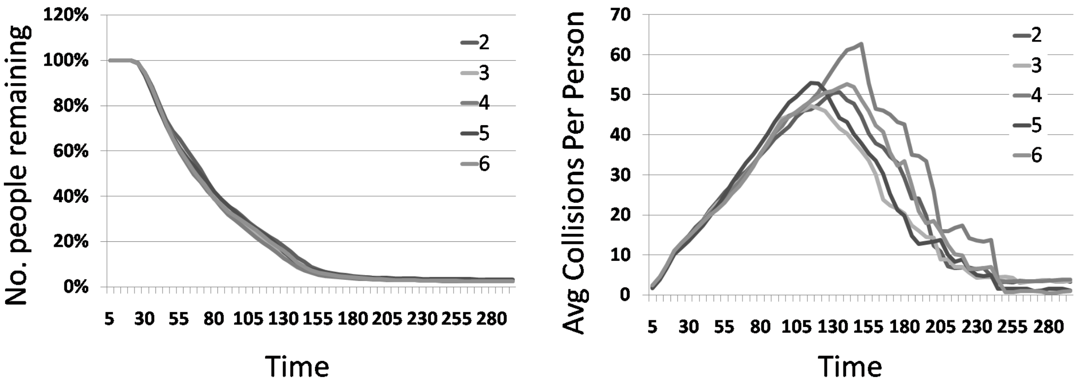
(b) Percent new-fear pedestrians

Figure 8.4: Durupinar model: Variations of K on contagion

I again explored the second-order impacts of parameter variations on the safety of the evacuation by measuring the evacuation rates and average number of collisions of pedestrians in the simulation. Figure 8.5a shows the percentage of people that have not yet evacuated on the y-axis and the time steps on the x-axis, while Figure 8.5b shows the average number of agent-agent collisions accumulated for each person remaining in the simulation on the y-axis and the time steps on the x-axis. As in the ASCRIBE results shown previously, both of these graphs show extremely similar results across the parameter space tested. Variations of other parameters showed very similar results.

8.3.3 Key Differences

In Sections 8.3.1 and 8.3.2 I have shown the ASCRIBE model to be robust to parameter variations (except at the extreme of zero) and the Durupinar model to be robust if we stay within a tolerance of a baseline. In conducting these simulation tests and taking a closer look at the contagion effect, I already find that a number of key differences can be identified between the two models. One difference can be seen by comparing Figures 8.2b and 8.4b, where the spikes occurring throughout



(a) Percent pedestrians unevacuated

(b) Average collisions per remaining person

Figure 8.5: Durupinar model: Variations of K on safety

the graph indicate that Durupinar model produces newly fearful agents throughout the life of the simulation, *regardless of the nature of the event*, and the ASCRIBE model only exhibits a spike due to the impact of the event. Under the Durupinar model, fear can be transferred indefinitely under certain parameter settings. In the ASCRIBE model, encounters with agents who are less fearful will slowly erode the average fear level, eventually reaching zero after sufficiently many agents have been encountered.

Also, combining the binary fear metric with a speed modifier, as done in ESCAPES, results in only extremes of movement speeds. While one could argue that this is a result of the simulation, the model itself cannot incorporate any gradation of effect. For example, even if we directly map the fear level (as it decays) to the speed modifier, an agent that is near zero-fear (and is hence traveling slowly) can infect another agent who will then dart off at maximal speed since he begins at maximal fear, as evidenced by the spikes in Figure 8.4b. This may occur as a result of physiological or informational changes, but no evidence suggests this would occur from emotional contagion alone. A more fundamental alteration is needed to change this aspect of epidemiological / social contagion models for convincing application to emotional contagion.

Finally, as mentioned, the Durupinar model does not include a proximity of effect, whereas the ASCRIBE model does. This obviously means that the Durupinar model could potentially cause contagion of emotions to agents randomly throughout the world of the simulation, a very unrealistic effect, as emotional contagion requires *some* form of interaction by definition. As seen in a comparison between Figures 8.2a and 8.4a, the Durupinar model induces more fearful agents far more rapidly than the ASCRIBE model does because its contagion calculation incorporates the entire population immediately.

8.4 Scene Reproduction

Now I discuss the validation method used to evaluate the models of emotional contagion, first used in [Bosse et al., 2011]. In their work, VU University researchers used a 15-second portion of a crowd panic scene in Amsterdam caused by a screaming person[1] as their dataset for validating their general mental state contagion model. In processing the data, the researchers traced the locations of 35 people scattered through the crowd through the 15 seconds, converted these into top-down coordinates and built a simulator to reproduce the paths of the people in simulation (for more detail on the spatial parametrization, please refer to [Bosse et al., 2011]). The 35 people chosen can be interpreted as point estimations of the speed and trajectory of subsets of the crowd throughout the scene. Thus, accurately predicting the movement of these individuals would translate into accurate prediction of the overall movement of the crowd. Furthermore,

[1] `http://youtu.be/0cEQp8OQj2Y`

since realistic agent collisions remains an open research question, using every individual in the crowd would present novel challenges beyond the scope of this work.

The operating hypothesis was that a simulator without their mental state contagion model would not be able to reproduce the scene as accurately as a simulator with it. To test this hypothesis, the researchers tuned parameters associated with each agent's maximum speed, a global parameter specifying a 'sight range' within which agents could 'see' the event, and an initial desire to remain in place. The tuning was done via hill-climbing to minimize the error produced by the simulator, testing each parameter and moving a single parameter at a time in the direction of highest error reduction until a local optimum was reached. Error was defined as the sum of the average distances from each simulated agent to the corresponding real people's locations over the life of the simulation. Errors are reported in pixel-distances, measured in the converted top-down coordinate system. Real-world distance equivalents are discussed at the end of each section. Finally, VU University researchers incorporated the mental state contagion model, tuning a parameter associated with the proximity of contagion and showed that lower error was achieved with this addition.

I extend the approach of the VU University researchers by importing the 35 agent traces into the ESCAPES simulator and setting 3 exit locations towards which agents proceed when the simulation starts. The locations were chosen to roughly mimic the real situation, leading to most agents moving in the same direction as the people did. Some agents did not move precisely in the simulated direction as a result of obstructions that I did not model and a person very close to the screaming person that barely moved. The primary task was to match the crowd's location over time, first without contagion effects and then with each contagion model in turn. Since people's directions did not vary based on the emotion, the contagion model could only impact the speed of each agent.

The speed of an agent, without incorporating contagion effects, is based on the emotional level multiplied by the maximum speed multiplied by a distance-based modifier. The distance-based modifier is σ_s if the agent is within sight range and ω_s if the agent is only within hearing distance. I include these tunable speed modifiers so that the simulation is robust to the choice of initial fear levels, which is particularly helpful given the lack of data surrounding how to set the initial fear levels.

As an example of the speed calculation, under the ASCRIBE model, if an event occurred within hearing distance but not seeing distance of an agent, and the hear-range speed modifier was 0.2, the agent's speed would be $(0.1)(0.2)(S_{max})$, where S_{max} is the maximum speed allowed in the simulation. Under the Durupinar model, the 0.1 would be replaced with a 1.0. σ_d, ω_d, σ_s, and ω_s are global parameters applied to all agents that I tune experimentally via the same methodology as used in [Bosse et al., 2011], where my measure of error is the sum of simulation-space coordinate distance between the simulated agents and the actual agents.

For each contagion model, I use the default settings discussed in Section 8.3, with the exception of the ASCRIBE model's channel strength, which I set to 1.0 or 0.0 depending on the proximity of other agents, as was done in [Bosse et al., 2011]. In the ASCRIBE model, we follow [Bosse et al., 2011] and fix Receiver Openness and Sender Expressiveness each to 0.5 *for every agent*, but allow the proximity parameter to be tuned. In the Durupinar model, I set the dose history to 6, the mean and standard deviation of the dose strength distribution to 2 and 0.5, and the mean and standard deviation of the threshold distribution to 7 and 2. The ESCAPES contagion model, used as a baseline for comparison, only requires tuning of the proximity parameter as it simply brings all agents to the highest level of fear found in surrounding agents. In an attempt to not only identify which model is more appropriate but also to discern key features from unsupported augmentations, I used each model as given, then turned on/off implementations of ‘decay’, emotional level impacting speed, and proximity effects. For each parameter setting, 30 trials were run.

8.4.1 Amsterdam Crowd

I first use the Amsterdam crowd scene featured in [Bosse et al., 2011]. In their results, VU University researchers found that the inclusion of contagion effects achieved significantly less error in reproducing the movement of a selection of 35 agents from the crowd scene. Upon closer inspection, the data revealed that the subset of agents within a particular radius surrounding the event caused the majority of every model’s error. Specifically, approximately 80% of the error in each of the models’ results can be attributed to the 13 agents nearest the explosion. The distinction between ‘near’ and ‘far’ agents is an empirical categorization based on the error attribution mentioned.

I show the error breakdown in Figure 8.6. Three categories of error are shown: faraway agents, the agent closest to the yelling, and the other nearest agents excluding the closest agent. The agent closest to the yelling barely moved in the video, which is a situation that the cognitive model of ESCAPES does not naturally simulate. Hence, all models produce large errors quite unrelated to the underlying emotional contagion model. The faraway agents, by contrast, move extremely little, making it easy to fit any model to them by simply forcing those agents to remain completely still. Thus, the largest portion of the error, that caused by the agents near the event

(a) Base models

Model	Overall	Near
None	0.375	0.699
ESCAPES	0.375	0.698
ASCRIBE	0.362	0.663
Durupinar	0.383	0.758

(b) ESCAPES variations

Model	Overall	Near
Base	0.375	0.698
Decay	0.379	0.703
No Speed	0.381	0.721
No Prox	0.385	0.721

(c) ASCRIBE variations

Variation	Overall	Near
Base	0.362	0.663
Decay	0.363	0.687
No Speed	0.387	0.767
No Prox	0.414	0.797

(d) Durupinar variations

Model	Overall	Near
Base	0.383	0.758
No Decay	0.387	0.771
Speed	0.388	0.784
Prox	0.380	0.754

Table 8.3: Amsterdam crowd: Average error (in pixels) per agent per frame

(except the closest agent) also provides the most potential for the emotional contagion models to differ.

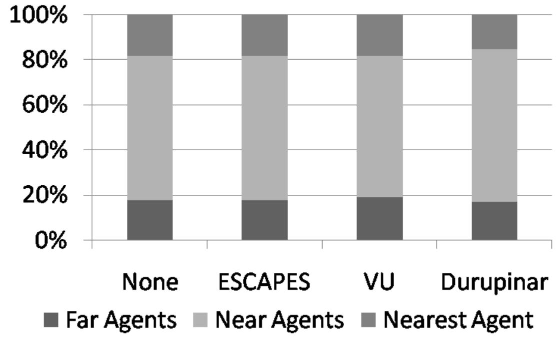


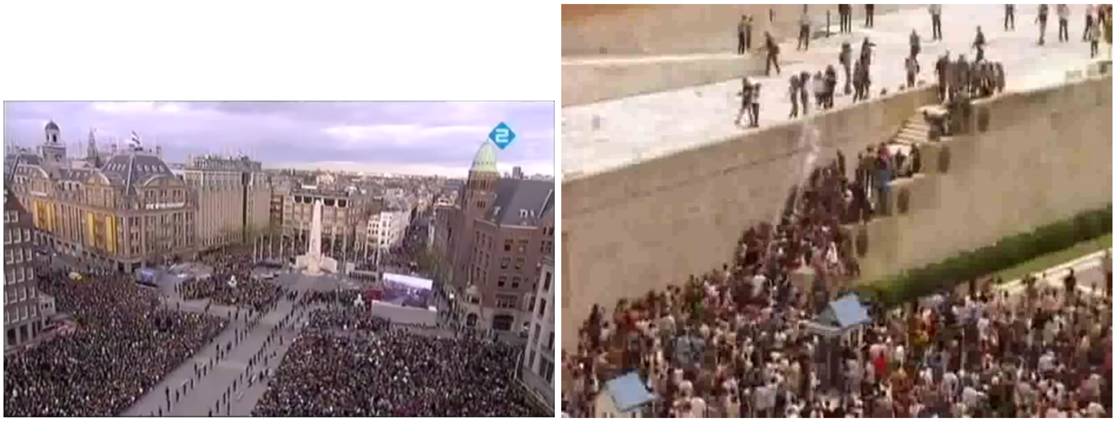
Figure 8.6: Amsterdam crowd (35 agents): Error attribution

The results from the different variations of each model is listed in Table 8.3. Table 8.3a shows the results for the base models as defined previously, illustrating OVERALL error (for all 35 agents) as well as the error associated with the most substantial group of agents, the 12 NEAR the event, excepting the closest agent. Table 8.3b shows the variations associated with the original ESCAPES formulation. The second line of the table indicates that a ‘decay’ feature was added to the base model. The third line indicates that I turned on/off the effect that different levels of fear have on speed. When off, this means that any level of fear causes agents to travel at maximum speed. When on, the speed of travel is proportional to the fear level. Finally, the fourth row represents whether the contagion effect was moderated with a tuned proximity effect. Tables 8.3c and 8.3d show the analogous set of variations for the ASCRIBE and Durupinar models.

No results from the ESCAPES contagion formulation were statistically significantly better than the No Contagion case. This, as well as all remaining statistical tests in this work, was measured with a one-tailed t-test. This indicates that the ESCAPES contagion model does not add anything in the context of this dataset. In sharp contrast, all results for ASCRIBE and Durupinar were statistically significantly different from the No Contagion case, although in the case of Durupinar, they were significantly *worse* ($p < 0.001$). As found in [Bosse et al., 2011], the ASCRIBE model's formulation provided substantial improvements in the simulation's ability to reproduce this scene (14% superior to Durupinar for NEAR agents in the Base cases for the 15s clip).

For ESCAPES, no feature change offered statistically significantly different results from the base case, implying that in this formulation, for this data set, adding 'decay' did not help and the presence of 'speed' and 'proximity' features did not add value to the model either. In the ASCRIBE model, adding 'decay', removing 'speed', and removing 'proximity' all had statistically significantly negative impacts on the results ($p < 0.001$). This implies that the 'speed' and 'proximity' features were crucial to generating the positive result in the Base case and adding 'decay' does not improve it. Finally, removing 'decay' produced significantly worse results in the Durupinar model, and the other two variations did not produce statistically different results.

These results imply that the ASCRIBE model's contagion mechanism and current formulation provides the highest fidelity in modeling this dataset versus other variations and models tested. To properly frame the magnitude of improvement, consider a crowd being modeled for five minutes. In real terms, the 14% average difference between ASCRIBE and Durupinar amounts to over *two meters* of error over the 12 NEAR agents in a *single frame*. ‘Small’ errors like this in the first 15s can easily snowball into a completely different crowd structure after five minutes, suggesting much larger implications to this 14% improvement.



(a) Amsterdam video

(b) Greece video

Figure 8.7: Amsterdam and Greece video screenshots

8.4.2 Greece Crowd

Since one dataset could be particularly well-suited to the ASCRIBE model, I elected to perform the same process on a second video from protests in Greece in 2010[2], where officers fired tear gas into the middle of a small crowd. The clip used was from 0:16 to 0:20, from which 24 frames were extracted for analysis. 10 figures throughout the crowd were traced for the duration of the clip. Conversion of the pixel coordinates into top-down coordinates was done by first estimating true axes in the top-down view by tracing the sidewalk and steps that were perpendicular to the

[2] `http://www.youtube.com/watch?v=NsoDwM_KKfo`, posted May 5, 2010

(a) Base models

Model	Error
None	1.635
ESCAPES	1.478
ASCRIBE	1.478
Durupinar	1.656

(b) ESCAPES

Model	Error
Base	1.478
Decay	1.474
No Speed	1.567
No Prox	1.658

(c) ASCRIBE

Variation	Error
Base	1.478
Decay	1.466
No Speed	1.653
No Prox	1.660

(d) Durupinar

Model	Error
Base	1.656
No Decay	1.653
Speed	1.669
Prox	1.654

Table 8.4: Greece crowd (10 agents): Average error (in pixels) per agent during the simulation

sidewalk. Then, the distance to each of the axes was calculated (where 'distance' is measured from the point to the axis, parallel to the other axis) and used as the new coordinates.

Even in such a short video clip with such a small crowd that we are able to match extremely well, the emotional contagion models still showed significant differences. Surprisingly, the original ESCAPES model performs extremely well, matching the ASCRIBE model's accuracy. However, as before, we see the Durupinar model again performing substantially worse than all other models, implying some generality of the previous result. In fact, this scene is an even stronger testament than the previous one, as the ASCRIBE model performs 12% better than Durupinar in the Base case per agent per frame during only a *four-second* clip as opposed to the 15s Amsterdam clip. For both the original ESCAPES model and the ASCRIBE model, removing fear's impact on speed and the proximity effect statistically significantly worsen's the model's accuracy ($p < 0.001$). Surprisingly, the ASCRIBE model benefits from the addition of a decay component ($p < 0.001$), implying that a decay effect may be context-dependent.

Chapter 9: Emotional Contagion with Virtual Characters

This chapter is the final one that outlines my contributions to contagion studies and describes a series of experiments conducted to examine the contagion effect of virtual character emotions with human users. As the use of simulation for personnel training becomes more prevalent, contagion in these tools must be understood to properly recreate the real-world scenarios these people are being trained for. Only then can people who train on these simulations be effectively deployed by the game-theoretic techniques I previously discussed.

The vast majority of emotional contagion research has come from the social sciences and examines the spread of emotions from humans to other humans. Emotional contagion's impact in virtual agents' interactions with humans is a largely untouched area of research. The effects are assumed to either be nonexistent and therefore overlooked entirely or to mimic human-human emotional influences. However, these assumptions are not supported by my experiments. As virtual agents enter high-risk and emotionally delicate applications such as virtual psychotherapy [Riva, 2005; Rizzo et al., 2005; Rothbaum et al., 2001], for example, researchers must be cognizant of all potential emotional influences characters can have on users.

This work serves as a first study to find experimental support for the aforementioned results in agent-human emotional contagion. Pursuant of this goal, three sets of studies are conducted.

The first study examines the pure contagion case by simply showing subjects a still image of a virtual character with either a happy expression or a neutral expression and then assessing the subject's self-reported happiness thereafter. The use of a still image as a manipulation follows from previous studies in emotional contagion [Small and Verrochi, 2009; Wild et al., 2001]. The second study adds the presentation of a game-theoretic situation known as a Stag Hunt along with the character image to assess both the contagion and the behavioral impact of the virtual character in a strategic setting. While studies have shown that emotional contagion can impact one's propensity to trust and enhance perceived cooperation among other findings [Barsade, 2002; Dunn and Schweitzer, 2005], there has been far less work showing behavioral impacts in strategic situations. Thus, I also attempt to examine whether behavioral impacts arise in strategic situations to better understand its potential impacts in real-world agent applications. Finally, the third study examines the post-hoc hypothesis that the presentation of a decision to the user dampens the emotional contagion effect. Specifically, I present the same strategic situation as in the second study, but with the decision already made for the subject.

In this work, I provide the first experimental results supporting the existence of emotional contagion between virtual agents and humans. Results show a very large increase in self-reported happiness from only adding a smile to an otherwise identical still image of a virtual character. In the second study, when the character is placed in the context of a strategic decision, both subject behavior and subject self-reports of happiness are only impacted significantly by one character. The last study, which removes the user's decision from the previous experiment, finds that the character's expression's affect on emotion returns significantly, implying that a strategic decision posed to users will dampen the emotional contagion effect beyond only reading about a situation.

These results serve as a preliminary study to alert agent researchers to the impacts that virtual character emotions may have on human users.

9.1 Pure Contagion Study

In this study, I test the existence of and factors contributing to emotional contagion between an image of a virtual character's facial expression and a human subject. The experiment setup involved a still image of a character, a self-report of emotion, and a character assessment. Participants were randomly assigned to see one of the images shown in Figure 9.1, and participants were informed that they would be questioned about the character later. Thus, the study was a 4 (characters) × 2 (expressions) between-subjects design. Ellie is part of the SimCoach[1] project, while Utah is part of the Gunslinger[2] project. Dia was taken from screenshots from Final Fantasy XIII.[3] Finally, Roy was taken from screenshots of the game L.A. Noire.[4] In the self-report of emotion, I asked subjects how strongly they felt each of 8 emotions on a 0-8 Likert scale: angry, joyful, upset, sad, happy, gloomy, irritated, and calm. Only the measure of Happy was used as the other emotions were only included for compliance checking. Specifically, participants that rated both Angry and Joyful higher than 5 and participants that rated Happy and Joyful more than 3 points apart were considered not in compliance.

Finally, a 15-question survey was administered to gauge subjects' perception of the characters shown. Attributes were drawn primarily from the BSRI [Bem, 1974] and included: Aggressive, Affectionate, Friendly, Attractive, Self-Reliant, Warm, Helpful, Understanding, Athletic, Gentle,

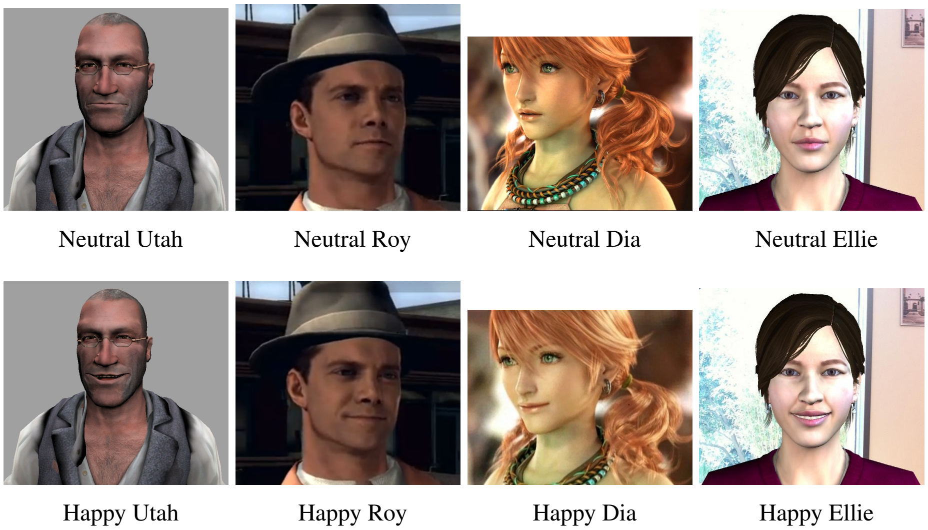
[1] http://ict.usc.edu/projects/simcoach

[2] http://ict.usc.edu/projects/gunslinger/

[3] www.finalfantasyxiii.com

[4] www.rockstargames.com/lanoire/

and Likable. Every question was asked on a 0-8 Likert scale. Compliance tests included duplicating the Attractiveness question and ensuring both occurrences were within 2 points of each other, an Unattractiveness question which could not exceed 5 if Attractiveness exceeded 5, and finally a question that simply asked participants to 'Pick number eight'. Participants were also asked to rate how happy the character seemed.


Neutral Utah Neutral Roy Neutral Dia Neutral Ellie

Happy Utah Happy Roy Happy Dia Happy Ellie

Figure 9.1: Characters used, neutral and happy expressions (color)

A total of 415 participants that responded to the experiment, conducted on Amazon Mechanical Turk, passed the compliance tests. Participants were required to be over 18 years of age and were compensated $0.25. The gender distribution was approximately one-third female and two-thirds male, and approximately two-thirds of respondents indicated their ethnicity as Indian.

	Condition	Mean	SD	n	p
Utah	Neutral	3.96	2.54	57	< 0.001
	Happy	5.60	2.12	52	
Roy	Neutral	4.00	2.45	45	< 0.001
	Happy	5.75	1.86	55	
Dia	Neutral	4.04	2.26	46	< 0.001
	Happy	5.96	2.19	47	
Ellie	Neutral	4.49	2.37	66	< 0.001
	Happy	5.27	2.10	47	

Table 9.1: Happiness statistics for Pure Contagion Study

9.1.1 Results

I examined whether the facial emotion expressed affected subjects' self-report of emotion. For each of the characters used, participants rated the image used in the Happy condition as significantly happier than the image used in the Neutral condition ($p < 0.001$ for all characters). Based on previous findings in human-human contagion [Wild et al., 2001], participants should report greater happiness in the Happy condition compared to the Neutral condition. Table 9.1 shows the means, standard deviations, sample size, and p-values for each experiment. As can be seen, greater happiness was reported in the Happy condition for every character and one-way ANOVA tests revealed significance in every case. This supports our primary hypothesis that an image of a virtual character will cause emotional contagion with a human viewer, since the display of happiness resulted in reports of higher happiness in subjects as compared to the neutral display. Analysis was also conducted to examine a number of additional hypotheses that have been observed in human-human contagion, but none yielded consistent, statistically significant results. These included differences in contagion strength depending on subject gender, ethnicity, perceived character happiness, and perceived character attractiveness.

9.2 Strategic Situation Study

Having found preliminary experimental support for the existence of agent-human emotional contagion, I extend the research to include a strategic interaction. Studies into the effects of emotional contagion have primarily been in mimicry, self-reports of emotion, and other non-decision-based effects such as changes in trust inventory responses and judge ratings of 'cooperativeness' [Barsade, 2002; Dunn and Schweitzer, 2005]. While there has been some work in behavioral changes due to emotional contagion, such as its impact on donation amounts [Small and Verrochi, 2009], my work is the first to consider impacts in a strategic context. The experimental setup involved a still image of a character along with the presentation of a strategic situation for which a decision must be made, followed finally by a self-report of emotion.

I used a cooperation situation based on the standard game-theoretic Stag Hunt situation. The actual story used in this experiment casts the Stag Hunt scenario in a less outlandish context in which the subject and a coworker he/she has never met are tasked with decorating specific rooms in the office and can either choose to work separately (taking more time) or work together through both of their assigned rooms (taking less time). The amount of time it would take to perform the decoration task was not explicitly stated. The coworker in question was the character whose image is presented with the situation. Subjects were asked how likely they were to help the character with the task on a 0-8 Likert scale. A total of 572 participants responded to the experiment, which was again conducted via Amazon Mechanical Turk, passed the compliance tests.

9.2.1 Results

In light of the very strong effect found in the Pure Contagion Study and research indicating that the emotional contagion of happiness leads to more trust [Dunn and Schweitzer, 2005], we expect to see increased happiness in Happy conditions lead to increased likelihood of cooperation. Indeed, we do see a tight link between likelihood of cooperation and participant happiness as shown in Figure 9.2. The x-axis plots the happiness rating, and the y-axis indicates the average likelihood of cooperation for all respondents with the given happiness rating across all conditions. As the regression's very high R-squared of 0.852 indicates, the two measures are very tightly linked.

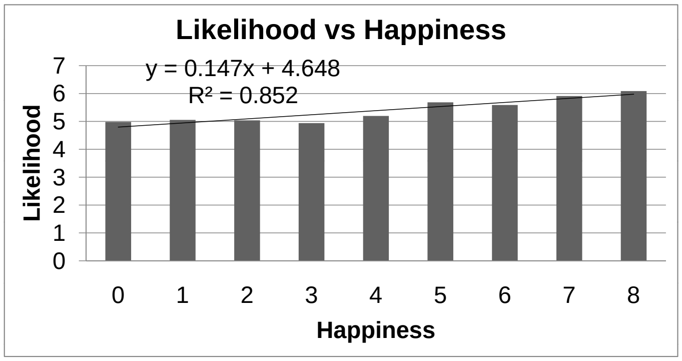


Figure 9.2: Likelihood of cooperation versus happiness

However, only the experiment with Dia yielded a statistically significant change in responses. This suggests that the change results from a character-specific attribute and not simply an expression-based mechanism. The lack of effect for the other characters is due partially to the regression's low coefficient of 0.147, which implies that huge changes in happiness are required to induce changes in the likelihood of cooperation. However, the Pure Contagion Study *did* find very large changes in happiness that should have been sufficient. A closer look at the emotional influence of our manipulation reveals the second half of the story.

(a) Strategic Situation Study

	Condition	Mean	SD	*n*	*p*
Utah	Neutral Happy	4.92 5.02	2.56 2.48	105 125	0.7638
Roy	Neutral Happy	4.53 4.86	2.38 2.76	36 49	0.2098
Dia	Neutral Happy	4.37 5.68	2.57 2.30	41 38	0.019
Ellie	Neutral Happy	5.24 5.69	2.59 2.39	93 85	0.2231

(b) Strategic Decision Study

	Condition	Mean	SD	*n*	*p*
Utah	Neutral Happy	4.04 5.09	2.67 2.63	27 32	0.1329
Roy	Neutral Happy	4.83 5.66	2.33 2.53	24 29	0.2247
Dia	Neutral Happy	5.88 6.28	2.11 2.08	48 46	0.3485
Ellie	Neutral Happy	4.76 5.95	2.33 1.77	46 41	0.008

Table 9.2: Self-reported happiness of participants

While the Pure Contagion Study reported astoundingly large effects of a smile in a still image of a virtual character, the addition of a strategic situation and decision may have altered the contagion effect. Thus, I examine them in this experiment again. We summarize the overall results for each character in Table 9.2a. As before, we expect subjects in the Happy condition to report higher happiness than subjects in the Neutral condition across all characters. This was indeed the case, however, the effect sizes are much smaller than in the Pure Contagion Study and, in fact, statistical significance was found only in the experiment using Dia, indicating that something character-specific is allowing her to retain more of her emotional impact while all other characters experienced a much greater dampening of emotional impact. In exploring the attributes surveyed in this work, no candidate for a consistent explanatory variable was found.

These results suggest that the presentation of a strategic situation and a trust-based decision dampens the emotional contagion effect. This is in line with findings by researchers in social psychology [Wilson et al., 2000; Small and Verrochi, 2009] that found that deliberative thinking can dampen emotional influences. However, in light of the tight correlation between the decision and reported happiness, I hypothesize that the decision itself contributes to the dampening effect beyond the impact of simply reading about the situation.

9.3 Strategic Decision Study

This study was pursued to disentangle the novel effect of making a strategic decision from the previously found effect of reading a situation description [Wilson et al., 2000; Small and Verrochi, 2009]. It presents subjects with the same situation as in the Strategic Situation Study but removes the decision element from it and simply states that the subject will be cooperating with the character shown to complete the office decoration task. In Table 9.2b, the overall results of the experiment are shown. As would be expected following findings in social psychology that even reading additional material can dampen emotional influence [Wilson et al., 2000; Small and Verrochi, 2009], the effect observed in the Pure Contagion Study has not returned in full force. However, the average happiness reported by participants shows a much larger differential than in the Strategic Situation Study, supporting the hypothesis that the decision itself contributed substantially to the dampening of emotional contagion.

Chapter 10: Conclusions

Game-theoretic approaches to scheduling randomized patrols have proven highly effective in deployed applications such as ARMOR for the LAX [Pita et al., 2008], IRIS for the FAMS [Tsai et al., 2009], GUARDS for the TSA [Pita et al., 2011], PROTECT for the Boston Coast Guard [Shieh et al., 2012], and TRUSTS for the Los Angeles Metro Rail System [Yin et al., 2012]. Applying game-theoretic techniques to networked competitive contagion scenarios, however, is non-trivial. Indeed, past work in competitive influence maximization focused only on best-response problems which were repeatedly shown to be NP-Hard under a variety of settings. Given the intractability of even the best-response problem, it is unsurprising that researchers have not pursued equilibrium solutions in the past. My work has shown not only the feasibility of equilibrium calculations in these domains via a double oracle method, but also shown a surprising robustness to uncertainty across a variety of experimental settings. Extending these game-theoretic techniques to contagion-based situations in social networks opens the door to an entirely new set of real-world problems for game theory to help address. These include applications in marketing [Goldenberg et al., 2001; PHELPS et al., 2004; Kiss and Bichler, 2008; Trusov et al., 2009], public health [Christakis and Fowler, 2007; Fowler and Christakis, 2008], and misinformation control [Heussner, 2009; Morozov, 2009] among numerous others.

10.1 Contributions: Game-theoretic Resource Allocation

- Large-scale networked domains without contagion: RANGER is a novel, polynomial-time algorithm for determining resource allocation in large-scale networked domains without contagion effects that reasons over marginal distributions and assumes additive capture probabilities on edges. Combined with the proposed sampling techniques, Radius Sampling and Comb Sampling, it provides a provably efficient solution to path-based problems in networks such as urban road network security. Although the quality of solutions is not guaranteed, experiments in synthetic and real-world networks suggests that the techniques provide high quality solutions in practice. Furthermore, this work has given rise to an independently active line of research [Jain et al., 2011a; Yang et al., 2012a; Jain et al., 2013] that can now provide optimal security allocation strategies to urban areas as large as the city of Mumbai in a matter of hours [Jain et al., 2013].

- Large-scale networked domains with contagion: Combining my studies in large-scale networked domains with my work in contagion phenomena, I provide the first techniques to calculate equilibrium strategies for networked competitive contagion scenarios via a double oracle approach. I prove approximation quality bounds on the double oracle approach when one of the oracles is approximated and combine this with a greedy approximate oracle to produce an approximate algorithm. To further increase scalability, I introduce two heuristic oracles, LSMI and PAGERANK, that offer much greater efficiency. I provide an extensive experimental exploration of a variety of combinations of oracles, testing runtime and quality on random scale-free graphs, a real-world leadership network in Afghanistan, synthetic leadership networks, and a real-world social network. The performance of the

PAGERANK oracle suffers minimal loss compared to LSMI in leadership networks that possess clusters of highly interconnected nodes, but performs far worse in sparsely interconnected real-world social networks and scale-free graphs.

- Large-scale networked domains with contagion under uncertainty: Extending my first step into networked competitive contagion scenarios, I address the common concern of uncertain network structure information in these domains. Modeling the problem as a Bayesian game, I explore a random sampling technique by only sampling *one or two* out of 40+ types. Despite theoretical and empirical intuition suggesting that this should fail spectacularly, I find random sampling to yield near-optimal solutions across a wide array of experimental conditions. Experiments were conducted on 3 different synthetic graph models with and without resource imbalances on both sides, 5 models of uncertainty, weighted/unweighted counting of nodes, varied edge weight distributions, varied graph sizes, varied degrees of uncertainty, and varied degrees of sampling. I also conducted experiments on two real-world social networks using two different models graph construction. In all, I studied over 200 experimental settings and consistently observed the same result: simple sampling techniques perform near-optimally.

10.2 Contributions: Contagion studies

- Contagion management in crowds: To better understand the interaction of fear contagion in crowds and fear-mitigation of authority figures during evacuations, I have built an evacuation simulation that models a novel set of features that experts have identified as critical in airport evacuations, including: (i) emotional contagion; (ii) different agent types; (iii) informational interactions; (iv) behavioral interactions. The features result in a breadth of emergent behaviors that have been observed in the literature, implying increased fidelity of my simulation as a result of their inclusion. I also show results based on a model of Los Angeles International Airport's Tom Bradley International Terminal with concrete recommendations for better managing the contagion of fear that can be produced with ESCAPES.

- Empirical validation of computational contagion models: In my work on computational emotional contagion, I make the first attempt to compare existing models of emotional contagion and identify key attributes of appropriate models using real data. The ASCRIBE model produced a 14$\%$ improvement *per agent per frame* over the Durupinar model in a 15s clip and a 12$\%$ improvement in only a *four-second* clip. After attempts to transform the Durupinar model into one more similar to the ASCRIBE model with little success. This suggests that the primary cause of the statistically significantly worse performance found with the epidemiological / social contagion model is in the mechanism of contagion itself, which is probabilistic and uses a binary representation of the effect. Although the ASCRIBE model requires setting $(N^2 + N)$ parameters to model N agents, even when we do away with them by fixing openness/receptiveness and only formulaically varying channel strength, the model produces superior results, implying that the underlying heat dissipation-style mechanism is better-suited to the phenomenon.
- Emotional contagion with virtual characters: In this work, I provide an examination of agent-human emotional contagion across a wide variety of character types. There is support for its existence with a pure contagion study with strong results. In a second study, a

strategic decision is added that greatly dampens the contagion effect and, with one exception, did not impact behavior. The final study, which removes the user's decision from the previous experiment, finds that the emotional contagion effect returns significantly. This supports the hypothesis that a strategic decision posed to users will dampen the emotional contagion effect beyond the dampening effect of reading the situation itself. In addition, there is evidence of a gender-based difference in susceptibility to cognitive load's dampening effect on emotional contagion.

10.3 Future Work

As society becomes more interconnected and data about these connections becomes more readily available, social networks have take center-stage as the new battleground for businesses, governments, and individuals to push their competing agendas. Analysis of these networks as well as techniques for strategic interaction in them will only become more critical in the future. While my work has introduced the use of game-theoretic resource allocation for networked competitive contagion scenarios, numerous avenues of future research can readily be envisioned.

Within the existing framework I have introduced, scalability, solution quality, and uncertainty remain open challenges. My techniques have introduced methods that allow for scaling to networks with hundreds or thousands of nodes, but far larger social networks with millions of nodes remain out of reach. With my algorithms being heuristic in nature, although the general problem being NP-Hard, the door remains open for specialized techniques that provide superior quality in particular domains of interest. Finally, although I have shown a surprising robustness to random uncertainty and headway has been made in understanding the mechanisms by which this occurs under conditions of interest for counterinsurgency, there remain other forms of uncertainty to explore such as biased or even adversarial uncertainty that may behave very differently.

Beyond the model that I have introduced, other approaches may offer additional insight into specific types of networked competitive contagion scenarios. For example, in some domains, the contagious effect occurs slowly enough that information can be gathered about its current progress and intervention can occur during the spreading process. This would be more appropriately modeled as a multi-stage game in which players can choose *when* to act as well as what action to take. As computational techniques become more sophisticated, refining the models to capture the intricacies of real-world scenarios more accurately will become possible. Constantly improving these models will then drive the need for more scalable, robust algorithms. In an area as complex, dynamic, and pervasive as networked competitive contagion scenarios, this cycle will undoubtedly be a lifelong pursuit for many researchers in the future.

Bibliography

James Aspnes, Kevin Chang, and Aleksandr Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 43–52, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0-89871-585-7. URL `http://dl.acm.org/citation.cfm?id=1070432.1070440`.

James Aspnes, Navin Rustagi, and Jared Saia. Worm versus alert: who wins in a battle for control of a large-scale network? In *Proceedings of the 11th international conference on Principles of distributed systems*, OPODIS'07, pages 443–456, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-77095-X, 978-3-540-77095-4. URL `http://dl.acm.org/citation.cfm?id=1782394.1782426`.

Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, October 1999.

Sigal G. Barsade. The Ripple Effect: Emotional Contagion and Its Influence on Group Behavior. *Administrative Science Quarterly*, 47:644–675, 2002.

Sigal G. Barsade and Donald E. Gibson. Group Emotion: A View from Top and Bottom. In D. Gruenfeld, E. Mannix, , and M. Neale, editors, *Research on Managing on Groups and Teams*, pages 81–102. JAI Press, 1998.

Nicola Basilico and Nicola Gatti. Automated abstractions for patrolling security games. In *AAAI*, 2011.

Kyle Beardsley. Peacekeeping and the contagion of armed conflict. *The Journal of Politics*, 73 (4):1051–1064, October 2011.

Sandra L. Bem. The measurement of psychological androgyny. *Journal of Consulting and Clinical Psychology*, 42:155–162, 1974.

Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *WINE*, pages 306–311, 2007.

Garrett Birkhoff. Tres observaciones sobre el algebra lineal. Univ. Nac. Tucuman. Revista A. 5:147-151 (1946), 1946.

Allan Borodin, Yuval Filmus, and Joel Oren. Threshold models for competitive influence in social networks. In *WINE*, pages 539–550, 2010.

Tibor Bosse, Rob Duell, Zulfiqar A. Memon, Jan Treur, and C. Natalie Van Der Wal. A Multi-Agent Model for Mutual Absorption of Emotions. In *ECMS-09*, 2009a.

Tibor Bosse, Rob Duell, Zulfiqar A. Memon, Jan Treur, and C. Natalie Van Der Wal. A Multi-Agent Model for Emotion Contagion Spirals Integrated within a Supporting Ambient Agent Model. In *PRIMA-09*, 2009b.

Tibor Bosse, Mark Hoogendoorn, Michel Klein, Jan Treur, and Natalie van der Wal. Agent-Based Analysis of Patterns in Crowd Behaviour Involving Contagion of Mental States. In *IEA/AIE-11*. Springer Verlag, 2011.

Michele Breton, A. Alg, and Alain Haurie. Sequential Stackelberg equilibria in two-person games. *Optimization Theory and Applications*, 59(1):71–97, 1988.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30:107–117, April 1998.

Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *WWW*, pages 665–674, 2011.

Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Implementing Random Assignments: A Generalization of the Birkhoff-von Neumann Theorem. In *2009 Cowles Summer Conference*, 2009.

C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3-4):507–525, 2001.

C. Burstedde, A. Kirchner, K. Klauck, A. Schadschneider, and J. Zittartz. Cellular automaton approach to pedestrian dynamics-application. In *Pedestrian and Evacuation Dynamics*, pages 87–97. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-42690-5.

W.M. Chao and T.Y. Li. Simulation of Social Behaviors in Virtual Crowd. In *In CASA-10*, 2010.

Po-An Chen, Mary David, and David Kempe. Better vaccination strategies for better people. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, pages 179–188, New York, NY, USA, 2010a. ACM. ISBN 978-1-60558-822-3. doi: 10.1145/1807342.1807370. URL `http://doi.acm.org/10.1145/1807342.1807370`.

Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, pages 1029–1038, 2010b.

Jerome M Chertkoff and Russell H. Kushigian. *Don't Panic: The Psychology of Emergency Egress and Ingress*. Praeger Publishers, 1999.

Nicholas A. Christakis and James H. Fowler. The spread of obesity in a large social network over 32 years. *New England Journal of Medicine*, 357(4):370–379, 2007.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Rev.*, 51:661–703, November 2009.

Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, New York, NY, USA, 2006. ACM. ISBN 1-59593-236-4.

Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, Xiangyang Lan, and Siddharth Suri. Sequential influence models in social networks. 2010. URL `http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1530`.

Celso de Melo, Peter Carnevale, and Jonathan Gratch. The Effect of Expression of Anger and Happiness in Computer Agents on Negotiations with Humans. In *AAMAS-11*, 2011.

Morris H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69 (345):118–121, March 1974.

Jonathan Diamond, Melinda McVay, and Mary Wassel Zavala. Quick, Safe, Secure: Addressing Human Behavior During Evacuations at LAX. Master's thesis, UCLA Department of Public Policy, June 2010.

Peter Dodds and Duncan J Watts. A generalized model of social and biological contagion. *Journal of Theoretical Biology*, 232(4):587–604, 2005.

William Doherty. The Emotional Contagion Scale: A Measure of Individual Differences. *Journal of Nonverbal Behavior*, 21(2), 1997.

D.S.Mileti and J.L.Sorensen. Communication of Emergency Public Warnings: A Social Science Perspective and State-of-the-Art Assessment. 1990.

Jennifer R. Dunn and Maurice E. Schweitzer. Feeling and Believing: The Influence of Emotion on Trust. *Psychology of Personality and Social Psychology*, 88(5):736–748, 2005.

Funda Durupinar. *From Audiences to Mobs: Crowd Simulation with Psychological Factors*. PhD dissertation, Bilkent University, Dept. Comp. Eng, July 2010.

Funda Durupinar, Jan Allbeck, Nuria Pelechano, and Norman Badler. Creating crowd variation with the ocean personality model. In *AAMAS-08*, pages 1217–1220, Richland, SC, 2008. ISBN 978-0-9817381-2-X.

Daniel Enemark, Mathew McCubbins, Ramamohan Paturi, and Nicholas Weller. Does more connectivity help groups to solve social problems. In *Proceedings of the ACM Conference of Electronic Commerce*, pages 21–27, 2011.

Leon Festinger. A theory of social comparison processes. *Human Relations*, pages 117–140, 1954.

Gadi Fibich and Ro'i Gibori. Aggregate diffusion dynamics in agent-based models with a spatial structure. *Oper. Res.*, 58(5):1450–1468, September 2010. ISSN 0030-364X. doi: 10.1287/opre.1100.0818. URL `http://dx.doi.org/10.1287/opre.1100.0818`.

J. P. Forgas. Affective influences on individual and group judgments. *European Journal of Social Psychology*, (20):441–453, 1990.

James H. Fowler and Nicholas A. Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *BMJ*, 337(dec04_2): a2338+, December 2008.

Natalie Fridman and Gal A. Kaminka. Comparing human and synthetic group behaviors: A model based on social psychology. In *ICCM-09*, 2009.

Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001.

Alicia A. Grandey. Emotional regulation in the workplace: A new way to conceptualize emotional labor. *Journal of Occupational Health Psychology*, 5(1):95–110, January 2000.

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

Jonathan Gratch, Anya Okhmatovskaia, Francois Lamothe, Stacy Marsella, Mathieu Morales, R. J. van der Werf, and Louis-Philippe Morency. Virtual Rapport. In *IVA-06*, 2006.

Erik Halvorson, Vincent Conitzer, and Ronald Parr. Multi-step multi-sensor hider-seeker games. In *IJCAI*, pages 159–166, 2009.

Elaine Hatfield, John T. Cacioppo, and Richard L. Rapson. *Emotional Contagion*. Cambridge University Press, 1994.

Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, pages 463–474, 2012.

D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5): 4282–4286, 1995.

Dirk Helbing, Illes J. Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

LF Henderson. On the fluid mechanics of human crowd motion. *Transportation research*, 8(6): 509–515, 1974.

Ursula Hess, Pierre Philippot, and Sylvie Blairy. Facial Reactions to Emotional Facial Expressions: Affect or Cognition? *Cognition and Emotion*, 12(4):509–531, 1998.

Ki Mae Heussner. Ft. hood solider causes stir on twitter. *ABC News (online)*, 2009.

M. Hoogendoorn, J. Treur, C.N. van der Wal, and A. van Wissen. An Agent-Based Model for the Interplay of Information and Emotion in Social Diffusion. In *IAT-10*, 2010.

S. Hoogendoorn and P.H.L. Bovy. Gas-kinetic modeling and simulation of pedestrian flows. *Transportation Research Record: Journal of the Transportation Research Board*, 1710(-1): 28–36, 2000.

Nicholas J. Howard. *Finding optimal strategies for influencing social networks in two player games*. Masters thesis, MIT, Sloan School of Management, June 2011.

Benjamin W. K. Hung. *Optimization-Based Selection of Influential Agents in a Rural Afghan Social Network*. Masters thesis, MIT, Sloan School of Management, June 2010.

Benjamin W. K. Hung, Stephan E. Kolitz, and Asuman E. Ozdaglar. Optimization-based influencing of village social networks in a counterinsurgency. In *SBP*, pages 10–17, 2011.

Eitan Israeli and R. Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

Manish Jain, Erim Kardes, Christopher Kiekintveld, Fernando Ordóñez, and Milind Tambe. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, 2010.

Manish Jain, Dmytro Korzhyk, Ondrej Vanek, Vincent Conitzer, Michal Pechoucek, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 327–334, 2011a.

Manish Jain, Milind Tambe, and Christopher Kiekintveld. Quality-bounded solutions for finite bayesian stackelberg games: Scaling up. In *In AAMAS*, pages 997–1004, 2011b.

Manish Jain, Vincent Conitzer, and Milind Tambe. Security scheduling for real-world networks. In *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.

J.L.Bryan. Behavioral response to fire and smoke. In *SFPE Handbook of Fire Protection Engineering*, pages 3315–3341. National Fire Protection Association, third edition, 2002.

Darrick Jolliffe and David P. Farrington. Development and validation of the basic empathy scale. *Journal of Adolescence*, 29(4):589–611, 2006.

Michael Kearns, Siddharth Suri, and Nick Montfort. An experimental study of the coloring problem on human subject networks. *Science*, 313(5788):824–827, 2006. doi: 10.1126/science.1127207. URL `http://www.sciencemag.org/content/313/5788/824.abstract`.

Herbert C. Kelman. Compliance, identification, and internalization: Three processes of opinion change. *Journal of Conflict Resolution*, 2:51–60, 1958.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of The Royal Society of London. Series A, Containing Papers of A Mathematical and Physical Character (1905-1934)*, 115:700–721, 1927.

Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Milind Tambe, and Fernando Ordóñez. Computing Optimal Randomized Resource Allocations for Massive Security Games. In *AAMAS-09*, 2009.

Christopher Kiekintveld, Janusz Marecki, and Milind Tambe. Approximation methods for infinite bayesian stackelberg games: Modeling distributional payoff uncertainty. In *International Conference on Autonomous Agents and Multiagent Systems*, 2011.

Masahiro Kimura, Kazumi Saito, Ryohei Nakano, and Hiroshi Motoda. Extracting influential nodes on a social network for information diffusion. *Data Min. Knowl. Discov.*, 20(1):70–97, 2010.

Christine Kiss and Martin Bichler. Identification of influencers - measuring influence in customer networks. *Decis. Support Syst.*, 46(1):233–253, December 2008. ISSN 0167-9236. doi: 10.1016/j.dss.2008.06.007. URL http://dx.doi.org/10.1016/j.dss.2008.06.007.

Jan Kostka, Yvonne Anne Oswald, and Roger Wattenhofer. Word of mouth: Rumor dissemination in social networks. In *SIROCCO*, pages 185–196, 2008.

James A. Kulik and Heike I. M. Mahler. Social comparison, affiliation, and emotional contagion under threat. In *Handbook of social comparison: Theory and research*. New York: Plenum, 2000.

V. S. Anil Kumar, Rajmohan Rajaraman, Zhifeng Sun, and Ravi Sundaram. Existence theorems and approximation algorithms for generalized network security games. In *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems*, ICDCS '10, pages 348–357, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4059-7. doi: 10.1109/ICDCS.2010.70. URL http://dx.doi.org/10.1109/ICDCS.2010.70.

Mayank Lahiri, Arun S. Maiya, Rajmonda Sulo Caceres, Habiba, and Tanya Y. Berger-Wolf. The impact of structural changes on predictions of diffusion in networks. In *ICDM Workshops*, pages 939–948, 2008.

George Leitmann. On generalized Stackelberg strategies. *Optimization Theory and Applications*, 26(4):637–643, 1978.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M. VanBriesen, and Natalie S. Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420–429, 2007.

Yiqing Lin, Igor Fedchenia, Bob LaBarre, and Robert Tomastik. Agent-based simulation of evacuation: An office building case study. In *Pedestrian and Evacuation Dynamics 2008*, pages 347–357. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-04504-2.

Lars-Olov Lundqvist. Factor Structure of the Greek Version of the Emotional Contagion Scale and its Measurement Invariance Across Gender and Cultural Groups. *Journal of Individual Differences*, 29(3):121–129, 2008.

Arthur Lupia and Mathew D. Mccubbins. *The Democratic Dilemma : Can Citizens Learn What They Need to Know? (Political Economy of Institutions and Decisions)*. Cambridge University Press, March 1998.

Mathew D. McCubbins, Ramamohan Paturi, and Nicholas Weller. Connected Coordination: Network Structure and Group Coordination. August 2008. URL http://ssrn.com/abstract=1201387.

H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *ICML*, pages 536–543, 2003.

Evgeny Morozov. Swine flu: Twitter's power to misinform. *Foreign Policy*, 2009.

James D. Murray. *Mathematical Biology*. Springer, third edition, 2002.

Mark Newman. *Networks: An Introduction*. Oxford University Press, 2010.

Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *The Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 895–902, 2008.

Nuria Pelechano. Crowd simulation incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation*, pages 21–30, 2005.

JOSEPH E. PHELPS, REGINA LEWIS, LYNNE MOBILIO, DAVID PERRY, and NIRANJAN RAMAN. Viral marketing or electronic word-of-mouth advertising: Examining consumer responses and motivations to pass along email. *Journal of Advertising Research*, 44:333–348, 11 2004. ISSN 1740-1909. doi: 10.1017/S0021849904040371. URL `http://journals.cambridge.org/article_S0021849904040371`.

James Pita, Manish Jain, Janusz Marecki, O Ordez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport. In *The Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS) - Industry Track*, 2008.

James Pita, Milind Tambe, Chris Kiekintveld, Shane Cullen, and Erin Steigerwald. Guards - game theoretic security allocation on a national scale. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2011.

James Pita, Richard John, Rajiv Maheswaran, Milind Tambe, and Sarit Kraus. A robust approach to addressing human adversaries in security games. In *European Conference on Artificial Intelligence (ECAI)*, 2012.

Guylene Proulx and Rita F. Fahy. Human behavior and evacuation movement in smoke. *ASHRAE Transactions*, July 2008.

S. Douglas Pugh. Service with a smile: Emotional contagion in the service encounter. *Academy of Management Journal*, 44(5):1018–1027, 2001.

Giuseppe Riva. Virtual Reality in Psychotherapy: Review. *CyberPsychology & Behavior*, 8(3): 220–230, 2005.

Albert Rizzo, Jarrell Pair, Peter J. McNerney, Ernie Eastlund, Brian Manson, Jonathan Gratch, Randall W. Hill, and William Swartout. *Development of a VR Therapy Application for Iraq War Military Personnel with PTSD*, volume 111 of *Medicine Meets Virtual Reality*, pages 407–413. IOS Press, 13th Annual Medicine Meets Virtual Reality Conference, Long Beach, CA, 2005. URL http://ict.usc.edu/pubs/Development%20of%20a%20VR%20Therapy%20Application%20for%20Iraq%20War%20Veterans%20with%20PTSD.pdf.

E.M. Rogers. *Diffusion of Innovations*. The Free Press, New York, 1962.

Barbara O. Rothbaum, Larry F. Hodges, David Ready, Ken Graap, and Renato D. Alarcon. Virtual reality exposure therapy for vietnam veterans with posttraumatic stress disorder. *Journal of Clinical Psychiatry*, 63(8):617–622, August 2001.

Harold E. Russell and Alan Beigel. *Understanding Human Behavior for Effective Police Work*. Basic Books, 1976.

Thomas C. Schelling. Hockey helmets, concealed weapons, and daylight saving: a study of binary choices with externalities. *Journal of Conflict Resolution*, 17:381–428, 1973.

Thomas C. Schelling. *Micromotives and macrobehavior*. Fels lectures on public policy analysis. Norton, 1978.

C. Seshadhri, Tamara G. Kolda, and Ali Pinar. Community structure and scale-free collections of Erdös-Rényi graphs. *Phys. Rev. E*, 85:056109, May 2012.

A. Shapiro and T. Homem-de Mello. On the rate of convergence of optimal solutions of monte carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11(1):70–86, 2000. doi: 10.1137/S1052623498349541. URL http://epubs.siam.org/doi/abs/10.1137/S1052623498349541.

Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. Protect: A deployed game theoretic system to protect the ports of the united states. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

Deborah A. Small and Nicole M. Verrochi. The Face of Need: Facial Emotion Expression on Charity Advertisements. *Journal of Marketing Research*, 46(6):777–787, December 2009.

C. A. Smith and P. C. Ellsworth. Patterns of cognitive appraisal in emotion. *Journal of Personality and Social Psychology*, (48):813–838, 1985.

Weiguo Song, Yanfei Yu, Weicheng Fan, and Heping Zhang. A cellular automata evacuation model considering friction and repulsion. *Science in China Series E: Technological Sciences*, 48:403–413, 2005. ISSN 1006-9321.

Siddharth Suri and Duncan J. Watts. Cooperation and contagion in networked public goods experiments. *CoRR*, abs/1008.1276, 2010.

P.C. Tissera, M. Printista, and M.L. Errecalde. Evacuation simulations using cellular automata. *Journal of Computer Science and Technology*, 7(1):14–20, 2007.

Michael Trusov, Randolph E. Bucklin, and Koen Pauwels. Effects of word-of-mouth versus traditional marketing: Findings from an internet social networking site. *Journal of Marketing*, 73, September 2009.

Jason Tsai, Shyamsunder Rathi, Christopher Kiekintveld, Fernando Ordez, and Milind Tambe. Iris: A tool for strategic security allocation in transportation networks. In *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS) - Industry Track*, 2009.

Jason Tsai, Zhengyu Yin, Junyoung Kwak, David Kempe, Christopher Kiekintveld, and Milind Tambe. Urban security: Game-theoretic resource allocation in networked physical domains. In *National Conference on Artificial Intelligence (AAAI)*, 2010.

Jason Tsai, Emma Bowring, Stacy Marsella, and Milind Tambe. Empirical Evaluation of Computational Emotional Contagion Models. In *International Conference on Intelligent Virtual Agents (IVA)*, 2011a.

Jason Tsai, Natalie Fridman, Emma Bowring, Matthew Brown, Shira Epstein, Gal Kaminka, Stacy Marsella, Andrew Ogden, Inbal Rika, Ankur Sheel, Matthew Taylor, Xuezhi Wang, Avishay Zilka, and Milind Tambe. ESCAPES: Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social Comparison. In *International Conference on Autonomous Agents and Multiagent Systems*, 2011b.

U.S. Dept. of the Army and U.S. Marine Corps. *The U.S. Army/Marine Corps Counterinsurgency Field Manual 3-24*. University of Chicago Press, 2007.

John von Neumann. A certain zero-sum two-person game equivalent to the optimal assignment problem. Contributions to the Theory of Games, II, Ann. Math. Stud. No. 28, 2-15 (1953)., 1953.

Heinrich von Stackelberg. *Marktform und Gleichgewicht*. Springer, Vienna, 1934.

Bernhard von Stengel and Shmuel Zamir. Leadership with commitment to mixed strategies. Technical Report LSE-CDAM-2004-01, CDAM Research Report, 2004.

Ning Wang and Jonathan Gratch. Rapport and Facial Expression. In *ACII-09*, 2009.

Alan Washburn and Kevin Wood. Two-person zero-sum games for network interdiction. *Operations Research*, 43(2):243–251, 1995. ISSN 0030364X. URL `http://www.jstor.org/stable/171833`.

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

Barbara Wild, Michael Erb, and Mathias Bartels. Are emotions contagious? evoked emotions while viewing emotionally expressive faces: quality, quantity, time course and gender differences. *Psychiatry Research*, 102(2):109–124, 2001.

Timothy D. Wilson, Samuel Lindsey, , and Tonya Y. Schooler. A Model of Dual Attitudes. *Psychological Review*, 107(1):101–126, 2000.

Wendy Wood, Sharon Lundgren, Judith A. Ouellette, Shelly Busceme, and Tamela Blackstone. Minority influence: a meta-analytic review of social influence processes. *Psychological Bulletin*, 115(3):323–345, 1994.

Rong Yang, Fei Fang, Albert Xin Jiang, Karthik Rajagopal, Milind Tambe, and Rajiv Maheswaran. Designing better strategies against human adversaries in network security games. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1299–1300, 2012a.

Rong Yang, Fernando Ordonez, and Milind Tambe. Computing optimal strategy against quantal response in security games. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012b.

Sha Yang, Mantian (Mandy) Hu, Russell S. Winer, Henry Assael, and Xiaohong Chen. An empirical study of word-of-mouth generation and consumption. *Marketing Science*, 31(6): 952–963, November 2012c. ISSN 1526-548X. doi: 10.1287/mksc.1120.0738. URL http://dx.doi.org/10.1287/mksc.1120.0738.

Zhengyu Yin and Milind Tambe. A unified method for handling discrete and continuous uncertainty in bayesian stackelberg games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

Zhengyu Yin, Dmytro Korzhyk, Christopher Kiekintveld, Vincent Conitzer, , and Milind Tambe. Stackelberg vs. nash in security games: Interchangeability, equivalence, and uniqueness. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2010.

Zhengyu Yin, Albert Jiang, Matthew Johnson, Milind Tambe, Christopher Kiekintveld, Kevin Leyton-Brown, Tuomas Sandholm, and John Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems. In *Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 2012.

Appendix A: Complexity of Attacker's Best-Response in Path-based Network Security

In Chapter 4, the use of marginals in determining the defender's strategy was partially justified by the computational difficulty of calculating the attacker's best-response. Here we prove the NP-Hardness of the best-response problem via reduction from GRAPH 3-COLORING.

Theorem 5. *Given a defender strategy, computing the attacker's best response is NP-hard.*

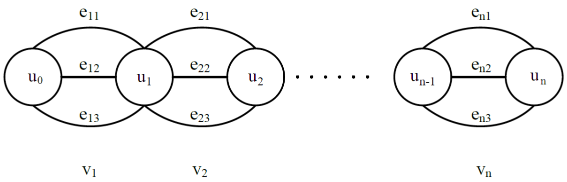


Figure A.1: Graph output of the reduction.

Proof. We will show the result for $r = 2$ checkpoints. Specifically, we prove hardness of the decision problem ATTACK PATH of whether the attacker can attain at least a payoff of M. The reduction is from GRAPH 3-COLORING. For any GRAPH 3-COLORING instance $G = (V, E)$ (with $n = |V|$ and $m = |E|$), we define an instance of the ATTACK PATH problem. There are $n + 1$ nodes $U = \{u_0, \dots, u_n\}$ and $3n$ edges as shown in Figure A.1. u_0 is the only source, and u_n is the only target, with a value of 1. For each $i = 0, \dots, n-1$, there are three parallel edges $e_{i,1}$, $e_{i,2}$, and $e_{i,3}$ from u_{i-1} to u_i. Designate this graph $H = (U, E')$.

The defender's strategy is defined by a probability distribution over pairs of edges; because the target value is 1, the attacker's payoff for a u_0-u_n path P is the probability that the defender selects no edge on P. Consider a pair of edges $e_{i,s}, e_{j,s'} \in E'$. If $i = j$, or v_i and v_j are not connected in G, then the probability of sampling $e_{i,s}$ and $e_{j,s'}$ together is 0. If there is an edge between v_i and v_j in G, then the probability of sampling both edges simultaneously depends on the indices s, s', and is:

$p(\cdot,\cdot)$	$e_{i,1}$	$e_{i,2}$	$e_{i,3}$
$e_{j,1}$	0	$\frac{1}{6m}$	$\frac{1}{6m}$
$e_{j,2}$	$\frac{1}{6m}$	0	$\frac{1}{6m}$
$e_{j,3}$	$\frac{1}{6m}$	$\frac{1}{6m}$	0

This is a valid strategy since $\sum_{(e,e') \in \mathcal{L}} p(e,e') = 6 \cdot \frac{1}{6m} \cdot m = 1$. Denoting the degree of v_i by $d(v_i)$, we can compute $x_{e_{i,s}}$ as

$$x_{e_{i,1}} \quad = \quad x_{e_{i,2}} \quad = \quad x_{e_{i,3}} \quad = \quad \textstyle\sum_{e'} p(e_{i,s}, e') \quad = \quad \frac{d(v_i)}{3m}.$$

Therefore, for any path P,

$$\textstyle\sum_{e \in P} x_e \quad = \quad \sum_{i=1}^{n} \frac{d(v_i)}{3m} \quad = \quad \frac{1}{3m} \cdot \sum_{i=1}^{n} d(v_i) \quad = \quad \frac{2}{3},$$

and the attacker payoff is $\frac{1}{3} + \sum_{e,e' \in P} p(e,e')$.

The mapping between colorings and attacker paths is straightforward: $e_{i,c}$ is in the attacker path if and only if v_i is assigned color c. By the definition of the joint sampling probabilities, a coloring is valid if and only if the corresponding attacker path has $\sum_{e,e' \in P} p(e,e') \geq \frac{1}{6}$. The reason is that $\frac{1}{6}$ is attained if and only if each edge pair $e, e' \in P$ contributes $\frac{1}{6m}$, which happens if and only if each pair $e = e_{i,c}, e' = e_{j,c'}$ with an edge $(v_i, v_j) \in E$ has $c \neq c'$. Therefore, G has a valid 3-coloring if and only if in the instance we construct, there is an attacker path giving the attacker a payoff of at least $\frac{1}{3} + \frac{1}{6} = \frac{1}{2}$. □

Appendix B: Bayesian Game-Theoretic Contagion Blocking in Networks

B.1 Supplemental Experimental Results

Here I include the complete set of experiments run to test the surprisingly strong performance of Random Sampling in handling the Bayesian formulation of the contagion blocking game presented in Chapter 6. The results are organized based on the model of uncertainty that is being tested. Results include graph type variations, graph size variations, variations in the number of Bayesian types, variations in the number of types sampled, variations in the number of resources, variations in community density in BTER graphs ($\rho = 0.5$ versus $\rho = 0.9$), and variations in the average contagion probability on edges. The experimental settings default to the same values as were presented in Chapter 6, restated here for clarity.

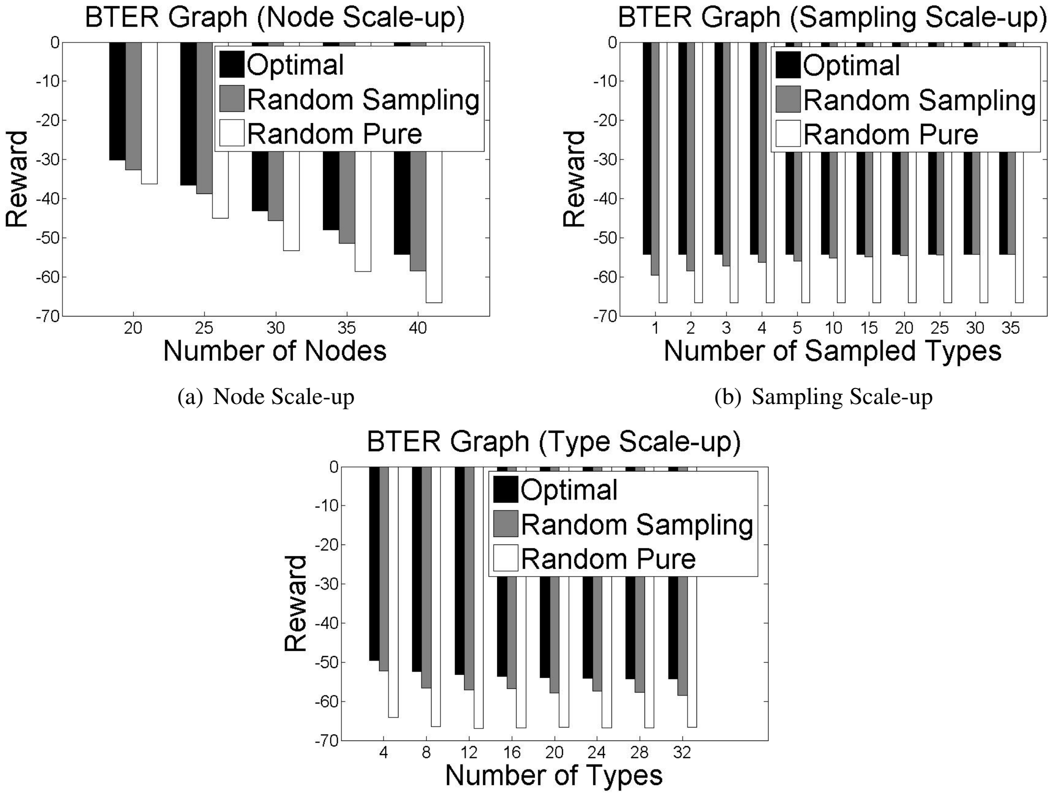
All results are an average of 20 game instances and were run on machines with CPLEX 12.2, 2.8 GHz CPU, and 4GB of RAM. Unless otherwise stated, experiments were run on 40-node graphs (130 to 200 edges), contagion probabilities on edges drawn from a $\mathcal{N}(0.4, 0.2)$ distribution, node values varying uniformly from 1-10, each player having two seed nodes ($|\omega_M| = |\omega_I| = 2$), and payoffs estimated using the LSMI heuristic. Monte Carlo payoff estimations produced similar results but could not be meaningfully scaled. Since an optimal benchmark is necessary, the best-response oracles iteratively evaluate each available action to determine the best response, rather than using greedy hill-climbing common in the influence maximization literature. Unless otherwise stated, Influential Node uncertainty selects 3 nodes and gives each 4 additional edges. Moreover, *only* these 12 edges could potentially connect communities, making the chosen nodes not only more connected (average degree, excluding uncertain edges, varies from 3-5 with maximums of 9), but also incident to the more consequential edges. For Inter-Community Edge Set Uncertainty, each type had 6 randomly chosen intercommunity edges added by default. For Inter-Community Edge, Inter/Intra-Community Edge, and Random Edge uncertainty I varied the number of uncertain edges between 1 and 6 (the optimal technique could not scale to more edges). I focus throughout on the mitigator strategy obtained by drawing a random subset of the influencer's types and solving the game assuming no other types exist (referred to as *Random Sampling*). A worst-case benchmark is also included for comparison and is dubbed RANDOM PURE, which represents the performance of a randomly selected pure strategy by the mitigator against a best-responding influencer. Some graphs include an alternative baseline, *Max Prob*, which solves the Bayesian type with the highest probability.

B.2 Indian Village Network Weighting Scheme

The variable weighting scheme mentioned in Chapter 6 is shown in the list below with the question listed and then the weight following it in parentheses. The total weights were then normalized to define contagion probabilities for each edge. The other weighting scheme tested was to simply fix all weight values to the same value, implicitly weighing all questions equally. These results are shown in Figure **??**.

Individuals were asked who they:

- ..borrow money from (0.3)
- ..give advice to (0.1)
- ..help with a decision (0.1)
- ..borrow kerosene or rice from (0.1)
- ..lend kerosene or rice to (0.1)
- ..lend money to (0.6)
- ..obtain medical advice from (0.1)
- ..engage socially with (0.8)
- ..are related to (0.1)
- ..invite to one’s home (0.4)
- ..visit in another’s home (0.4)

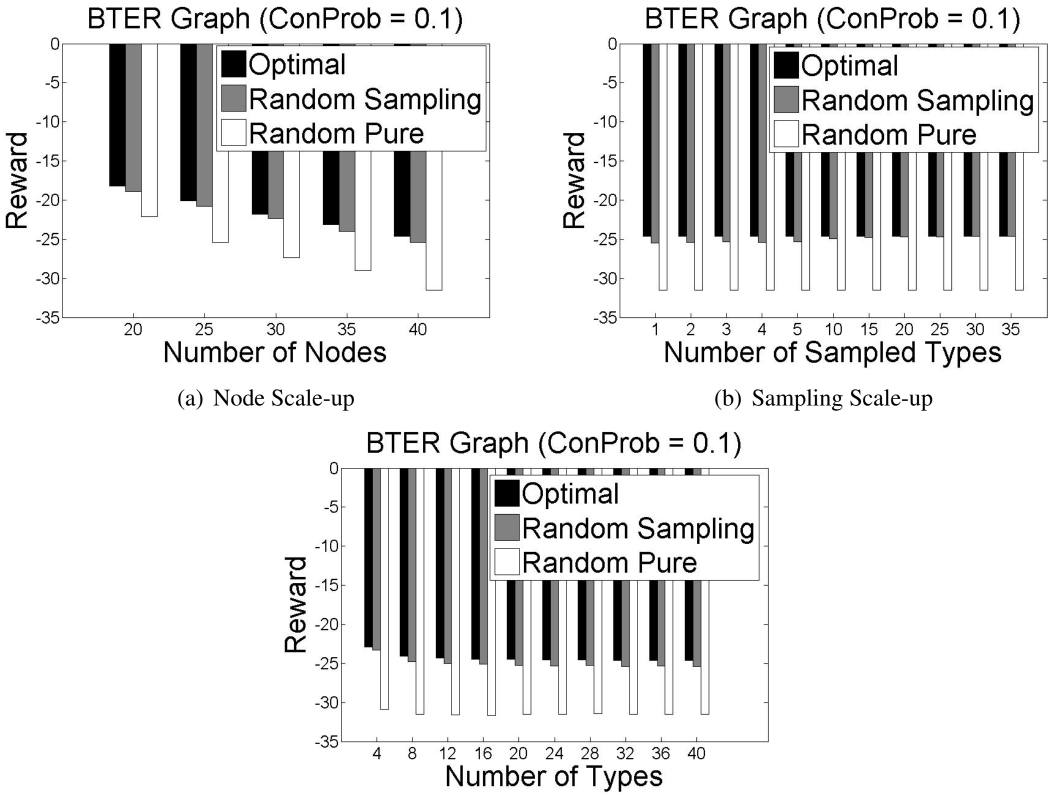


(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

Figure B.1: BTER graph results for standard Influential Node Uncertainty experiments

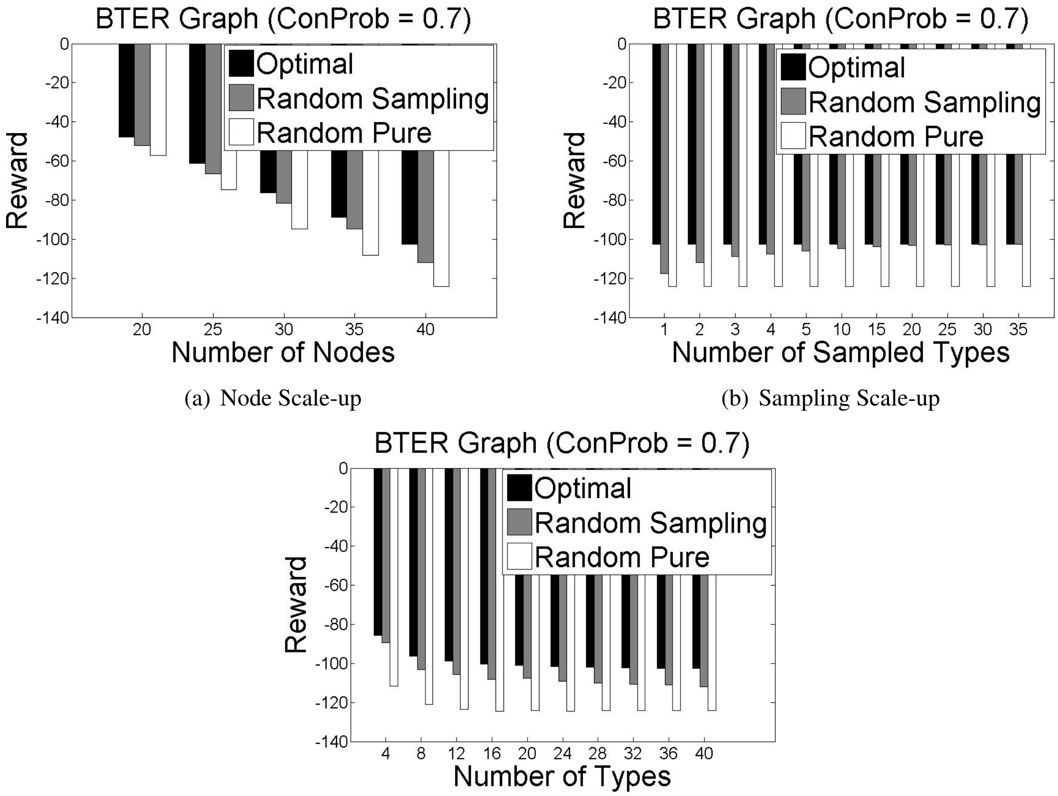


(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

Figure B.2: BTER graph results for Influential Node Uncertainty experiments. Contagion probability on edges drawn from a $\mathcal{N}(0.1, 0.2)$ distribution instead of the standard $\mathcal{N}(0.4, 0.2)$

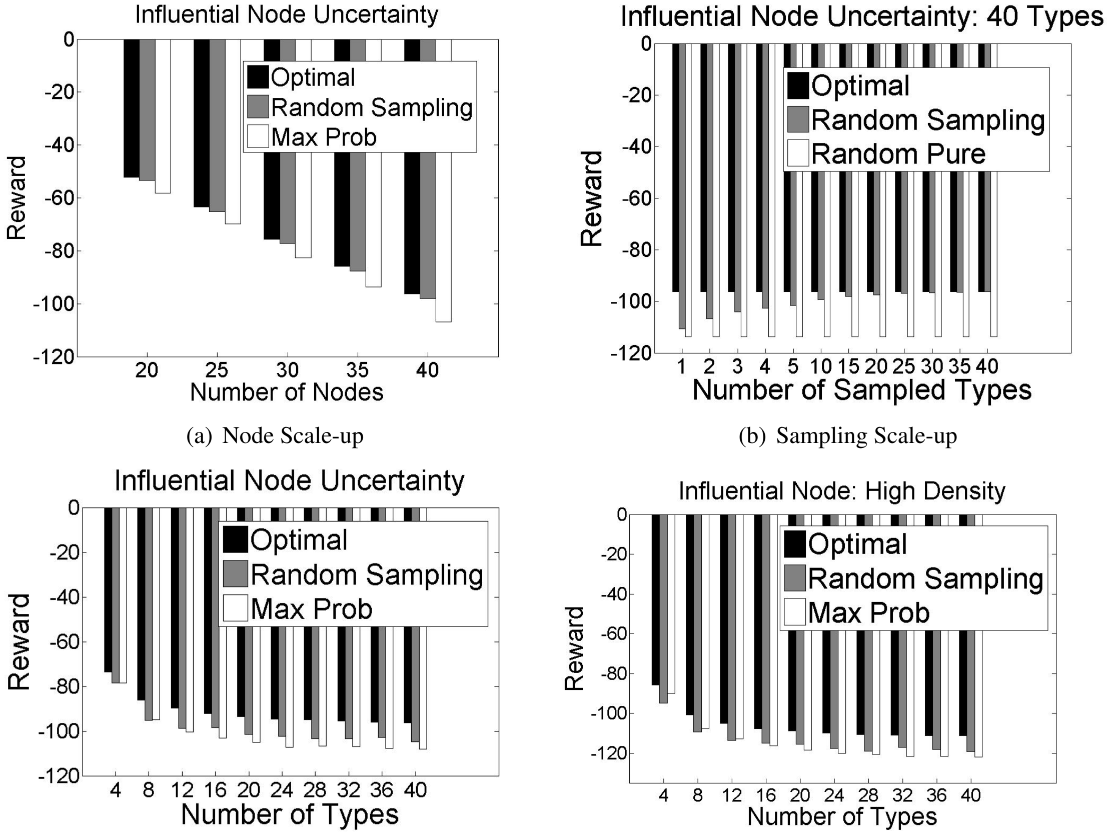


(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

Figure B.3: BTER graph results for Influential Node Uncertainty experiments. Contagion probability on edges drawn from a $\mathcal{N}(0.7, 0.2)$ distribution instead of the standard $\mathcal{N}(0.4, 0.2)$



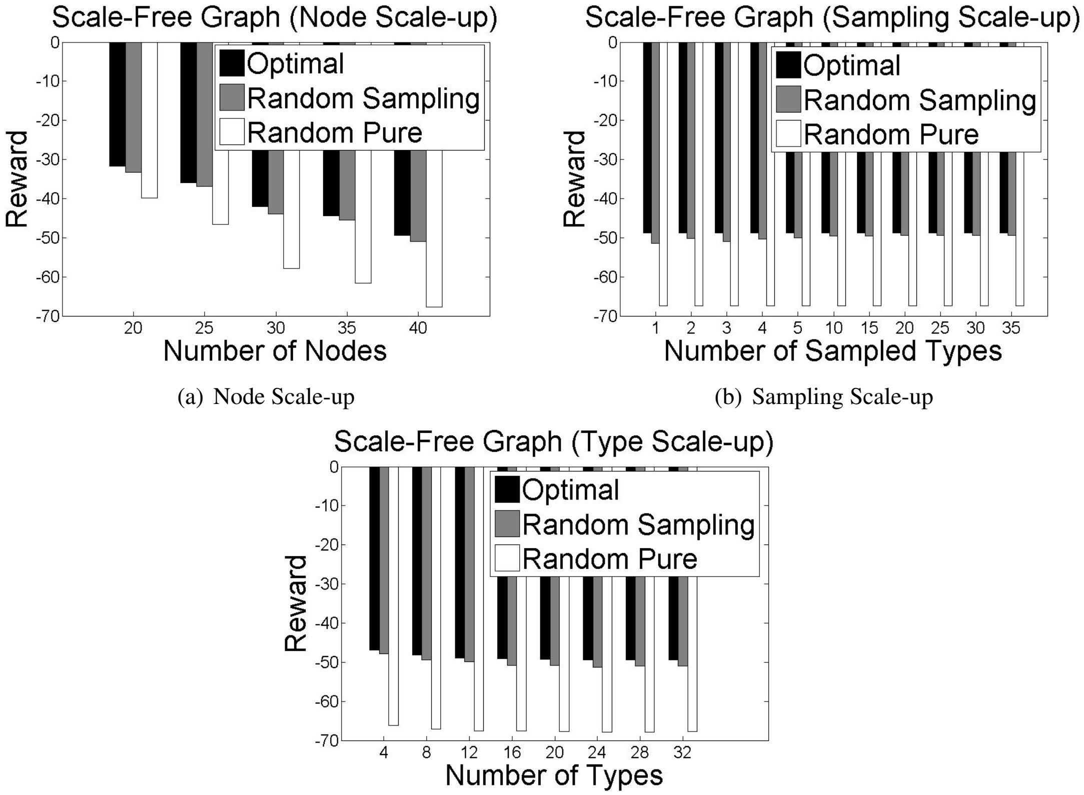
(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

(d) Type Scale-up (High Density)

Figure B.4: BTER graph results for Influential Node Uncertainty experiments, where new edges have 1.0 transmission probability instead of being drawn from a distribution.

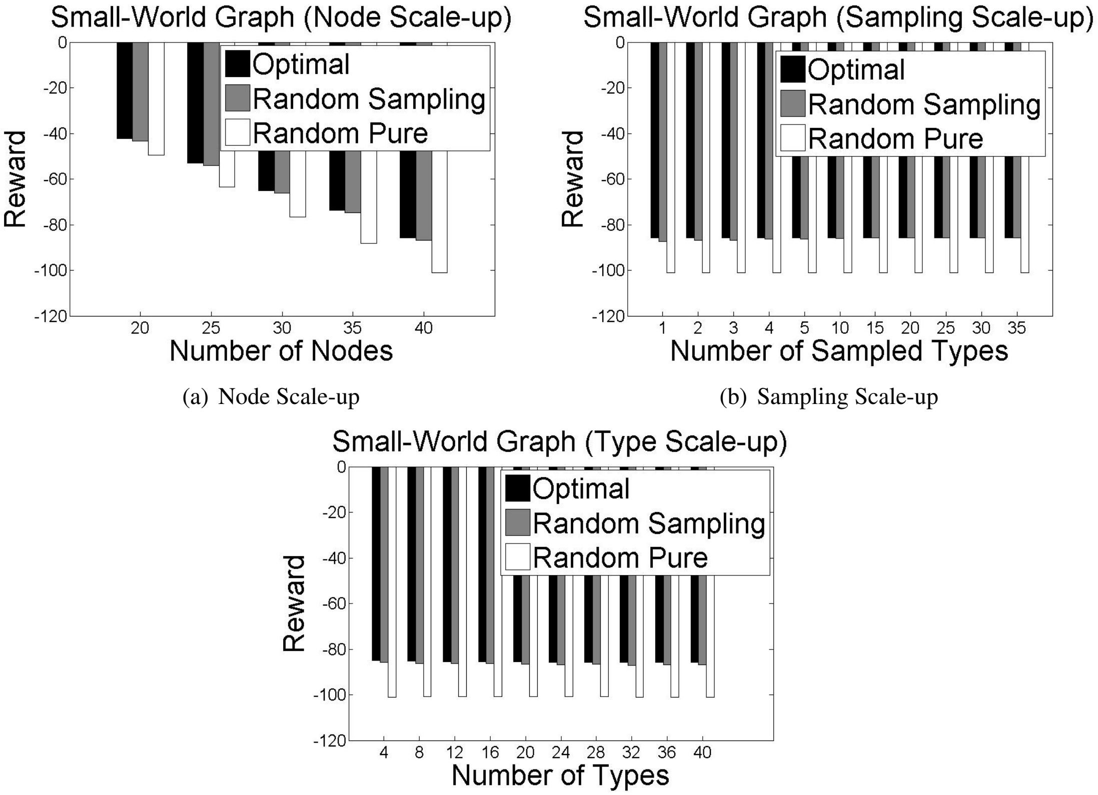


(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

Figure B.5: Scale-Free graph results for standard Influential Node Uncertainty experiments

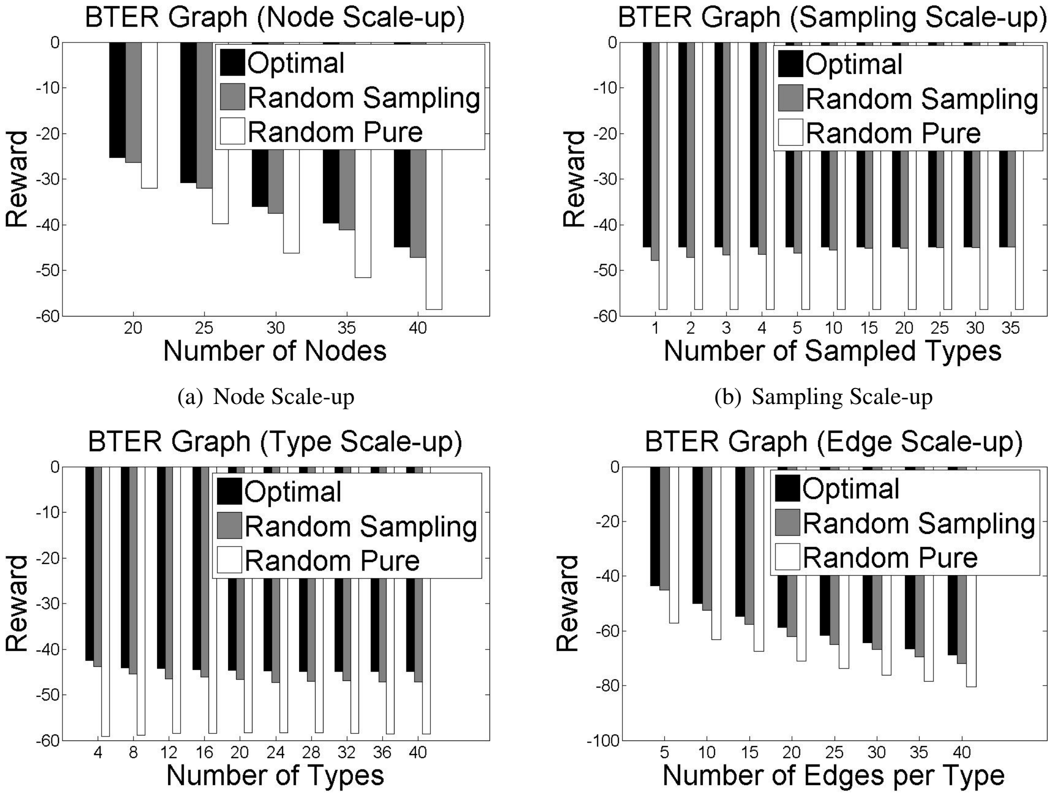


(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

Figure B.6: Small-World graph results for standard Influential Node Uncertainty experiments



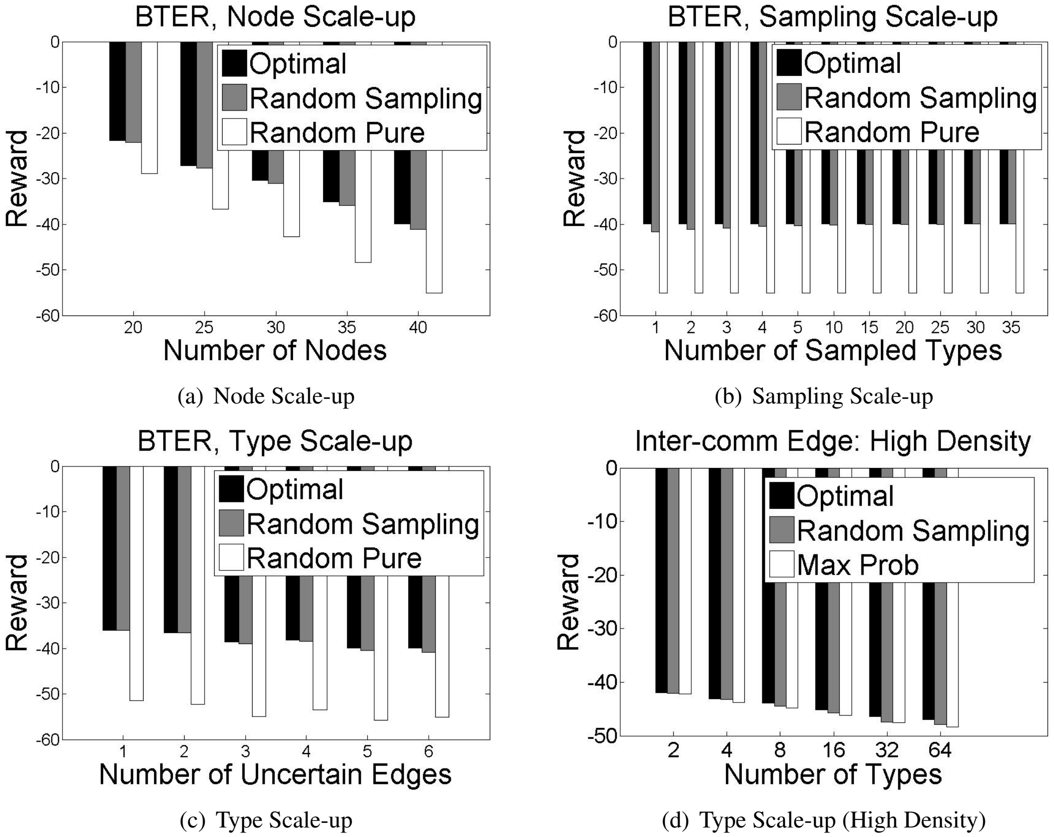
(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

(d) Edge Scale-up

Figure B.7: Results for standard Inter-Community Edge Set Uncertainty experiments



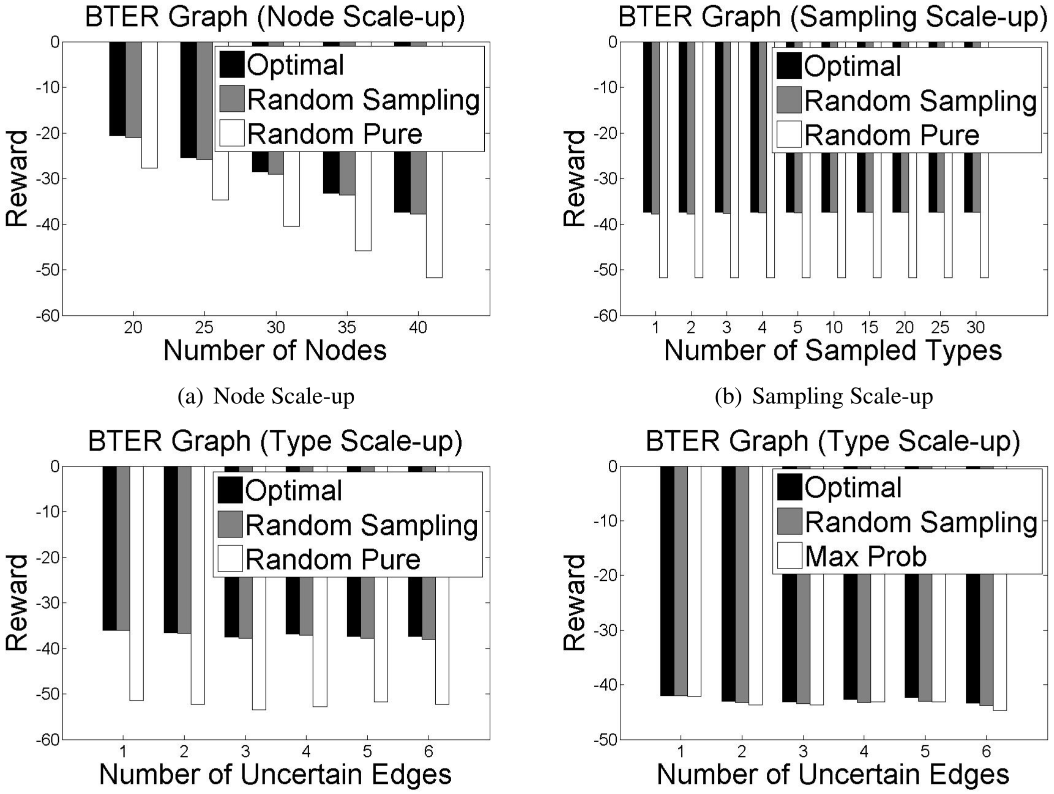
(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

(d) Type Scale-up (High Density)

Figure B.8: Results for standard Inter-Community Edge Uncertainty experiments, also includes BTER graph with high community density



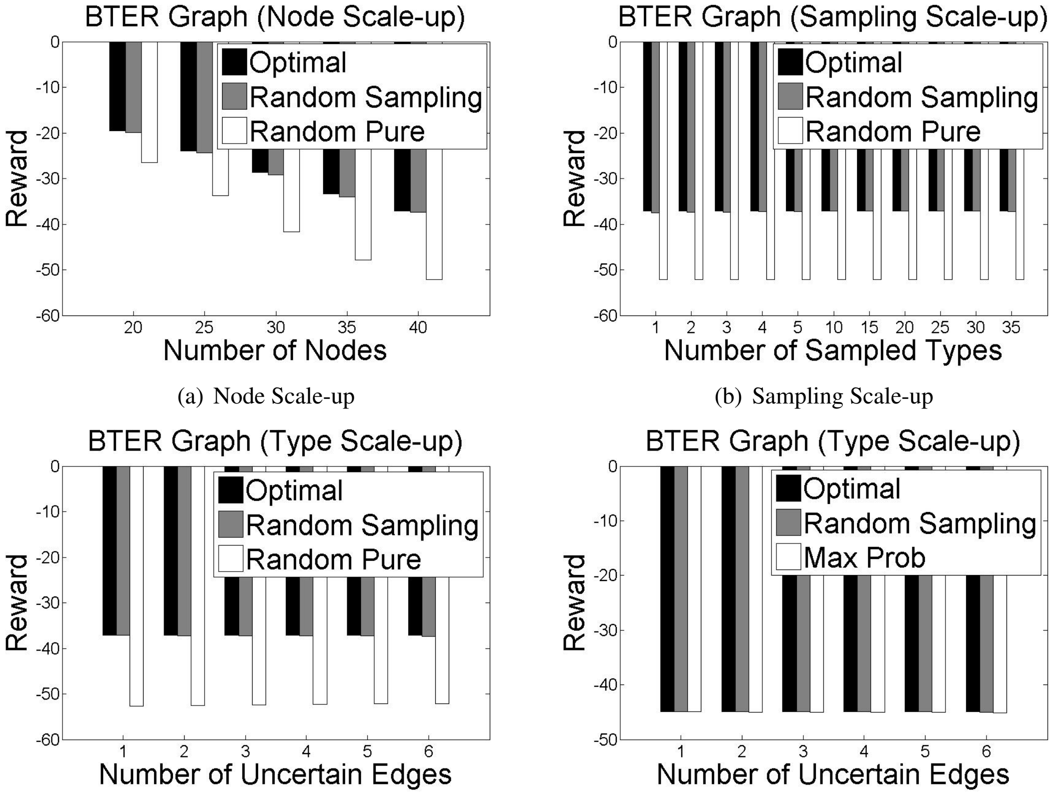
(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

(d) Type Scale-up (High Density)

Figure B.9: Results for standard Inter/Intra-Community Edge Uncertainty experiments, also includes BTER graph with high community density



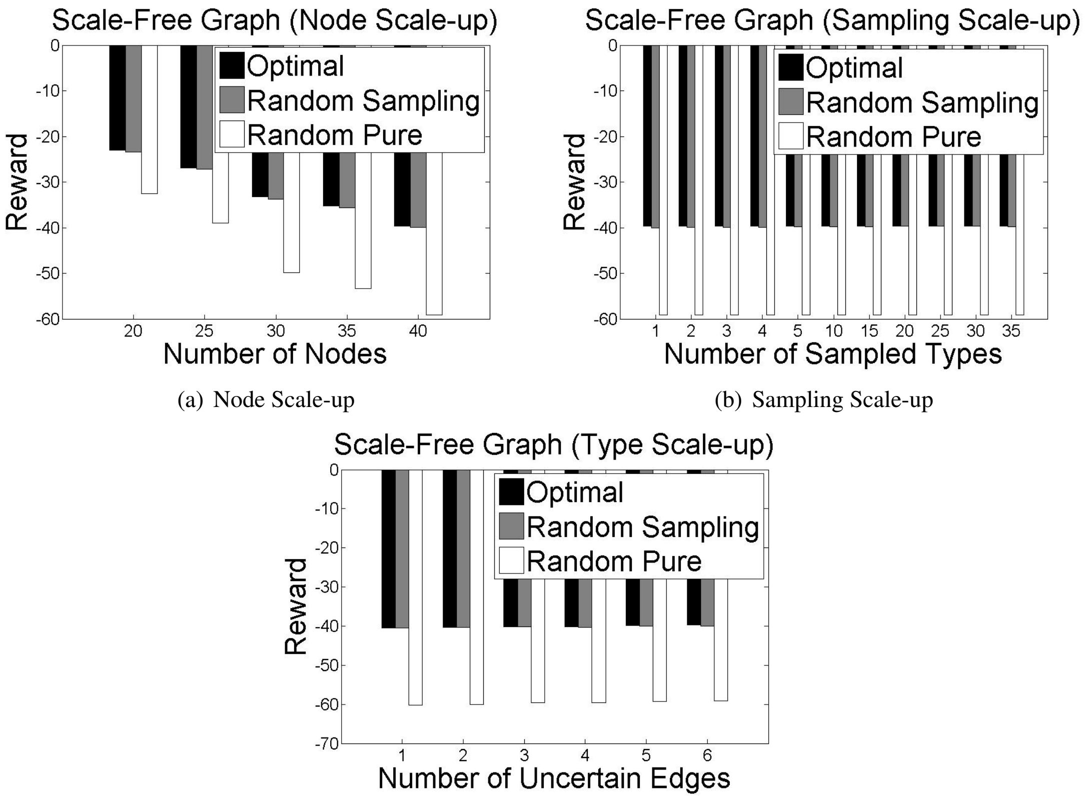
(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

(d) Type Scale-up (High Density)

Figure B.10: BTER graph results for standard Random Edge Uncertainty experiments, also includes BTER graph with high community density

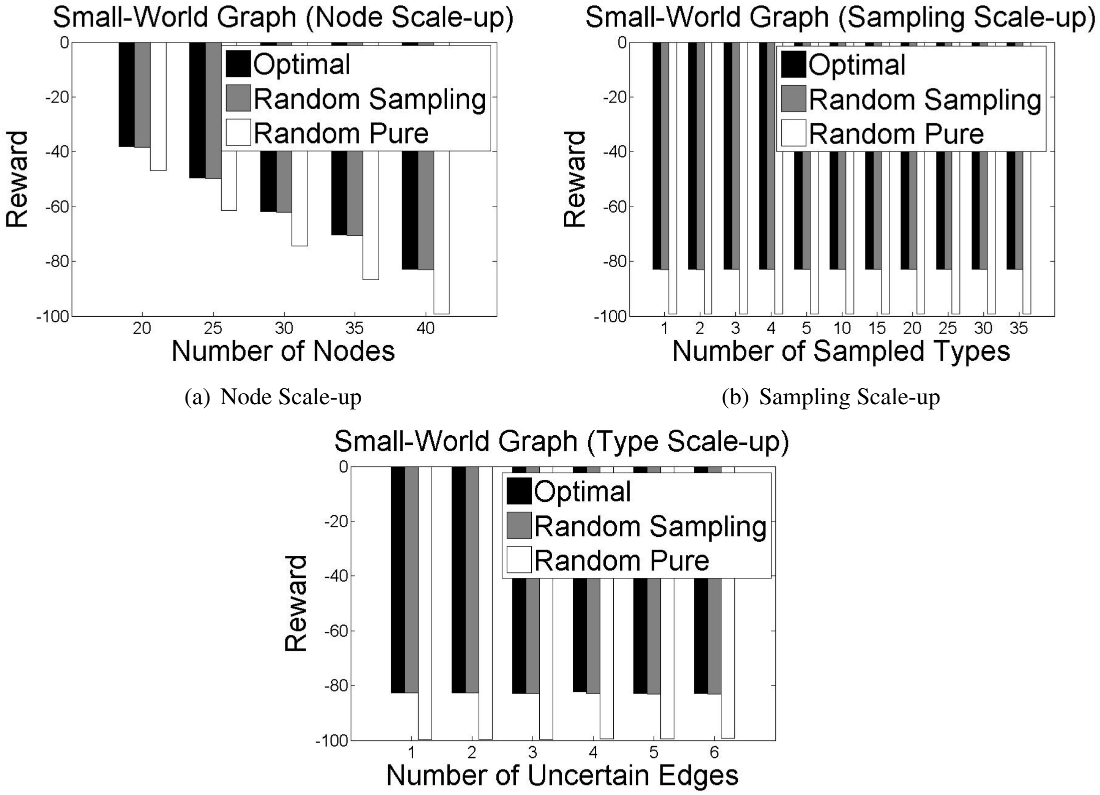


(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

Figure B.11: Scale-Free graph results for standard Random Edge Uncertainty experiments

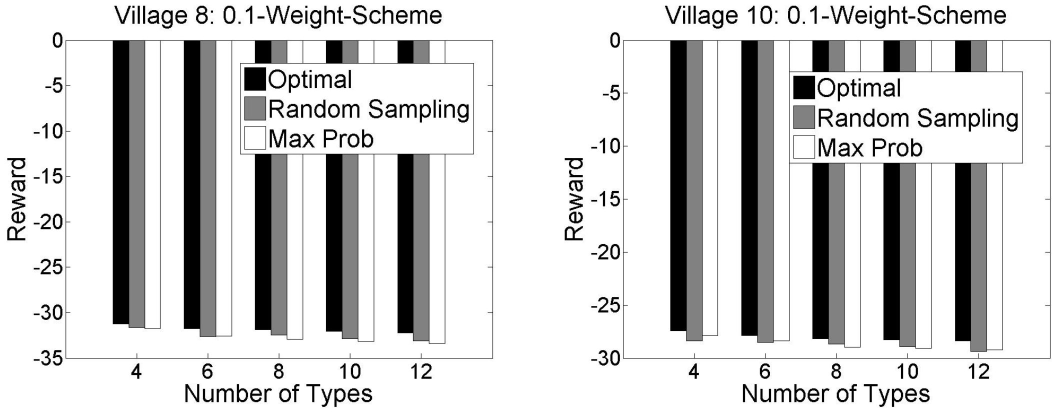


(a) Node Scale-up

(b) Sampling Scale-up

(c) Type Scale-up

Figure B.12: Small-World graph esults for standard Random Edge Uncertainty experiments



(a) Village 8

(b) Village 10

Figure B.13: India Village results with uniform weights