Containing Cascading Failures in Networks: Applications to Epidemics and Cybersecurity

Sudip Saha

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science And Application

Anil S. Vullikanti, Chair
Madhav V. Marathe
Naren Ramakrishnan
Rajmohan Rajaraman
Abhijin Adiga
Manish Jain

Jun 15, 2016
Blacksburg, Virginia

Keywords: failure cascades, epidemic containment, stealthy diffusion, game theory, attack graph

Containing Cascading Failures in Networks: Applications to Epidemics and Cybersecurity

Sudip Saha

(ABSTRACT)

Many real-word networks exhibit cascading phenomena, e.g., disease outbreaks in social contact networks, malware propagation in computer networks, and failures in cyber-physical systems such as power grids. As they grow in size and complexity, their security becomes increasingly important. In this thesis, we address the problems of controlling cascading failures in various network settings. We target two categories of cascading phenomena: natural (e.g., disease outbreaks) and malicious (e.g., cyber attacks). We consider the nodes of a network as being individually or collectively controlled by self-interested autonomous agents, and study their strategic decisions in the presence of these failure cascades. There are many models of cascading failures which specify how a node would fail when some neighbors have failed, such as: (i) epidemic spread models, in which the cascading can be viewed as a natural and stochastic process and (ii) cyber attack models, where the cascade is driven by malicious intent. We present our analyses and algorithms for these models in two parts.

Part I focuses on problems of controlling epidemic spread. In particular, we consider the SIS model on networks. Epidemic outbreaks are generally modeled as stochastic diffusion processes such as SIS and SIR. While existing literature provides heuristic centralized approaches for containing epidemic spread in SIS/SIR models, no rigorous performance bounds are known for these approaches. We develop algorithms with provable approximation guarantees that involve either protective intervention (e.g., vaccination) or link removal (e.g., unfriending). Our approach relies on the characterization of the SIS model in terms of the spectral radius of the network. The centralized approaches, however, are sometimes not feasible in practice. For example, targeted vaccination is often not feasible because of limited compliance to directives. This issue has been addressed in the literature by formulating game theoretic models for the containment of epidemic spread. However these studies generally assume simplistic propagation models or homogeneous network structures. In contrast, we develop novel game formulations which rely on the spectral characterization of the SIS model. In these formulations, the failures start from a random set of nodes and propagate through the network links. Each node acts as a self-interested agent and makes strategic intervention decisions (e.g., taking vaccination), and each agent decides its strategy to optimize its payoff (modeled by some payoff function). We analyze the complexity of finding Nash equilibria (NE) and study the structure of NE for different networks in these game settings.

Part II focuses on malware spread in networks. In cybersecurity literature, malware spreads are often studied in the framework of “attack graph” models. In these models, a node represents either a physical computing unit or a network configuration, and an edge represents a physical or logical vulnerability dependency. A node becomes compromised if a certain set of its neighbors are compromised. Attack graphs describe explicit scenarios in which a single vulnerability exploitation cascades further into the network, exploiting inherent dependencies among the network components. Attack graphs are used for studying cascading effects in many cybersecurity applications, e.g., component failure in enterprise networks, botnet spreads, advanced persistent attacks. One distinct feature of cyber attack cascades is the stealthy nature of the attack moves. Also, cyber attacks are generally repeated. Controlling stealthy and repeated attack cascades poses an interesting problem. Van Dijk et al. [119]

first proposed a game framework called “FlipIt” for reasoning about the stealthy interaction between a defender and an attacker over the control of a system resource. However, in cybersecurity applications, systems generally consist of multiple resources connected by a network. Therefore, it is imperative to study stealthy attack and defense strategies in networked systems. We develop a generalized framework called “FlipNet” which extends the work of Van Dijk et al. [119] for networks and we present analyses and algorithms for different problems in this framework. However, if the security of a system is limited to the vulnerabilities and exploitations that are known to the security community, often the objective of the system owner is to take cost-effective steps to minimize potential damage in the network. This problem has been formulated in the cybersecurity literature as hardening attack graphs. Several heuristic approaches have been shown in the literature so far, but no algorithmic analyses have been shown. We analyze the inherent vulnerability of the network and present approximation hardening algorithms.

Contents

1 Introduction **1**

1.1 Cascading effects in networks 1

1.2 Controlling cascades in networks 2

1.3 A unifying framework for cascades and approaches for their control 2

1.4 Limitations of the prior work 3

1.5 Cascading as epidemic spread 5

1.6 Cascading failures in attack graphs 8

2 A Unifying Framework for Cascades and Containment Approaches **12**

2.1 Graph Dynamical System (GDS) 12

2.2 SIS and attack graph models as instances of GDS 13

2.3 Approaches for minimizing cascades 14

2.3.1 Epidemic Containment: Making Topological Changes in GDS 15

2.3.2 Containing Cyberattack Cascades: Adjusting Local Node Functions in GDS 16

2.3.3 Comparison Between the Two Approaches 16

3 Related Work **19**

3.1 Epidemic Spread 19

3.1.1 Algorithms and Game Models for Epidemic Control 20

3.2 Cyber-attack Cascades 21

3.2.1 Attack Graph Models and Methods 21

3.2.2 Game Models for Stealthy and Persistent Cyber Attacks 22

I Epidemic Containment in SIS Model 24

4 Introduction 25

5 Epidemic Containment with Decentralized Strategies 27

5.1 Preliminaries and Epidemic Containment Game Model 29

5.2 Existence and complexity of Nash Equilibria 30

5.3 The structure of NE in general graphs 33

5.4 The structure of NE in random graph models 34

5.4.1 The Erdős-Rényi model 34

5.4.2 Random power law graphs 35

5.5 Empirical results 39

5.5.1 Estimates of the minimum and maximum NE cost 40

5.5.2 Estimates of the Price of Anarchy (PoA) 41

5.5.3 Degree Distribution in random NE 41

5.5.4 Impact of Stackelberg strategies 42

5.5.5 Analyis of the community structure in NE 42

5.6 Conclusion 44

6 Epidemic Containment with Centralized Strategies 45

6.1 Preliminaries 46

6.2 GreedyWalk: $O(\log n \log \Delta)$-approximation 48

6.2.1 GreedyWalk in small networks using matrix multiplication 49

6.2.2 GreedyWalk in large sparse networks using a dynamic programming approach 49

6.3 Using sparsification for faster running time: Algorithm GreedyWalkSparse 50

6.4 Primal-Dual: $O(\log n)$-approximation 52

6.5 Extensions 54

6.5.1 Non-uniform transmission rates 54

6.5.2 Node version: SRMN problem 56

6.6 Popular heuristics and lower bounds 56

6.7 Experiments 60

6.7.1 Methods and Dataset 60

6.7.2 Experimental results 60

6.8 Conclusions 63

7 Comparison among Different Epidemic Thresholds 65

7.1 Epidemic Parameters 65

7.1.1 Basic reproduction number, R_0 65

7.1.2 Spectral radius based epidemic parameter, R_ρ 66

7.1.3 Critical transmissibility based epidemic parameter, R_T 66

7.2 Comparison among the epidemic parameters 68

7.3 Conclusions 68

II Containing Cascading Failures in Attack Graphs 69

8 Introduction 70

9 FlipNet: Modeling Covert and Persistent Attacks on Networked Resources 71

9.1 Introduction 72

9.2 Formal Definitions and Notations 73

9.3 The complexity of finding optimal strategies 78

9.4 Existence of Nash equilibria 78

9.5 Results on optimizing attacker's gain 79

9.6 Computing gains 87

9.7 Extension: FlipNetUndir 89

9.7.1 FlipNetUndir 89

9.7.2 FLIPNETCUMULATIVE 90
9.8 Experimental Results 92
9.8.1 Dominant Strategies and Nash Equilibria 94
9.8.2 Critical Strategies 95
9.8.3 Best Response Strategy of the Defender 97
9.8.4 Distribution of Attacker Budget 99
9.9 Conclusion 100

10 Vulnerability Analysis and Hardening Solutions 104
10.1 Preliminaries 104
10.1.1 Characterizing the inherent vulnerability of the system 106
10.1.2 Network hardening by the defender 107
10.2 The MAXATTACKERUTIL problem 107
10.2.1 Greedy Solution for OR-only decompositions 108
10.3 The DEFENDERHARDENING problem 109
10.3.1 Algorithm for DEFHARD for constant k 110
10.3.2 Algorithm for DEFHARD for large k 111
10.4 Empirical Results 113
10.5 Conclusion 116

III Conclusion 117

11 Conclusion 118

12 Bibliography 119

Chapter 1

Introduction

1.1 Cascading effects in networks

Networks arise as underlying structures in many dynamical systems. One common example is the spread of epidemics, a phenomenon which involves social contact networks. Disease infections, such as the flu virus, spread in a population through the social contacts that people make among themselves. Therefore, for proper understanding of disease dynamics, one has to consider a realistic model of the underlying social contact network where nodes represent the individuals and the edges represent their pairwise contacts [35]. Networks also have use in many cybersecurity applications, which involve different forms of networks. For example, the "mass-mailing worms" such as Melissa and Sircam [111] propagate from one email-id to the others while exploiting the contact lists. Therefore, a network of email-ids can be considered here where nodes represent the email ids and the edges represent the email communications. Network structures also exist inherently in infrastructure systems, such as electricity power grids. The electricity transmission and distribution system or the power grid can be considered a network, where a node represents a substation or transformer or a consuming unit and an edge represents a physical cable that connects two nodes [91]. Other examples include cascading of financial crises in financial networks [34] [4], botnet spread in the Internet [29], malware spread and data leak attacks in enterprise networks [97] [41], rumor spreads on Facebook [37] etc.

As networks grow in size, the risk of failure is increased. We use the word "failure" as a general term to denote undesirable events in different contexts, which spread through network connections in different ways. For instance, disease infection is a common example of failure that spreads through social contact networks. In the Internet, when a router goes down, it can potentially cause traffic overloading in other routers, which can result in their failure and create a cascading effect [25] (the earliest reported incident of such congestion collapse happened at Berkeley, CA in 1986 [48]). In this case, a router being down refers to a failure. In power grids, overloading a single element can cause spikes in large number of other elements resulting in a power outage [47] (e.g., northeast blackout of 2003 [91], 2012 blackout in India [103]). Similarly, many recent cyber breaching incidents of protected computer networks start when a single machine is compromised by phishing or similar attacks, which is then followed by the compromise of other machines through insider links. Failure, in this case, refers to the compromise of a machine. Some of these network failures are human-initiated and actively managed, such as cyber attacks, while others occur naturally without human initiation, such as disease outbreaks. Irrespective of the contexts and the mechanism, cascading of failures in networks in one form or another is very common.

1.2 Controlling cascades in networks

The problem of managing cascades has been a subject of considerable research interests for its applications to many scenarios. For example, the adoption of new products or technologies is generally affected by "word-of-mouth" effects or viral marketing campaigns. This has led to the study of maximizing cascading effects by Domingos and Richardson [32], Kemple et al. [54] among others. On the other hand, the problem of minimizing the effects of cascades has been studied for its applications to epidemic control, cybersecurity, infrastructure security, etc [77, 35, 40, 95]. In general, these prior works provide insights and propose heuristic centralized approaches for cascade minimization. However these approaches share the following limitations - (i) they make simplistic model assumptions, (ii) the solutions are centralized in nature, (iii) no bounds on the performance of the solutions have been presented. One particular model assumption that many of the prior works have made is that of homogeneous mixing among the entities in the system; that is, cascade minimization in networks is not a well studied problem in the literature. In this thesis, we addresses these limitations. Particularly we study SIS and attack graph models, which are two widely used cascade models in the area of epidemics and cybersecurity respectively. This thesis systematically investigates the problem of minimizing cascading effects in networks in these two models.

1.3 A unifying framework for cascades and approaches for their control

The dynamics of failure spread in networks takes different forms in different application areas. As a result, there exist different propagation models that capture the cascading effects in various network applications. For example, disease outbreaks, rumor spread and certain kinds of malware propagation are commonly modeled by stochastic diffusion models such as SIS and SIR [6, 11]. Cyber attack propagations are often studied in the framework of attack graph models which describe the explicit scenarios that lead to attack cascades [87, 30]. A unifying framework that has been used to describe these models for cascades in networks is called the Graph Dynamical System (GDS) [60]. GDS involves a state space for the nodes; each node is in one of the states of that state space. At each time step (in a discrete or continuous time span) the state of a node is updated according to a local function associated with the node. Both stochastic diffusion models (e.g., SIS) and attack graph models can be viewed as special cases of the general GDS model. We will discuss this further in chapter 2.

Many studies involving GDS are concerned with how to control the dynamics of state changes of the nodes in GDS. In this thesis we address one central aspect - containing cascades of various forms. Specifically, we study epidemic spread and cyber-attack cascades. The dynamics of node state changes in GDS can be controlled mainly in two ways: either by making changes to the network topology, or modifying the local node functions. The first approach is more relevant in the epidemic containment problem, while the second approach has more applications in the cybersecurity domain. Controlling epidemic spread typically involves vaccinating key individuals, quarantining infected individuals etc., which effectively corresponds to node and edge removal efficiently. Our epidemic containment study in Part I relies on this approach, and we have studied decentralized and centralized node intervention and edge intervention solutions. In cybersecurity applications, on the other hand, system administrators or defenders generally keep updating or modifying security rules to guard against potential security threats. This corresponds to the local function change approach in GDS. We take this approach in Part II in developing solutions for containing cyber-attack cascades.

1.4 Limitations of the prior work

There is a significant volume of existing literature which centers on modeling and analyzing various cascading effects in networks, including information cascades [42, 43], tweets spread in the twitter network [68], marketing and product penetration [102], blogs and propagations [46, 65]. Characterizing the size and persistence of the cascade has been an active line of research, which includes finding epidemic thresholds [92, 122, 40, 95]. In an all-to-all connected network, differential equation-based models provide a well known epidemic threshold parameter called the "basic reproduction number", R_0, which describes the strength of an epidemic. But, no epidemic threshold has been established for general network models; however, the persistence of an outbreak has been shown to be characterized by the spectral radius of the network [40, 95]. These epidemic summary parameters provide useful information for controlling epidemics.

Making use of the cascading effects in networks has been a topic of interest to many for its value in viral marketing campaigns in social networks. Domingos and Richardson first posed the problem of maximizing cascades by making the optimal marketing action - selecting customers for product discounts that would result in the maximum "word-of-mouth" effect. In their work, the cascading process, i.e., the social influence in product purchasing has been modeled as a Markov random field where the probability of product adoption depends on the adoption of others, as well as the initial marketing action. Kempe et al. [53] considered a more realistic influence model in which the influence spreads according to the independent cascade model. They analyze the number of individuals who are influenced as a function of the set of "seed" individuals who initiate the influence propagation. They show that this function has submodularity property, and a greedy algorithm provides a solution which is provably within $(1 - \frac{1}{e})$ of the optimal. This work shows the first provable algorithm for maximizing network cascades. A drawback of this work is that it uses expensive Monte Carlo methods to estimate the influence function. The efficiency of computing the influence function has been further improved by the work of Leskovec et al. [72] and Chen et al. [22].

Although the problem of maximizing network cascades has been addressed in sufficient depth, as discussed above, the opposite problem of minimizing cascades has received limited attention. However, it has been studied in the public health literature in terms of the optimal allocation of limited number of vaccines in the wake of an epidemic. A number of studies have addressed this problem of controlling epidemic spreads by using epidemic characterizations and other heuristics. For instance, Medlock and Galvani [77] described cost-effective strategies to vaccinate different age groups based on their transmission dynamics. Vaccinating high degree nodes, i.e., highly connected individuals have been shown to be an effective strategy in the work of Eubank et al. [35] and Pastor-Satorras and Vespignani [9]. While these approaches rely on the global degree distribution information, Cohen et al. [26] have presented a strategy that involves local information: that is, vaccinating random acquaintances of a randomly selected individual. Ganesh et al. [40] and Prakash et al. [95] have shown the spectral characterization of epidemic spread in SIS/SIR models, which has led to the following spectral radius based approaches for epidemic containment [115, 79, 96, 116]: choose the optimal set of individuals to vaccinate or optimal set of contacts to remove, so that the spectral radius of the network falls below a critical threshold. They present several heuristic approaches based on the node degree, principle eigenvector and node page rank, among others. However, no known strategy has attained provable performance bounds. In this thesis, we address this limitation and present algorithms with provable performance bounds for containing epidemics by reducing the spectral radius below the critical threshold.

The tactics discussed above are all centralized approaches. However, applying centralized approaches is not always feasible. For example, selecting a subset of individuals for vaccination does not always work due to limited compliance to directives. This complication provides a motive to understand decentralized approaches - particularly within game theoretic models. In these models, each node in the network has to determine its own strategy of securing itself or not, as the node's benefit depends on which other nodes are secure; this is a natural game-theoretic setting. While there has been a lot of work on network security game models, most of the focus has been either on simplified epidemic models [7][66] or homogeneous network structures [12, 38, 100, 56]. In this thesis, we address this limitation and develop a novel formulation of an epidemic containment game, which relies on the characterization of the SIS model in terms of the spectral radius of the network.

Although in some instances malware spreads as an epidemic (e.g., mobile malware [76]), malware proliferation commonly presents a different dynamics. The containment of malware propagation has commonly been studied with the “attack graph” framework in the cybersecurity literature. Attack graphs generally represent logical or physical vulnerability dependencies among the various network components in a network [30, 99, 120]. One major limitation of most prior work is that, it assumes complete information on part of the defender and the attacker. But in practice, major cyber attacks involve zero-day vulnerabilities, which are not known apriori to the security community, e.g., Stuxnet [36], Aurora [71]. Therefore, covertness is a distinct and important feature of cyber attak cascades. Wang et al. [121] have proposed a model called “zero-day attack graph” to assess the vulnerability of a networked system in the face of zero-day attacks. Van Dijk et al. [119] have proposed a game framework called “FlipIt” to reason about the covert and persistent interaction between an attacker and a defender over the control of a resource. “FlipIt” was analyzed for a single resource, which has been extended for multiple resources in the work of Laszka et al. [69]. However, covert and persistent attacks have often been found to be directed towards sensitive government and corporate networks. Therefore, it is imperative to study such models for covert and persistent cyber attacks and defensive measures in networked systems. In part II of this thesis, we study this problem.

In addressing known vulnerabilities and attacks, researchers have studied the control of cyber attacks as the “hardening” problem of attack graphs. Wang et al. [120] have formulated the hardening problem as a satisfiability problem, while Dewri et al. [30] have used evolutionary algorithms to find sub-optimal hardening solutions. Although heuristic and evolutionary approaches have been presented in the literature, no formal study has discussed the algorithmic underpinnings of the hardening problem. In this thesis we have presented approximation algorithms for hardening attack graphs.

1.5 Cascading as epidemic spread

The spread of epidemics and malware is commonly modeled by diffusion processes, such as the SIS/SIR models [83, 44] on a network, in which the infection spreads independently from an infected node to its neighbors. A typical method of controlling their spread is to vaccinate nodes or install antivirus software patches. This involves a certain cost for the individual (e.g., the economic cost of securing or buying an antivirus software, and the additional inconvenience). On the other hand, if all the neighbors or contacts of an individual are protected, there is no need for the individual to protect itself (a notion termed as “herd immunity”). An alternative approach is to remove some social contacts or delete edges so that an infection or malware becomes less likely to spread. However, in many settings, nodes also want to form links with others to receive potential benefits.

The above scenarios present a natural setting for game-theoretical analysis, and much research centered on the use of non-cooperative game models for network security and the control of epidemics, e.g., the works in [8, 7, 67, 12, 89, 70, 45, 57, 55, 56]. All of these models involve individual utility functions with some notion of the cost of becoming infected. Computing and analyzing this utility function is difficult in general for heterogeneous networks. A large part of the prior research has addressed this by making simplistic assumptions about either the diffusion process (e.g., a very high transmission probability), or about the network (e.g., homogeneity, allowing for differential equation models). Understanding the dynamics of epidemic games in realistic scenarios is a fundamental open problem.

In this thesis, we consider topological properties of the contact network in studying epidemic games. In an important result that highlights the impact of network structure on the dynamics of epidemic spread, Wang et al [123] and Ganesh et al. [40] have developed a characterization for the dynamics of the SIS model in terms of the spectral properties of the adjacency matrix; their results show that an epidemic in the SIS model dies out soon if $\lambda_1(G) < \gamma/\alpha$, where $\lambda_1(G)$ is the *spectral radius* or the *largest eigenvalue* of the adjacency matrix of the contact graph G, and γ and α are the recovery rate and transmission rate of the SIS model, respectively. We use T to denote the threshold γ/α. This result is extended to other models by Prakash et al. [95], and is analogous to the characterization of differential equations based epidemic models in terms of the reproductive number, R_0 [83]. In general, however, the characterization is not tight, and the converse is not true; instead, Ganesh et al. [40] show that the epidemic persists for a “long” time if the difference between the first and the second eigenvalues of the laplacian of G is bounded. However, for many kinds of graphs, the condition based on spectral radius is quite close. This characterization has motivated the strategy of containing an epidemic by reducing the spectral radius [115, 79]— for differential equations based models, the corresponding strategy is to influence parameters so that R_0 reduces [83]. We build on this approach and study this problem from a game-theoretical perspective.

A natural approach for containing an epidemic, motivated by this characterization, is to design interventions so that λ_1 becomes smaller than T, as discussed in [115, 79] (for differential equation based models, the corresponding strategy is to influence parameters so that R_0 reduces [83]). Two of the common interventions are to vaccinate/secure some nodes (which corresponds to deleting them from the network) and remove links (e.g., social distancing). We study these intervention strategies in a game theoretic setting, where nodes represent autonomous agents. Each agent trades off the benefit of vaccinating a node against the cost of vaccination. Similarly, he forms a new link or removes an existing link, taking into consideration the potential benefits and risks of infection due to the link and the cost of removing it. We address the different aspects of this scenario in the chapters of part I. First, we introduce a game theoretic formulation (*Epidemic Containment (EC) game*) in chapter 5 to study the epidemic control using vaccination. In chapter 6 we develop approximation algorithms for computing the social optimum in *EC* game. Below, we give a brief overview of the chapters and highlight our main contributions.

Epidemic control as a vaccination Game (chapter 5): We introduce the *Epidemic Containment (EC)* game, a vaccination game formulation which is based on the spectral characterization of SIS model on networks [40]. We assume that the agents have an underlying contact graph. In the presence of the risk of epidemic outbreak of failure, each agent either vaccinates or enjoys "herd immunity" due to effective vaccination of its neighboring agents. We study the cost of strategic node vaccination in Nash Equilibria (NE) and the price of anarchy or PoA (the ratio of the cost of the worst NE to the social optimum cost) for different network structures, such as arbitrary power law graphs, Erdos-Renyi random graphs and Chung-Lu power law graphs. Specifically, our contributions are as follows.

- *A game formulation based on spectral properties:* We introduce the *Epidemic Containment (EC)* game on a network $G = (V, E)$ of players, for the SIS model of epidemic spread.
- *Structure of equilibria in arbitrary graphs:* We derive bounds on the cost of the worst NE and the PoA in terms of the maximum degree in general graphs. When G has a power law degree sequence with exponent $\beta > 2$, we show that the cost of the social optimum $\in [c_1 n/T^{2(\beta-1)}, c_2 n/T^{(\beta-1)}]$, for constants c_1, c_2; this implies the PoA is $O(T^{2(\beta-1)})$.
- *Structure of equilibria in random graph models:* We study the structure of NE in the Erdős-Rényi and Chung-Lu random graph models. We prove that in Erdős-Rényi graphs $G(n, p)$, if $p = \Omega(\log n/n)$ (which is needed for the graph to be connected), and if $T^2 = O(np)$, every NE has cost $\Omega(n)$, and the PoA is $\Theta(1)$.
- *Empirical analysis of the properties of the equilibria:* We study the structure of equilibria in EC games in different real and random networks where failure could spread.

Social optimum cost for epidemic containment (chapter 6):

We develop approximation algorithms for computing the social optimum cost in the *EC* game, introduced in the previous chapter. We formulate an optimization problem, (SRMN), which corresponds to reducing the spectral radius by deleting nodes. A similar formulation is SRME problem, which is the edge deletion version of SRMN. We show provable bounds for the solutions of our algorithms in different networks. Specifically, our contributions are as follows:

- We present a bi-criteria approximation algorithm, GreedyWalk for finding the optimal cost of edge quarantining which achieves $O(\log n \log \Delta)$ approximation ratio while ensuring that the spectral radius becomes at most $(1+\epsilon)$ times the epidemic threshold; here Δ denotes the maximum node degree and ϵ arbitrarily a small constant.
- We propose a faster variant of GreedyWalk, called GreedyWalkSparse, that performs careful sparsification of the graph and achieves a better running time while retaining similar asymptotic guarantee.

- We present another algorithm, PrimalDual, which achieves a better approximation guarantee $O(\log n)$ while at the expense of higher running time.
- We show that these algorithms apply to an extension of the problem where failure transmission rate is non-uniform.
- We analyze popular heuristic algorithms - ProductDegree, EigenScore, PageRank and show that they can perform quite poorly in general.
- We compare the solutions produced by our algorithms GreedyWalk and PrimalDual with solutions by some of the popular heuristic algorithms on large real networks which reveals the stronger performance of our algorithms.

1.6 Cascading failures in attack graphs

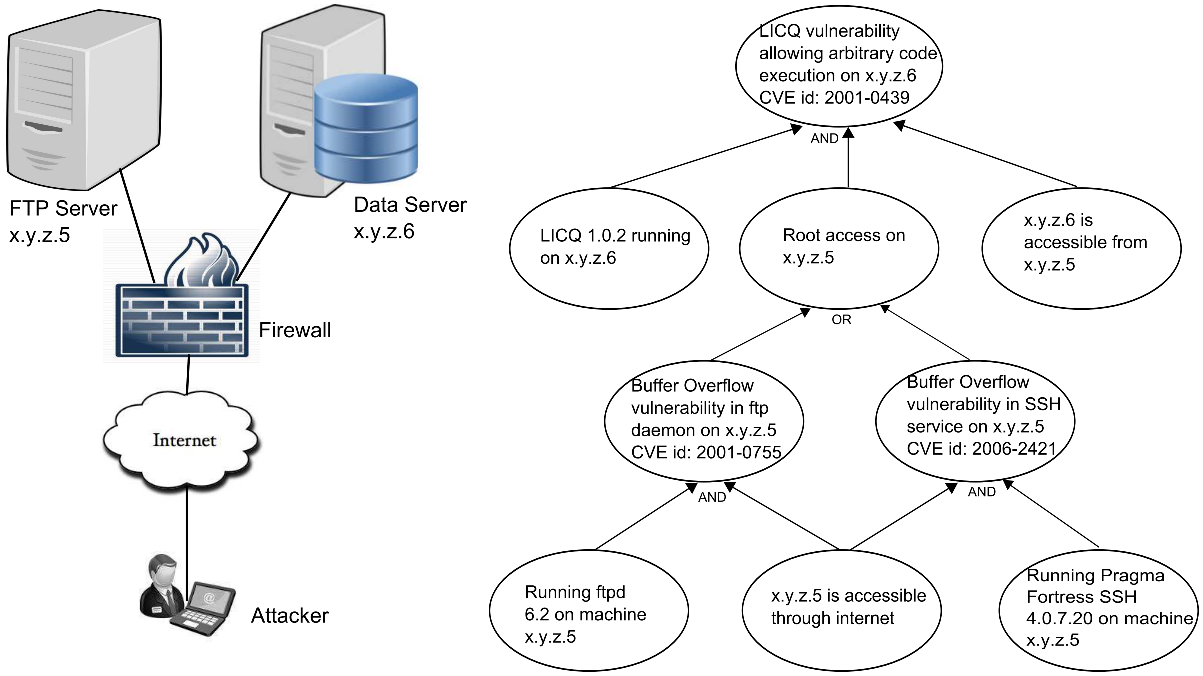
In many network security problem domains, e.g., enterprise networks, smart grid, etc., the spread of failures can be modeled using attack graph models. Explicit vulnerabilities and their inter-dependencies trigger the cascade of failures. Prior work in the network security literature have introduced the “attack graph” [88], [109], [93], [50], [5] and “attack tree” [99], [30] models, which capture these interdependencies. Specifically, these models capture the scenarios and their interconnections that can lead to damage, allowing network designers and attackers to understand overall vulnerabilities and develop attack/defense methods. Attack graphs have been described in different ways across existing literature; we consider attack graphs as directed acyclic graphs, in which nodes represent either physical computing machines or vulnerabilities in the form of network configurations (e.g., open ports or unsafe firewall configurations). The source nodes (also called leaf nodes) represent intrinsic or low level vulnerabilities, which have no dependencies. An internal node represents a vulnerability that occurs after some/all the descendant nodes have been exploited. Attacks start from source or leaf nodes and propagate through the edges exploiting the corresponding vulnerabilities.

In the attack graph framework, we consider two models in which attacks propagate - (1) repeated stochastic attack propagation and (2) one-time deterministic cascade.

In the first model of repeated stochastic propagation, the attacks start from the source nodes and repeatedly propagate to the other nodes through the edges repeatedly according to a stochastic process. That is, a node is repeatedly attacked in a stochastic manner from a compromised node. This model of attack propagation is very relevant in cybersecurity applications since it captures a variety of scenarios, especially covert attacks involving advanced persistent threats [16]. Such covert and repeated attacks has been studied in the literature for single and multiple resources by Van Dijk et al. [119] and Laszka et al. [69] respectively. However, no such study has been done for networked resources. Therefore, it is a natural extension of previous literature to study covert and persistent attacks in networks, since many recent advanced attacks involve networks, e.g., Stuxnet, Rsa hack of 2011. We consider the network as represented by an attack graph. In studying covert and persistent attacks, the attack graphs can be considered in both the forms. The attack graph can represent the actual physical network of computing machines which exemplify cases such as the RSA hack of 2011. In this case, an attacker repeatedly mounts attacks along an edge or physical link, exploiting different or newer vulnerabilities. The attack graph can also refer to cases in which the nodes represent vulnerabilities or network configurations, e.g., the "zero-day attack graph" presented by Wang et al. [121]. In this attack graph, the nodes in the attack graph represents vulnerabilities or network configurations. The edges are modeled by the defender as possible zero-day exploitations, the specifics of which are not known to the defender.

In the second model of one-time deterministic cascade, the attacker chooses a subset of source nodes to attack, which in turn causes a deterministic cascading attack in other nodes; this cascade follows according to the attack dependencies in the attack graph. This kind of attack cascade can be considered for both forms of attack graphs. This cascade model naturally applies to attack graphs in which nodes represent physical computing machines, such as in the case of wireless networks. However, hardening is more relevant and has been discussed more in the literature for attack graphs in which the nodes represent vulnerabilities or network configurations. Such an attack graph is illustrated in figure 1.1. Figure 1.1b shows an example attack graph representation for the physical network in Figure 1.1a; the node labels illustrate different kinds of vulnerabilities. In this example, there are two machines in a Local Area Network (LAN)–one ftp and one data server–behind a firewall. The firewall allows a remote user to use only the ftp server; the data server is accessible only from the ftp server. The ftp server runs certain ftp daemon and ssh services that have buffer overflow vulnerabilities, e.g., among those documented in the CVE vulnerability database [1]. A remote user may exploit either of these vulnerabilities to get root access on the ftp server. This, in turn, allows the attacker to mount another attack on the data server, exploiting an arbitrary code execution vulnerability attached to its LICQ client, an online messaging client. In this way, a sequence of attacks can be mounted and can lead to the hacking of some network resources that are not directly accessible. Note that some vulnerabilities require all their preconditions to be met – this is denoted by an AND function at the node; at other nodes, only one precondition needs to hold for the vulnerability to occur– this is denoted by an OR function associated with that node. Each node is associated with a potential damage, which would occur if that node is exploited. The strategies for attackers and defenders correspond to the sets of leaf nodes to be attacked or secured; this determines which internal nodes are compromised. See chapter 10 for more details and a formal description of this model.

As mentioned earlier convert and persistent attacks have been previously studied for single and multiple resources, but not for networked resources. And, although there has been a number of prior works in hardening attack graphs, the literature lacks formal analyses regarding understanding of the overall network vulnerability and performance of network


FTP Server
x.y.z.5
Data Server
x.y.z.6
Firewall
Internet
Attacker
LICQ vulnerability allowing arbitrary code execution on x.y.z.6 CVE id: 2001-0439
AND
LICQ 1.0.2 running on x.y.z.6
Root access on x.y.z.5
x.y.z.6 is accessible from x.y.z.5
OR
Buffer Overflow vulnerability in ftp daemon on x.y.z.5 CVE id: 2001-0755
Buffer Overflow vulnerability in SSH service on x.y.z.5 CVE id: 2006-2421
AND
AND
Running ftpd 6.2 on machine x.y.z.5
x.y.z.5 is accessible through internet
Running Pragma Fortress SSH 4.0.7.20 on machine x.y.z.5

(a) Network

(b) Attack Graph

Figure 1.1: An example network and the corresponding attack graph: (i) Example network: the data center, which is not directly accessible to the user can be attacked through the ftp server. (ii) Attack graph representation for the vulnerabilities in the network.

hardening algorithms. In Part II, we address these problems and study the failure cascade problem using attack graph models. In chapter 9 we develop a novel game theoretical framework for studying covert and repeated attacks and defensive measures in networks and present algorithms and analyses. In chapter 10 we develop approximation algorithms for finding efficient network hardening solutions in attack graphs.

Modeling stealthy and persistent attack/defense in networks (chapter 9): We develop a game theoretic model for reasoning about the stealthy and persistent interaction of a defender and an attacker over the control of networked resources. This is a generalized form of the time game called "FlipIt" which has been studied for stealthy and persistent interactions over a single resource [119]. In chapter 9 we formalize the model, analyze complexities of finding optimal strategies, and present a near-optimal algorithm for finding the optimal strategy of the attacker limited by a budget. Our contributions are summarized below.

1. We present a generalized game model called "FlipNet" for stealthy and persistent interaction between a defender and an attacker over the control of networked resources.

2. We show that finding the optimal strategies in this game is NP-complete.

3. We show methods for computing utilities of the players in a variant of the game.
4. We prove the marginal diminishing return property of the attacker's gain given a fixed defender strategy; this gives a $O(1 - \frac{1}{e})$ approximation algorithm for computing the optimal attacker's strategy where the defender's strategy is fixed and the attacker's strategy is budget limited.
5. We discuss a connection between FlipNet and the SIS epidemic and show that the persistence of the attacks in a variant of FlipNet can be characterized by the spectral radius of the graph.

Vulnerability analysis and hardening solutions (chapter 10): We use attack graphs to understand complex dependencies in the cascade of network failures, and present centralized algorithms for network hardening. System administrators confront the fundamental questions of developing techniques to quantify the vulnerabilities in terms of resource constraints of adversaries, and hardening the network to minimize the potential damage. In this chapter we show that some of these problems are NP-complete, which makes them challenging to analyze even for network instances. We study the algorithmic underpinnings of these problems from both the attacker's and defender's perspectives. Our contributions are summarized below:

1. We characterize the inherent vulnerability of the system in terms of the MaxAttackerUtility problem, which quantifies the maximum damage an adversary can cause, when constrained to exploit at most k leaf nodes. We show that this problem is NP-complete, and present a constant factor approximation algorithm for the case when all the predicates are OR.
2. We study the DefenderHardening problem, which involves finding the best set of leaves to secure, so that a potential damage of at most B is caused by an adversary capable of attacking at most k nodes. We show that this problem is NP-hard, and give an $O(\log n)$-approximation algorithm for this problem by reducing it to a hitting set problem. When k is not a constant, this formulation has a super-polynomial size. We show how this can still be solved using the ellipsoid method when all the predicates are OR functions.
3. We study the vulnerabilities and hardening cost in real and synthetic attack graphs. We use our approximation algorithms to observe that the cost of hardening is quite high– typically, over half of the leaves need to be secured in order to bring the defender's potential damage by half (these notions are described formally in Section 10.1). We also study the impact of various network parameters.

Chapter 2

A Unifying Framework for Cascades and Containment Approaches

Cascading effects can be observed in many different applications. A unifying framework that can be used to model various cascading dynamics and containment methods is a discrete dynamical system called the "Graph Dynamical System" (GDS) [62][118]. There is a number of work dealing with problems of controlling spreads in the framework of GDS. In this chapter we present SIS and attack graph models as special instances of GDS, and discuss their containment strategies in the general framework of controlling spreads in GDS.

2.1 Graph Dynamical System (GDS)

The cascading mechanism of failures varies by applications. There are several approaches in the literature that model different cascading scenarios [42, 58, 88]. According to these models, in general, a node fails if one or more of its neighboring nodes fail; the influence of a node on its neighbors is either deterministic or probabilistic in nature. A general mathematical framework that describes this cascading dynamics in networks is called the graph dynamical system (GDS) [80, 63] . A GDS $\mathcal{G}$ is specified by the tuple (G, F) where $G = (V, E)$ is the

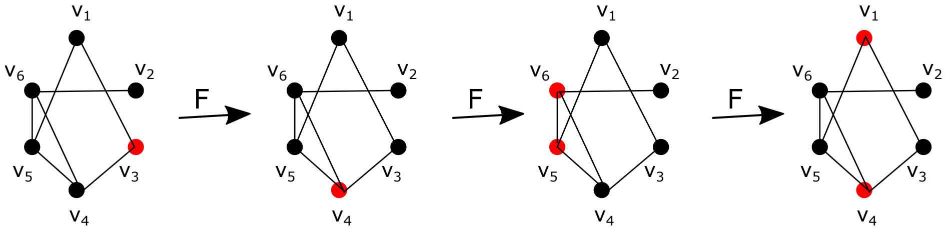


Figure 2.1: Evolution of node states in GDS.

underlying graph and $F = \{f_v | v \in V\}$ is the set of local functions associated with each of the nodes. Each node is in either of two states - 1 or 0 which represent that the node is infected (or compromised) or susceptible (or not compromised) respectively. The node function f_v is a local function over its own state and the states of its neighboring nodes; at any time instance t, the state of node $v \in V$, denoted by $s_t(v)$, is computed by applying f_v on the states of v and its neighbors at time t. The time in GDS progresses either in discrete steps or continuously. Also the graph can be directed or undirected. In directed graphs, the node function of $v \in V$ is defined only over its own state and the states of its in-neighbors. At any given time t, the configuration of GDS is given by the vector of the node states. The configuration space of $\mathcal{G}$ can be viewed as a Markov chain; the chain has 2^n vertices since each node can be in either of two states and there are n nodes in the graph. The GDS has been used as a general framework for studying dynamical mechanisms in networks [75] [60], and can be used to capture different kinds of cascading mechanisms in a network.

2.2 SIS and attack graph models as instances of GDS

Different instances of this model reveal different practical scenarios. We have studied two particular instances of this general model - SIS and attack graph model - that are relevant to a wide range of applications. We discuss below that SIS and attack graph models can be viewed as instances of the general GDS model.

The first cascading model that we address is the SIS or Susceptible-Infected-Susceptible model which is a stochastic diffusion model. SIS is popularly used to model various epidemic spread scenarios. This can be viewed as a special instance of GDS in the following way: let the state space of the nodes involve just two nodes - state 0 and 1 ($s(v) = 0$ and $s(v) = 1$ correspond to the susceptible and infected states of node v respectively). And, let the local function associated with each node v is as follows: if $s(v) = 0$, then $s(v)$ remains 0 with probability $p = \prod_{u \in N(v)} (1 - \beta)^{s(u)}$ and changes to state 1 with probability $1 - p$ where $N(v)$ is the set of v's neighbors; if $s(v) = 1$ it changes to state 0 with probability γ.

It is easy to see that, the cascading process due to this state and local function definition corresponds to the SIS process. At any time instance, a susceptible node (or a node that has not incurred failure) becomes infected (or incurs failure) if any of its infected neighbors propagates the infection to it with probability β; each infection propagation happens independently with probability β and therefore at any time step a susceptible node remains susceptible with probability $(1 - \beta)^k$ where k is the number of infected neighbors at that time. On the other hand, at any time instance, an infected node becomes susceptible with probability γ. Therefore, SIS can be viewed as a special case of GDS with the above state configuration and local function definition.

Similarly, the cascading phenomena in many cybersecurity scenarios can also be viewed as instances of GDS. A popular model to represent vulnerabilities in computer networks and explain cyber attack cascades is a model called “attack graph”. Attack graphs are graph based models to describe the explicit dependencies among different security states and configurations in a network; the compromise of each state depends on the neighboring states or configurations in the attack graph and the dependencies are often described as AND or OR functions. It is easy to observe that attack graphs can be viewed as a special instance of GDS. In GDS, if we consider the states of the nodes as either $True$ or 1 (which corresponds to attacked or compromised) and $False$ or 0 (which corresponds to secured or not compromised), and the local function of a node v as either AND or OR function of the states of $N(v)$, i.e., the set of v's neighbors, then the correspondence between the attack graph models and cybersecurity is clearly evident. This means that in the attack graph, if the node function is AND, then the failure propagates to a node if all its neighbors have incurred failure. Similarly, in the case of OR function, a node fails if any of its neighbors fail. Such cascading processes generally have been studied in the cybersecurity literature in the form of “attack graphs” [88], [30]. This is the second instance of GDS that we examine in this thesis. The attack graph model captures many instances in cyber and infrastructure security applications (e.g., power grid failures [58]).

2.3 Approaches for minimizing cascades

To contain the cascading effects in networks in the framework of GDS, two approaches can be adopted - either to make changes to the network topologies, or modify the local functions. This thesis studies both these strategies in containing failure cascades in two different application areas - epidemics and cybersecurity.

Making topological changes is a common method adopted in epidemiology literature. Failure cascades can be minimized by making the networks less connected. For this, either nodes or edges can be removed. Node removals correspond to interventions such as vaccinating an individual, while edge removal is more akin to social distancing. These node and edge intervention strategies have particularly been studied for epidemic spread scenarios. In the first part of this thesis, we study the node and edge removal strategies for containing cascades in the SIS model.

On the other hand, modifying local functions is more applicable to various cybersecurity applications, e.g., updating the firewall rule or changing access privileges. In the second part of this thesis, we study the strategies of changing local functions in dealing with failure cascades in cybersecurity. Making changes to the local functions can be both static and dynamic in cybersecurity settings. We study approaches for making static changes to the local functions in the form of network hardening in attack graphs. This refers to enacting certain security measures that make positive impacts in vulnerability exploitation scenarios. However, interaction between the cyber attacks and defenses is a never-ending arms race and typically involves very intelligent and adaptive agents. Therefore it is more practical to study approaches for making dynamic changes to the local functions in GDS. We study the

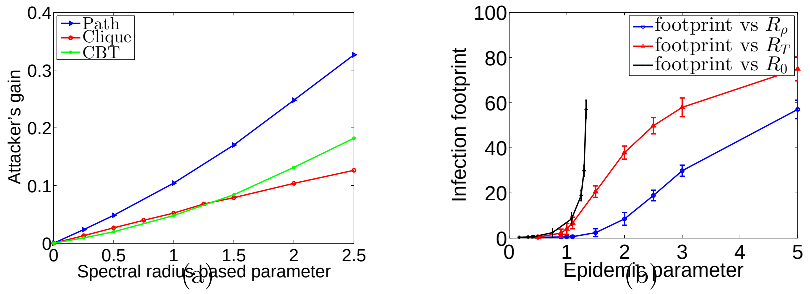


Figure 2.2: Spectral radius based parameter does not make a phase change to the attacker's gain, while it practically does so to epidemic duration and footprint in SIS/SIR model.

effects of making dynamic local function changes in the form of a novel game of persistent and stealthy moves on network resources, which we call "FlipNet." In this game setting, the defender repeatedly and stealthily impose changes on the node functions in its favor, while the attacker also makes changes to the node functions in its favor repeatedly and stealthily. We study this game and present insights into the attacker's and the defender's best strategies analytically and experimentally.

2.3.1 Epidemic Containment: Making Topological Changes in GDS

In applications related to epidemic spread, the containment strategies generally involve interventions such as vaccination and social distancing, which corresponds to node removal and edge removal respectively. These in turn correspond to making topological changes in networks in the framework of GDS. It is a question of interest as to how much topological change is needed that can effectively contain the spread of the epidemic. In an interesting result by Ganesh et al. [40], the epidemic spread in SIS model has been shown to be characterized in terms of the spectral radius of the network, i.e., the largest Eigenvalue of the adjacency matrix of a network. Particularly, Ganesh et al. show that if the spectral radius of a network is less than an epidemic parameter equal to the ratio between the recovery rate and the infection rate, then the spread is going to die out quickly. This is illustrated in figure 2.3.1. This allows for developing efficient methods for removing nodes and edges that bring the spectral radius below this epidemic parameter. We study both the decentralized and centralized approaches for removing nodes and edges, i.e., making topological changes in the network. The decentralized strategies are studied in chapter 5, and the centralized strategies are presented in chapter 6.

2.3.2 Containing Cyberattack Cascades: Adjusting Local Node Functions in GDS

Unlike the epidemic spread scenario, node intervention or edge intervention strategies are not always applicable in cybersecurity applications. For example, the system admin may not be able to terminate a service, but she may be able to change firewalls or modify access privileges to ensure better security of the system. Often these policy changes are dynamic in cybersecurity settings since both the defenders and the attackers continue to adapt to the opponent's strategies. Therefore, as a means of containing cyberattack cascades, it is more applicable to adjust local node functions in the framework of GDS.

Some of the objectives in cybersecurity applications involve the potential damage that an attacker causes in a static function configuration or the average time that the defender/attacker controls the resources in the network (called the gain), etc. These objectives have been studied in the cybersecurity literature in different contexts. However, unlike epidemic duration, no result has been reported in the literature characterizing the potential damage or the gain. It is not obvious if there exist any summary parameter that makes phase changes to the potential damage or the gain. We also have observed that spectral radius does not characterize the gains (unlike in the case of epidemic spread); see figure 2.3.1.

2.3.3 Comparison Between the Two Approaches

Epidemics and cyber-attack cascades capture different dynamics in the framework of GDS. Their containment approaches also concern different control strategies in GDS. In this thesis we present our study of these two application areas in two parts. We discuss below the similarities and differences between the two application areas in terms of containment objectives, cascade models, and control methods.

While both epidemics and cyberattack cascades involve failures of different kinds, there exist differences in the questions of interests regarding their containment. An epidemic spread is a natural phenomenon and the general question of interest in containing it is to find efficient intervention strategies that would make the epidemic completely die out in a short span of time or make the infection footprint small. On the other hand, the cybersecurity is viewed as a continuous arms race between competing attacker(s) and defender(s) who are both rational, self-interested and independent agents. Also, due to limited budgets, absolute defense against attacks (which corresponds to die-outs in epidemics) is often infeasible in cybersecurity. Generally, one asks the question of how to minimize the damage due to the attacks given a limited budget in a limited or infinite time span; this corresponds to minimizing the footprint in epidemics. So, while the objective of die-out applies only to epidemic containment, the problem of footprint minimization applies to both epidemics and cybersecurity. In this thesis, we have focused on the problem of making an epidemic to die out with minimum intervention cost using a spectral radius based characterization; the solutions also make the footprint small, as we experimentally show. For the cybersecurity problems, we directly address the footprint minimization question in two different settings. Addressing attack graph hardening problems, we study how potential damage in a network can be minimized given a limited budget. With a infinite timing game framework between a defender and an attacker, we study the problem of minimizing the attacker's control time over the resources, which corresponds to the footprint minimization problem.

Epidemics and cybersecurity applications differ in the fact that one is a natural process and the other is spread by malicious agents. However, the processes of their containment are similar. The intervention decisions for the containment can be taken by a single agent in a centralized manner in both the applications, e.g., CDC distributing limited vaccines, or a system admin putting security hardening measures in network components. The interventions can also be carried out in a decentralized manner by independent and self-interested rational agents. For example, in the epidemic outbreak scenario, each individual takes the decision of whether to take the vaccine for a cost or to simply enjoy “herd immunity” and not to take the vaccines. Similarly in a malware spread scenario, each user of a computing unit takes the decision of whether to install security measures. We studied these questions with game theoretic formulations. For the problems related to epidemics, we have formulated a game among the nodes as agents in a network. For the cybersecurity applications, we have studied problems in game frameworks involving two players - a defender and an attacker. We have limited the number of players to two, because many practical cybersecurity applications involve just two players - an attacker and a defender. Also, finding results in games involving more players in our cybersecurity problem settings have turned out to be more difficult. However, studying them for multiple attackers and multiple defenders will be valuable work in the future.

For the epidemics and cyberattack propagations, we have studied two different cascade models - SIS and attack graphs. Although they are different in dynamics, they are similar in their special instances. Consider the SIS process in discrete time model where the infection probability is 1. Now, consider the attacker graph model where each node function is OR. It is easy to observe that these two special instances of SIS and attack graph models correspond to the same cascading effect. The SIS process in its continuous model also corresponds to a version of the FlipNet game model that we study for cybersecurity applications. We discuss in chapter 9 that in FlipNet, when the attack graph is undirected and the source node is restricted to receive the attack only once, the attack propagation essentially follows the SIS process for exponential move strategies of the attacker. This allows us to use the spectral characterization of SIS and present results on the attack's persistence.

In our study of epidemic containment, which uses the SIS cascade model and focuses on the complete die-out objective, we have primarily used the spectral radius based characterization of SIS developed by Ganesh et al. [40]. However, as we have discussed above, we have considered the attack graph model for cascades and focused on the footprint minimization objective, as it makes more sense in cybersecurity applications. To the best of our knowledge, it remains unknown if there exists any parameter to characterize the failure footprint in these problem settings. We have experimented with the spectral radius based parameter and found that while it characterizes epidemic footprints to a large extent, it fails to do so in the cyberattack footprints. We have shown this for attacker's gain in the FlipNet problem setting in figure 2.3.1. As a result, while our epidemic containment solutions relie on the spectral characterization, we have taken different approaches for finding efficient cybersecurity measures. We discuss them in two parts in this thesis.

Chapter 3

Related Work

The related work that concerns the problems addressed in this thesis comes from multiple areas: epidemic spread, algorithms and game models for epidemic control, attack graph models for computer security and models for covert and persistent attacks. There is a general research interest in studying dynamic processes on large graphs, e.g., (a) blogs and propagations [46, 65], (b) information cascades [42, 43], (c) marketing and product penetration [102], power grid failures [47], and malware and cyber-attack cascades [10, 30]. These dynamic processes can be considered as different forms of failure propagation. A common issue with the related studies in the literature is that they rely on simplistic models involving unrealistic assumptions. Specifically, the problem of failure containment has not been well studied for networks.

3.1 Epidemic Spread

A classical text on epidemic models and analysis is by May and Anderson [6]. Most work in epidemiology is focused on *homogeneous models* [11, 6].

In this thesis, we study network-based models. Much work in this model has been devoted to finding epidemic thresholds (i.e., the minimum virulence of a virus that results in an epidemic) for a variety of networks [92, 122, 40, 95]. A fundamental property about the dynamics of epidemics in many models is that of a phase transition from a small number of infections to a large number of infections as the transmission probability is increased; this process is characterized in terms of the reproductive number in the case of differential equation-based models [83], and in terms of the spectral radius for SIS/SIR models defined on networks [40, 95].

3.1.1 Algorithms and Game Models for Epidemic Control

There has been much work focused on finding optimal strategies for vaccine allocation [18, 74, 21]. Cohen et al [27] studied the popular *acquaintance* immunization policy (in which a random neighbor of a randomly chosen person is immunized), while Kuhlman et al. [61] studied two formulations of the problem of blocking a contagion through edge removals under the model of discrete dynamical systems. Tong et al. [117, 115], Van Miegham et al. [79] and Prakash et al. [96] proposed various node-based and edge-based immunization algorithms based on minimizing the largest eigenvalue of the graph.

However, it is commonly observed that competing incentives result in limited compliance to directives to get vaccinated or to install anti-virus software. There is a vast amount of literature on modeling such behavior using non-cooperative game models, e.g., [8, 7, 67, 12, 89, 70, 45, 57, 55, 56].

A large part of this research is based on differential equation models, e.g., [12, 57, 55, 56, 38, 100] and commonly rely on simplified assumptions about uniform mixing of the players in the population. This greatly simplifies the problem, and enables the derivation of tight analytical bounds and a detailed characterization, e.g., [38, 100, 12]. However, it is not easy to extend these approaches to heterogeneous networks.

The work of Aspnes et al. [8] was among the first to examine these problems on networks, especially from an algorithmic perspective; this was further developed in [67]. They characterize NE in terms of the network properties, such as the maximum degree and conductance, and develop algorithms for approximating the PoA. However, both these approaches focus on an SIR model with a transmission probability of 1, so it suffices to consider connectivity instead of percolation. Omic et. al. [89] develop a formulation by combining a N-intertwined, SIS epidemic model with an non-cooperative game model, which simplifies the diffusion process by a mean-field approximation. In their game model, the level of protection for each node is captured by its recovery rate α_i and the relative cost it incurs for protection is c_i. Each node i chooses its recovery rate $\alpha_i \in [0, \alpha_{\max}]$, for a predetermined value $\alpha_{\max}$ so as to minimize its cost function $\text{cost}(i, \alpha) = c_i\alpha_i + v_{i\infty}$. Here, $v_{i\infty}$ is the steady state probability of infection for i. The paper characterizes the NE and derives bounds for PoA for this model.

Other network security aspects have also been extensively studied via non-cooperative games. Two classes of models are that of Interdependent Security games (IDS) [51] and the security game models of Grossklags et al. [45]. Another related thread is the use of Stackelberg strategies, e.g., [49]. Non-cooperative game theory has also been used for modeling and analysis of various problems in communication networks e.g., [33, 81].

The epidemic containment game model studied in chapter 5 is a simultaneous one round game, i.e. each node takes its decision once without knowing the decisions of others. Prior works in the context of security game have considered similar simultaneous game models as well, e.g. [45, 7, 67, 89]. Our work makes one key assumption about the utility function: the utility of an unsecured node depends on a global information. In our paper, the spectral radius of the whole network is precisely that global information. Similar assumptions have been made in prior works as well. The utility function of a node i in the security game model of [45] depends on global protection level $H(e_i, e_i)$, where e_i and e_{-i} are the protection level chosen by i and the rest respectively. They consider different variants of this game with different H functions (eg. $H(e_i, e_i) = min(e_i, e_i)$ or $H(e_i, e_i) = \frac{1}{|V|}\sum_i e_i)$; the utility of a node depends on the decisions taken by all the other nodes in each of those games. In the virus containment game of [7], the cost function of a player i (which is $a_i C + (1 - a_i) L.p_i(\overrightarrow{a}))$ depends on $p_i(\overrightarrow{a})$, which is the probability of node i becoming infected given overall strategy profile $\overrightarrow{a}$. Similar cost function dependent on a global property has been considered in [67], where the cost function is, $cost_v(\overrightarrow{a}) = a_v C_v + (1 - a_v) L_v.p_v(\overrightarrow{a})$. Similarly, in the virus protection game of [89], the cost of a node, $J^{(i)} = c_i \delta_i + v_{i\infty}$, depends on that node's steady state infection probability $v_{i\infty}$, which depends on other nodes' curing strategies, δ_i. Therefore, our assumptions are very common in the prior literature.

3.2 Cyber-attack Cascades

Modeling vulnerabilities in computer systems and developing corresponding security strategies are important and extensively researched problems in the cybersecurity literature. A large part of it deals with known vulnerabilities and the complex inter-dependencies among them with models such as attack graphs and model checking based techniques [88], [109], [99]. Specially, graph based models, such as attack graphs and attack trees [30], have been proven useful in identifying critical vulnerabilities in large networked systems. While these models serve as useful tools to the system administrators for assessing the security of a system, they also raises interesting algorithmic questions from the perspective of budget constrained defenders and attackers, which has not been sufficiently addressed in the literature.

While addressing known vulnerabilities is difficult enough due to their high volume, addressing unknown vulnerabilities and unrecognized exploitations are even more challenging. This has become a practical problem due to some of the big hacking incidents of the recent past, e.g. [106], [36]. Two of the important factors that characterize such advanced attacks are - stealthiness and persistence. These factors add to the complexity in understanding advanced cyber attacks, which has not been studied well in the literature.

3.2.1 Attack Graph Models and Methods

A number of works in the literature have used model checking based techniques for analyzing the security of networks with vulnerabilities. Vulnerabilities and attack scenarios in single host systems and heterogeneous networks have been studied by [93],[101] and [98]. This line of work has been extended and further analyzed by [109], [50], [110], [88]. In these works, nodes in the attack graphs represent network states - safe or faulty - and the edges represent the actions of the attacker that change the states; the goal is to find attack scenarios in which the system can be compromised, to identify probabilities of attacks, or to protect the system by hardening optimal subgraphs. One issue with such attack graph models is the scalability since the state space grows exponentially as the network grows. [5] has proposed a more compact and scalable representation of attack graph using the “monotonicity” property which states that the precondition of an exploit does not become invalid when other vulnerabilities are successfully exploited.

In a slightly different approach, attack graphs are used to represent logical exploit-dependencies [87], [120], [99], [30], [94]; in this representation, nodes represent vulnerabilities or exploits, and the edges represent their inter-dependencies. In the above-mentioned works, the problem of optimal hardening of network vulnerabilities have been addressed. [120] have formulated the hardening problem as a satisfiability problem, while [30] have used genetic algorithm to find optimal hardening solutions. In this work we have formally studied the algorithmic underpinnings of the hardening problem in logical attack graphs and analyzed computational complexities.

3.2.2 Game Models for Stealthy and Persistent Cyber Attacks

Dijk at al. [119] proposed a game theoretical model, called FlipIt for analyzing covert and persistent cyber attacks. The work presents analyses on the interaction between two players - a defender and an attacker - in controlling a single contested resource under different strategy classes. Both the players choose strategies for making repeated security moves to control the resource. The work considers two categories of strategies - non-adaptive and adaptive. Non-adaptive strategies can vary from simple periodic strategies where move intervals are a fixed number to a complex renewal process strategy where move intervals are chosen uniformly at random from a fixed probability distribution. In the adaptive strategies category, the players may obtain the full history of the previous moves(called “FH player”) or only the last move information (called “LM player”) while making a move. They compute dominant strategies and Nash equilibria under different settings and prove results on the equivalence of different strategy classes. One interesting finding is that simple periodic strategies are strongly dominant strategies for the attacker in many settings; e.g., a non-adaptive attacker’s best response against a renewal process defender strategy is to play periodically with a random phase.

The FlipItmodel for one resource has been extended to a multi resource model by [69], which they refer to as the “FlipThem” model. This work presents the gains and benefits expression of the players under different strategy class settings as introduced in FlipIt. Finding the best response strategy becomes more challenging in this multi resource model. However, the work presents a linear programming method for finding best response when there are only two resources involved and the players’ move strategies can be modeled with a Markov model.

One limitation of the FlipThem model is that, the resources in this model do not constitute a network. However, in practice, many advanced persistent attacks propagate through network links [106]. Our work is the first to consider networked models for stealthy and persistent attacks.

Part I

Epidemic Containment in SIS Model

Chapter 4

Introduction

Among the many diffusion models, SIS/SIR models are very commonly used in the literature, as they capture a wide range of cascading scenarios, such as disease outbreak in human population, malware propagation in email networks [6],[41] etc. In the SIS model of epidemic spread [83, 40], nodes are in states *Susceptible* (S) or *Infected* (I); initially, some source node gets infected (e.g., randomly, or from some distribution), and all other nodes are susceptible. Each infected node v infects each of its neighbors u currently in state S at rate α (the transmission rate); if neighbor u gets infected, it switches to state I. Also, each infected node v switches back to state S at rate γ. In the SIR model, an infected node, after recovery, switches to state *Recovered* (R) and never returns to S or I state. In this part, we will focus on the ratio γ/α, which will be denoted by T, and will be referred to as the epidemic threshold. The spectral radius, $\lambda_1(G)$, of G is the largest eigenvalue of the adjacency matrix of the contact graph. The characterization of [40, 95] implies that the epidemic dies out quickly (in $o(n)$ time) if $\lambda_1(G) < T$. This result is extended to other models by Prakash et al. [95]. Note that in this model, the epidemic dies out completely in the limit; however, nodes can get repeatedly infected while it has not died out. The characterization, however, is not tight, and the converse is not true; instead, Ganesh et al. [40] show that the epidemic persists for a "long" time if the difference between the first and the second eigenvalues of the laplacian of G is bounded. However, for many kinds of graphs, the condition based on spectral radius is quite close, and the epidemic lasts "long" if $\lambda_1(G)$ is much larger than γ/α.

In this part, we study epidemic control in network using game formulations based on the spectral characterization, discussed above. Our games involve an undirected graph $G = (V, E)$ on the set V of players (also referred to as "nodes"). For the players, two common strategies for epidemic control are vaccination (or, installing anti-virus software) and edge removal (e.g., social distancing). We introduce two game formulations using these two intervention strategies - a vaccination game that we call *Epidemic Containment (EC)* game and a network formation game. In the *Epidemic Containment (EC)* game, each node chooses its vaccination strategy (either vaccinate or not) by trading off the cost of vaccination against the risk of infection, which depends on the spectral radius of the network. In chapter 5 we study this game and its solutions in Nash equilibria (NE). This is followed by the study of algorithms for computing social optimum solutions in chapter 6; the algorithms compute efficient solutions for vaccination interventions, as well as, edge removal interventions.

In the following, we briefly describe some of the game theoretic notions [86] that will be used in the following chapters. In our games, nodes represent players or agents. Each player i chooses a *strategy*, a_i from a set of possible *strategies*, A_i (e.g., vaccinating or removing edge or adding adge). We consider *one shot simultaneous move games* for our two game formulations; in this, all the players simultaneously choose their strategies. Any vector of strategies selected by the players is called a *strategy profile*, $\mathbf{a}$. For each player i, the *payoff* or *cost* function, $u(i, \mathbf{a})$ or $cost(i, \mathbf{a})$ maps a strategy profile $\mathbf{a}$ to a real valued payoff or cost. For a strategy profile $\mathbf{a}$, the *social cost* is, $\text{cost}(\mathbf{a}) = \sum_{i \in V} \text{cost}(i, \mathbf{a})$. A strategy profile $\mathbf{a}$ is in a *Nash Equilibrium*, if no player can increase its payoff by unilaterally deviating from its strategy in $\mathbf{a}$. The *social optimum* cost for a game is the minimum social cost over all strategy profile. Finally, *price of anarchy* denotes the ratio of the worst cost of a Nash equilibrium strategy profile to that of the social optimum cost.

Now, we will explain some of the graph theoretic notions that we have used in our analyses. We are given a network of players represented by an undirected contact graph $G = (V, E)$. A node $v \in V$ is in contact with a set $N(v) = \{u \in V : (u, v) \in E\}$, also referred to as the neighbors of v. The degree of node v is denoted by $d(v) = |N(v)|$. We use $\Delta = \Delta(G)$ to denote the maximum degree of any node in G.

We will study two random graph models in our analyses - Erdős-Rényi and Chung-Lu random graphs. In Erdős-Rényi random graph model $G(n, p)$, each pair of vertices has an edge between them with probability p. The expected number of edges in $G(n, p)$ is $\frac{n(n-1)p}{2}$ and expected degree of each node is np. The spectral radius of $G(n, p) = (1 + o(1)) \max\{np, \sqrt{\Delta}\}$ [59]. The Chung-Lu [23, 24] model $G(\mathbf{w})$, given a weight sequence $\mathbf{w} = (w(v_1, V), w(v_2, V), ..., w(v_n, V))$ for nodes $v_i \in V$, is defined as follows: for every pair $v_j, v_k \in V$, v_j is adjacent to v_k with probability $\frac{w(v_j,V)w(v_k,V)}{\sum_{v_i \in V} w(v_i,V)}$.

Some basic properties of the spectral radius, denoted by $\rho(G)$ or $\lambda_1(G)$, for random graph models are as follows.

1. $\max(\sqrt{\Delta(G)}, \frac{2m}{n}) \leq \lambda_1(G) \leq \Delta(G)$, where m, n denotes the number of edges and nodes respectively.
2. $\lambda_1(G[V']) \leq \lambda_1(G)$, where $G[V']$ is the induced subgraph of $G = (V, E)$ on node subset, $V' \subset V$
3. Almost surely, $\lambda_1(G) = (1+o(1)) \max\{\sqrt{\Delta}, np\}$ for Erdős-Rényi random graph, $G(n, p)$

See [78] for proofs and an introduction to spectral graph theory.

Chapter 5

Epidemic Containment with Decentralized Strategies

In this chapter, we focus on the collective defense against epidemic spread when agents' strategies are limited to only vaccination or putting in security measures (e.g. installing anti-virus software). We develop a new formulation which is based on the spectral characterization of the SIS model on networks and motivated by the approaches to contain the epidemic by reducing the spectral radius [115, 79]. In this way, our formulation incorporates a realistic infection model in general heterogeneous networks. We show that, in this model, pure Nash equilibria (NE) always exist, and can be found by a best response strategy. We analyze the complexity of finding NE, lower and upper bounds on their costs, and the Price of Anarchy (i.e., the ratio of the costs of the worst NE to the best NE) in some general graph families as well as random graph models. In particular, for arbitrary power-law graphs with exponent $\beta > 2$, we show that the PoA is between $\Theta\left(T^{(\beta-1)}\right)$ and $\Theta\left(T^{2(\beta-1)}\right)$, where $T = \gamma/\alpha$ is the ratio of the recovery rate to the transmission rate in the SIS model. For the Chung-Lu random power-law graph model, we prove a tighter bound of $\Theta(T^{2(\beta-1)})$ for the PoA. We study the characteristics of Nash equilibria empirically in different real communication and infrastructure networks, and find that our analytical results can help explain some of the empirical observations. Finally, we study approaches to reduce the effects of non-cooperative behavior through Stackelberg strategies. We discuss our contributions in greater detail below.

A game formulation based on spectral properties: We introduce the *Epidemic Containment (EC)* game on a network $G = (V, E)$ of players, for the SIS model of epidemic spread in Section 5.1. The individual actions of a player are either to be secured (which has a fixed cost, corresponding to the price of a vaccination/anti-virus software), or not. If a player is not secured, and if $\lambda_1(G') > \gamma/\alpha$, where G' is the graph induced by the insecure nodes, he/she incurs a high cost of infection, since the epidemic is likely to last long (see Section 5.1 for the formal definition). We show that pure Nash equilibria (NE) always exist in an EC game, and can be found by a best response strategy. Further, a minimum cost Nash equilibrium is also a social optimum, and finding the social optimum (or a NE of cost within a constant factor of it) is NP-complete, in general (see Section 5.2).

Structure of equilibria in arbitrary graphs: We derive bounds on the cost of the worst NE and the PoA in terms of the maximum degree in general graphs (Section 5.3). There can be an exponential number of Nash equilibria, and the ratio of the maximum cost of any NE to that of the social optimum (also referred to as the price of anarchy, and denoted by PoA) can be $\Omega(\Delta(G))$, where $\Delta(G)$ denotes the maximum node degree in G. When G has a power law degree sequence with exponent $\beta > 2$, we show that the cost of the social optimum $\in [c_1 n/T^{2(\beta-1)}, c_2 n/T^{(\beta-1)}]$, for constants c_1, c_2; this implies the PoA is $O(T^{2(\beta-1)})$. Further, a NE with cost within a factor of $\Theta(T^{(\beta-1)})$ of the social optimum can be computed in polynomial time.

Structure of equilibria in random graph models: We then study the structure of NE in the Erdős-Rényi and Chung-Lu random graph models [23]; the latter has been shown to be relevant for a broad class of real world networks. We prove that in Erdős-Rényi graphs $G(n,p)$, if $p = \Omega(\log n/n)$ (which is needed for the graph to be connected), and if $T^2 = O(np)$, every NE has cost $\Omega(n)$, and the PoA is $\Theta(1)$. We study the Chung-Lu model defined by a power-law weight vector with exponent β (see definition in Section 5.1), which gives a random graph with degree sequence close to a power law. We prove that when $\beta > 2$, and the weight sequence and T satisfy additional weak assumptions, the social optimum has cost $\theta(\frac{n}{T^{2(\beta-1)}})$, and can be approximated by picking high weight nodes; note that this corresponds to the upper bound we prove for general power law graphs. In contrast, the worst cost NE is $\Omega(n)$, which can be obtained by favoring low weight nodes. This leads to a PoA of $\Theta(T^{2(\beta-1)})$ in this model (see Section 5.4).

Empirical analysis of the properties of the equilibria: We study the structure of equilibria in EC games in six different real and random networks, on which malware could spread. Our main observations are summarized below (see Section 6.7 for details): (i) We find that estimates of the minimum cost NE scale as $\Theta(\frac{n}{T^{2c(\beta-1)}})$ for the scale free networks, where c is a small constant close to 1, which is close to our bounds for general graphs and the Chung-Lu model; (ii) We compute a lower bound on the PoA and find that it scales roughly as $\Theta(T^{2c'(\beta-1)})$ where c' is a constant close to 1, which is again close to our theoretical bounds; (iii) An interesting observation is that the degree distribution in the graph induced by the insecure nodes in random NE is very close to the degree distribution in the graph G in most networks; (iv) We study the community structure in the networks, and find that generally larger communities have a disproportionately high fraction of secure nodes, in contrast to their size. This might be useful in understanding what kinds of nodes have the greatest incentive to secure themselves; (v) Finally, we consider Stackelberg strategies, to explore how to mitigate the effects of distributed control and the PoA by influencing a small set of nodes. We find that the cost of random NE reduces quite a bit for a small fraction of high degree nodes secured initially.

5.1 Preliminaries and Epidemic Containment Game Model

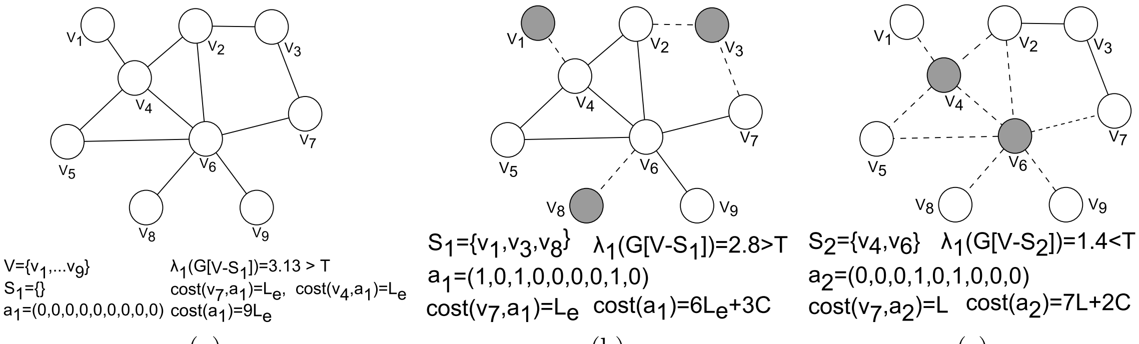

v1
v2
v3
v4
v5
v6
v7
v8
v9
V={v1,...v9}
S1={}
a1=(0,0,0,0,0,0,0,0,0)
λ1(G[V-S1])=3.13 > T
cost(v7,a1)=Le, cost(v4,a1)=Le
cost(a1)=9Le
(a)
S1={v1,v3,v8} λ1(G[V-S1])=2.8>T
a1=(1,0,1,0,0,0,0,1,0)
cost(v7,a1)=Le cost(a1)=6Le+3C
(b)
S2={v4,v6} λ1(G[V-S2])=1.4<T
a2=(0,0,0,1,0,1,0,0,0)
cost(v7,a2)=L cost(a2)=7L+2C
(c)

Figure 5.1: EC game instance (G, T, C, L, L_e) where $T = 1.5$ on a graph of nine nodes; (a) Strategy profile $\mathbf{a}_1$ in which no node is secured; (b) Strategy profile $\mathbf{a}_2$ where three nodes are secured but spectral radius of the attack graph is more than $T = 1.5$; (c)Strategy profile $\mathbf{a}_3$ where two nodes are secured and spectral radius is below T. For both $\mathbf{a}_1$ and $\mathbf{a}_2$, the epidemic is likely to last long. For $\mathbf{a}_3$, the epidemic dies out quickly; $\mathbf{a}_3$ is also in NE.

The *Epidemic Containment* (EC) game involves an undirected graph $G = (V, E)$ on the set V of players (also referred to as “nodes”). Each node x decides independently whether to become secured/vaccinated (denoted by $a_x = 1$) or not (denoted by $a_x = 0$); a_x is the strategy selected by node x, and $\mathbf{a} = (a_1, a_2, ..., a_n)$ denotes the strategy profile of all the nodes. If node x decides to get secured, i.e., $a_x = 1$, it incurs a cost C (e.g., the cost of a vaccination or anti-virus software). If node x does not get secured, i.e., $a_x = 0$, its cost (denoted by $\text{cost}(x, \mathbf{a})$) depends on whether or not the epidemic (restricted to the graph induced by the insecure nodes) dies out quickly or not under the strategy profile $\mathbf{a}$— we let $L < C$ and $L_e > C$ denote the costs in the former and latter cases, respectively. The motivation is that if the epidemic does not die out quickly, node x is more likely to be infected, and incurs a higher cost than C; however, if the epidemic dies out quickly, the cost incurred is much smaller than C. Let $S = S(\mathbf{a}) = \{x \in V : a_x = 1\}$ denote the set of secure nodes in the strategy profile $\mathbf{a}$. We call the graph $G[V - S(\mathbf{a})]$ induced by the set $V - S(\mathbf{a})$ of insecure nodes as the “attack” graph. Following the characterization of [40, 95], we have for any $v \in V$ and strategy profile $\mathbf{a}$:

$$\text{cost}(v, \mathbf{a}) = \begin{cases} C, & \text{if } a_v = 1, \\ L, & \text{if } a_v = 0 \text{ and } \lambda_1(G[V - S(\mathbf{a})]) < T, \\ L_e, & \text{if } a_v = 0 \text{ and } \lambda_1(G[V - S(\mathbf{a})]) \geq T. \end{cases}$$

For a strategy profile $\mathbf{a}$, the social cost is, $\text{cost}(\mathbf{a}) = \sum_{v \in V} \text{cost}(v, \mathbf{a})$. If the epidemic dies out quickly, then $\text{cost}(\mathbf{a}) = |S|C + |V - S|L$ where $S = S(\mathbf{a})$; otherwise, $\text{cost}(\mathbf{a}) = |S|C + |V_e|L_e + |V - S - V_e|L$,where V_e is the set of insecure nodes that fall in those components of the attack graph where the epidemic lasts long. Thus, an instance of the EC game is defined by the tuple (G, T, C, L, L_e). The optimum social cost of an instance is denoted by C_{OPT} where, $C_{\text{OPT}} = \min_{\mathbf{a}} \text{cost}(\mathbf{a})$.

A strategy profile $\mathbf{a}$ in an instance of this game is said to be a Nash Equilibrium (NE) if for any alternate strategy a_i' of any player i, in a strategy profile $\mathbf{a}'$, we have: $\text{cost}(i, \mathbf{a}) \leq \text{cost}(i, \mathbf{a}')$. That is, a strategy profile $\mathbf{a}$ is a NE, if no player i benefits by switching his/her strategy, if all other players' strategies are fixed. This is illustrated in Figure 5.1. In this example graph, $\mathbf{a}_3$ is a NE. This is because neither v_4 nor v_6 can benefit by switching from secure to insecure (as λ_1 of the attack graph becomes more than T). Also, no insecure node benefits by switching to a secured stated (as that would only increase its cost from L to C). On the other hand, $\mathbf{a}_1$ and $\mathbf{a}_2$ are not NE, since any of the unsecured nodes can secure itself and get its cost reduced from L_e to C. The following observation gives a simple characterization of a NE in the EC game.

Observation 1. *For an instance (G, T, C, L, L_e) of the* EC *game, a strategy profile* $\mathbf{a}$ *is a NE if and only if $\lambda_1(G_{V-S(\mathbf{a})}) < T$, and for all $S' \subset S(\mathbf{a})$ with $|S(\mathbf{a})| - |S'| = 1$ we have $\lambda_1(G_{V-S'}) \geq T$.*

To simplify the notation, for the rest of the chapter, we will focus on instances (G, T, C, L, L_e) of the EC game, where $C = 1$, $L = 0$ and $L_e > 1$. Under this assumption, and because of the above observation, it follows that if $\mathbf{a}$ is a NE, then $cost(\mathbf{a}) = |S(\mathbf{a})|$. All our results extend naturally to the general case. The *Price of Anarchy* (PoA) is defined as the maximum and minimum cost NE, while the *Price of Stability* (PoS) refers to the ratio between the best cost of an equilibrium and the social optimal. We will also study Stackelberg strategies, in which a centralized authority is allowed to control the strategies of a fraction of agents, while the remainder decide non-cooperatively [104].

5.2 Existence and complexity of Nash Equilibria

We first observe below that any "minimal" set of secure nodes corresponds to a NE.

Observation 2. *Let S be a minimal set such that $\lambda_1(G[V - S]) < T$. Then, the strategy profile* $\mathbf{a}$ *with $a_i = 1$ for $i \in S$, and $a_i = 0$ for $i \notin S$ is a NE.*

Proof. By definition, for any $S' \subsetneq S$, we have $\lambda_1(G[V - S']) \geq T$. Therefore, no secure node $i \in S$ has incentive to become insecure. Further, no insecure node $i \in V - S$ has incentive to become secure, since $\lambda_1(G[V - S]) < T$. Therefore, $\mathbf{a}$ is a NE. □

Nash equilibria in *EC game* can be constructed by simple iterative methods. Given two arbitrary permutations, π and ρ of nodes in V, the following strategy, that we call IterativeSecure, finds a NE of the EC game.

IterativeSecure(π, ρ): Method for *EC game* NE construction.
Input: Graph $G = (V, E)$, threshold T, permutations π,ρ on V

- *Initialize:* Empty secured set S.
- *Stage 1:* Secure nodes iteratively in the order, π : at step i, update $S = S \cup \{v_{\pi(i)}\}$. Stop if $\lambda_1(G[V - S]) < T$.
- *Stage 2:* Unsecure nodes of S in the order, ρ : at step i, if node $v_{\rho(i)} \in S$ and if $\lambda_1(G[V - (S - \{v_{\rho(i)}\})]) < T$, then unsecure that node, i.e., update $S = S - \{v_{\rho(i)}\}$.
- *Output:* Secure set S. Keep set $I = V - S$ insecured.

Observation 3. *For any permutations π and ρ, the* ITERATIVESECURE *procedure results in a NE.*

Proof. Since Stage 1 produces a secured set S for which $\lambda_1(G[V - S]) < T$, from Observation 2 it follows that the procedure constructs a NE if Stage 2 constructs a minimal set S' for which $\lambda_1(G[V - S']) < T$.

In Stage 2, at the ith step, let G_i be the subgraph induced by unsecured nodes. Suppose node $v_{\rho(i)}$ cannot be unsecured as it would make the spectral radius of the attack graph $\geq T$. Since $G_i \subseteq G_j$ and therefore, $\lambda_1(G_i) \leq \lambda_1(G_j)$ for any $i < j$, $v_{\rho(i)}$ cannot be unsecured at the end of Stage 2 without making the spectral radius more than or equal to T. Therefore, the set S' which results from Stage 2 is indeed a minimal set for which $\lambda_1(G[V - S']) < T$. Hence proved. □

Observation 4. *If S is such that $\lambda_1(G[V - S]) < T$, then, there exists $S' \subseteq S$ of secured nodes which corresponds to a NE.*

Proof. In ITERATIVESECURE, applying Stage 2 on S gives a minimal set S' which corresponds to a NE. Hence proved. □

From Observation 2, it follows that the smallest set S such that $\lambda_1(G[V - S]) < T$ is a NE, and is also the social optimum. This implies that the price of stability is 1, which is summarized below.

Lemma 1. *Any instance of the EC game has a pure NE, and the price of stability is 1.*

Proof. Since securing all the nodes makes the spectral radius of the attack graph 0 (i.e. $\lambda_1(G[V - V]) = 0$), therefore in any instance of *EC game*, the social optimum configuration, **a** has the cost that lies in $[0, T)$ (assuming $T > 0$). According to observation 2, **a** is a NE. Therefore there exists a pure NE in every *EC game*. And since **a** is the social optimum configuration, therefore it is also the best NE. So, the price of stability is 1.

□

Corollary 1. There exists a pure NE in any instance (G, T, C, L, L_e) of *EC game.*

Proof. This follows from lemma 1. □

Lemma 2. *Finding the social optimum of an EC game is NP complete. Moreover, the cost of social optimum cannot be approximated within a factor of* 1.3606 *unless P=NP.*

Proof. To prove this statement we reduce the problem of finding minimum vertex cover to this problem. Let I_{MVC} and I_{EC} be two general instances of the two problems defined as follows

1. $I_{MVC}(G', P')$: Given a graph $G = (V, E)$, is there a vertex cover set of size P' or less?
2. $I_{EC}(G, T, C, L, L_e, P)$: Given the *EC game* (G, T, C, L, L_e), is there a configuration with social cost P or less.

We reduce I_{MVC} to I_{EC} as follows: set $G = G', T = \epsilon$, where ϵ is arbitrarily close to 0, set $L = 0, L_e = \infty$, choose C such that $0 < C < \infty$ and set $P = P'C$. Clearly, the reduction takes polynomial time. Now we show this is a valid reduction. If there is a vertex cover set $V_1 \subset V$ of size $|V_1| = P'$, then V_1 corresponds to a secured node set of I_{EC} yielding social cost $P'C$. Because, removing the vertex cover set, by definition, leaves no edge in the graph and so, the spectral radius of the attack graph is 0 which is less than ϵ. Therefore all the unsecured nodes incur zero cost and social cost is $P'C$.

On the other hand, if there is a secured node set $V_1 \subset V$ of cardinality $|V_1| = P$, and social cost PC in I_{EC}, then that means removing V_1 fro G, leaves no edge in the graph. So, by definition, V_1 is a vertex cover to G of size P.

Therefore, this is a valid polynomial time reduction and finding the minimum social cost in *EC game* is NP complete.

Suppose the above claim about the approximation factor is not true and there exists an algorithm that finds a solution to an instance $(G, T, C, 0, L_e)$ where $0 < C < L_e$, that has social cost less than $1.3606 C_{\text{OPT}}$. First, without loss of generality we can assume that such an approximation solution represents a non-epidemic situation, since otherwise an even better solution can be achieved by securing one or more of the unsecured nodes. Therefore, we assume that this approximation solution consists of a set of secured nodes having cost C and a set of unsecured nodes having cost 0. Since all the nodes have the same security cost C and all the unsecured node has cost 0 in any non-epidemic situation, so the approximation solution infers that this solution consists of a number of secured nodes which is less than 1.3606 times that in the optimal solution. Then according to the reduction from vertex cover problem to the *EC game* game problem (described in the proof in Lemma 2), there exists a vertex cover solution of size less than $1.3606 V_{OPT}$ where V_{OPT} is the optimal vertex cover solution. But, this cannot be true, since we know that the vertex cover problem cannot be approximated within a factor of 1.3606 [31]. Therefore, finding the social optimum of the virus diffusion game cannot be approximated within a factor of 1.3606.

□

The above lemma easily follows from a reduction of vertex cover problem to EC game problem. For the proof, the reader is referred to [105].

5.3 The structure of NE in general graphs

We first analyze the PoA in complete bipartite graphs.

Lemma 3. *Consider a complete bipartite graph $K_{a,b}$ and threshold T. Let x, y, z be the largest integers strictly less than T, $\frac{T^2}{\min(a,b,x)}$ and $\frac{T^2}{\max(a,b)}$ respectively. Then, the PoA for $K_{a,b}$ is $\frac{a+b-\min(a,b,x)-y}{\min(a,b)-z}$.*

Proof. Note that $\lambda_1(K_{a,b}) = \sqrt{ab}$. Note that, removal of nodes from $K_{a,b}$ reduces the graph to another smaller complete bipartite graph $K_{a',b'}$ where $a' \leq a, b' \leq b$. It is easy to see that for non-negative integers a'_1, b'_1, a'_2, b'_2, if $a'_1 + b'_1 = a'_2 + b'_2$ and $|a'_1 - b'_1| < |a'_2 - b'_2|$, then $\sqrt{a'_1 b'_1} > \sqrt{a'_2 b'_2}$. This implies that $a'+b'$ is maximized with the constraint that $\lambda_1(K_{a',b'}) < T$ when $|a' - b'|$ is maximized and $a' + b'$ is minimized when $|a' - b'|$ is minimized.

For C_{OPT}, we need to minimize the number of nodes secured, which implies maximize $a' + b'$. Therefore, we remove nodes one by one from $\min(a, b)$ until λ_1 of the residual graph is $< T$. It is easy to see that the number of nodes to be removed is $\min(a, b) - z$, where z is as defined above. For the size of the worst NE, we need to maximize the number of nodes secure or minimize $a' + b'$. For this, we remove iteratively nodes from the bigger part (breaking ties arbitrarily) until $\lambda_1 < T$. This yields the expression in the numerator of PoA in the statement.

□

We now consider bounds for arbitrary power law graphs. Let n_i denote the number of nodes of degree i in G, for $i \in \{1, \ldots, d_{\max}\}$. We assume the degree sequence of G is a power law with exponent β, so that $n_i \propto 1/i^\beta$. We first observe the following useful property.

Lemma 4. *Let G be a power law graph with exponent $\beta > 2$ and let $x \leq c d_{\max}$ for a constant $c < 1$. Then, (1) $\mathcal{E}_0(x) = \sum_{i \geq x}^{d_{\max}} n_i = \Theta\left(n/x^{\beta-1}\right)$ and (2) $\mathcal{E}_1(x) = \sum_{i \geq x}^{d_{\max}} i \cdot n_i = \Theta\left(n/x^{\beta-2}\right)$.*

Proof. We will only prove Case (1); the proof for Case (2) follows on similar lines.

$$\mathcal{E}_0(x) = \sum_{i \geq x}^{d_{\max}} n_i = \Theta(1) \int_x^\infty \frac{n}{z^\beta} dz = \Theta\left(\frac{n}{x^{\beta-1}}\right).$$

Hence, proved. □

Lemma 5. *Let G be a power law graph with exponent $\beta > 2$, where β is a constant and let $T^2 \leq cd_{\max}$ for a constant $c < 1$. Then, there exist constants c_1 and c_2 such that $c_1 \left(n/T^{2(\beta-1)}\right) \leq C_{OPT} \leq c_2 \left(n/T^{(\beta-1)}\right)$.*

Proof. We first consider the lower bound. Consider any strategy profile $\mathbf{a}$ that is a NE; let $I = \{v : \mathbf{a}_v = 0\}$ be the set of insecure nodes in $\mathbf{a}$. We have $\lambda_1(G[I]) \geq \sqrt{\Delta(G[I])}$. Let $A = \{v : d(v) \geq T^2\}$. It follows that for any node $v \in A$, $d_{G[I]}(v) < T^2$, for otherwise, $\lambda_1(G[I])$ would be at least T. This implies that any node $v \in A$ is either secured (i.e., has $\mathbf{a}_v = 1$) or at least $d_G(v) - T^2$ neighbors of v are secured. Let D be the smallest set such that for any $v \in A$ either: (i) $v \in D$, or (ii) at least $d(v) - T^2$ neighbors of v are in D. Then, $\text{cost}(\mathbf{a}) \geq |D|$. Because of the power law degree distribution, it follows that $|A| = \theta\left(n/T^{2(\beta-1)}\right)$. Observe that $r_e := \frac{1}{2}\sum_{v\in A}(d(v) - T^2)$ denotes a lower bound on the number of edges with end points in D which need to be removed. Since G is a power law graph with exponent β,

$$\begin{aligned} r_e = \frac{1}{2}\sum_{i \geq T^2}^{d_{\max}} \left(i - T^2\right) n_i = \Theta(1) \int_{T^2}^{\infty} \frac{n\left(z - T^2\right)}{z^{\beta}} dz \\ = \Theta\left(n/T^{2(\beta-2)}\right). \end{aligned} \tag{5.1}$$

Next, note that $|D|$ is minimized if the largest degree nodes are selected. Consider the largest index j such that $\sum_{i \geq j} i \cdot n_i = \mathcal{E}_1(j) \geq r_e$; then, $|D| \geq \sum_{i \geq j} n_i = \mathcal{E}_0(j)$. From Lemma 4 and (5.1), it follows that $j = \Theta\left(T^2\right)$ and therefore, (again from Lemma 4) $\mathcal{E}_0(j) = \Theta\left(n/T^{2(\beta-1)}\right)$.

For the upper bound, we note that $\lambda_1(G[I]) \leq \Delta(G[I])$. Therefore, if all the nodes in $B = \{v : d(v) \geq T\}$ are secured, then, $\lambda_1(G[I]) < T$ and from Observation 4, $C_{\text{OPT}} \leq |B|$. From Lemma 4, it follows that $|B| = \mathcal{E}_0(T) = \Theta\left(n/T^{(\beta-1)}\right)$. □

Corollary 2. Let G be a power law graph with exponent $\beta > 2$, where β is a constant and let $T^2 \leq cd_{\max}$ for a constant $c < 1$. Then, the PoA is $O\left(T^{2(\beta-1)}\right)$.

5.4 The structure of NE in random graph models

We now analyze the structure of Nash equilibria in EC games in different random graph models.

5.4.1 The Erdős-Rényi model

In Erdős-Rényi random graph model $G(n, p)$, each pair of vertices has an edge between them with probability p. The spectral radius of $G(n, p) = (1 + o(1)) \max\{np, \sqrt{\Delta}\}$ [59].

Lemma 6. *For $p \geq \frac{c}{n}$, where c is a suitably large constant and $np \geq (1+\delta)T^2$ for any positive constant δ, the PoA for the EC game on $G \in G(n,p)$ is almost surely $O\left(\frac{np+\log n}{np}\right)$.*

Proof. We use an upper bound of n for the size of the secured set corresponding to the worst NE. Let $S_{\min}$ be the size of the secured set which corresponds to the best NE. We need to only show that $|S_{\min}| = \Omega\left(\frac{n^2p}{np+\log n}\right)$ a.s. The argument is on the lines of the proof of Lemma 5. First of all, (using Chernoff bound, we can show that) the number of edges in G is at least $\frac{n^2p}{2}(1-o(1))$ for a sufficiently large c a.s.

Let $G' = G[V \backslash S_{\min}]$ correspond to the graph induced by the insecure nodes; clearly, $\lambda_1(G') < T$. Since $\lambda_1(G') \geq \sqrt{\Delta(G')}$, no vertex in G' can have degree T^2 or more. Hence, the total number of edges in G' is at most $\frac{nT^2}{2}$. This implies that the number of edges to be removed from G is $r_e \geq \frac{n^2p}{2}(1-o(1)) - \frac{nT^2}{2} = \Theta(n^2p)$, the last expression following from the assumption that $np \geq (1+\delta)T^2$. Note that $|S_{\min}| \geq \frac{r_e}{\Delta(G)}$. Using Chernoff bound it follows that $\Delta(G) \leq c'(np+\log n)$ for some constant c'. Hence, $|S_{\min}| = \Omega\left(\frac{n^2p}{np+\log n}\right)$. □

Note that the connectivity threshold in $G(n,p)$ is $p = \Omega(\log n/n)$. This implies that if $np \geq (1+\delta)T^2$, any NE in the $G(n,p)$ model has cost $\Omega(n)$ above the connectivity threshold and therefore, the PoA is $\Theta(1)$.

5.4.2 Random power law graphs

In this chapter we consider a random graph model by Chung and Lu [23, 24]. Given a weight sequence $\mathbf{w} = (w(v_1,V), w(v_2,V), ..., w(v_n,V))$ for nodes $v_i \in V$, the Chung-Lu model $G(\mathbf{w})$ defines the random graph $G = (V,E)$ as follows: for every pair $v_j, v_k \in V$, v_j is adjacent to v_k with probability $\frac{w(v_j,V)w(v_k,V)}{\sum_{v_i \in V} w(v_i,V)}$. We define some of the notation that will be used throughout the section (these are summarized in Table 10.1):

It follows from [23, 24], that the expected degree of any node v_i equals its weight $w(v_i,V)$. Further, the degrees are concentrated around the expectation, as shown below:

$$|d(v,V) - w(v,V)| \leq 2(\sqrt{w(v,V)logn} + logn),\ \forall v \in V \tag{5.2}$$

We now state some additional properties of the Chung-Lu model, which are needed in the rest of the section.

Lemma 7. *([24]) For a random graph G in $G(\mathbf{w})$ and V' a subset of vertices, the induced subgraph $G[V']$ on V' is a random graph in $G(\mathbf{w}')$ where $\forall v \in V'$, $w'(v,V') = w(v,V)\frac{Vol(V')}{Vol(V)}$, and subsequently, $w_{\max}(V') = w_{\max}(V)\frac{Vol(V')}{Vol(V)}$.*

$G = (V, E)$	An instance of $G(\mathrm{w})$
V'	Any subset of V, $V' \subseteq V$
$G[V']$	Subgraph of $G = (V, E)$ induced on V'
$w(v, V')$	Weight of node $v \in V'$ in graph $G[V']$
$d(v, V')$	Degree of node $v \in V'$ in graph $G[V']$
$V_{\leq x}$	Set of nodes, $\{v : v \in V, w(v, V) \leq x\}$
$V_{>x}$	Set of nodes, $\{v : v \in V, w(v, V) > x\}$
$w_{\max}(V')$	Maximum weight of a node in $G[V']$
$d_{\max}(V')$	Maximum degree of a node in $G[V']$
$w_{\min}(V')$	Minimum weight of a node in $G[V']$
$vol(V')$	Volume of V' in $G = (V, E)$. $vol(V') = \sum_{v \in V'} d(v, V)$
$Vol(V')$	Expected volume of V' in $G = (V, E)$. $\mathrm{Vol}(V') = \sum_{v \in V'} w(v, V)$
$w(V')$	Expected average degree of nodes in $G[V']$. $w(V) = \frac{\sum_{v \in V} w(v,V)}{\lvert V \rvert}$
$\tilde{w}(V)$	Second order average degree of nodes in $G[V']$. $\tilde{w}(V) = \frac{\sum_{v \in V} w(v,V)^2}{\sum_{v \in V} w(v,V)}$

Table 5.1: Common notations used in section 5.4.2

Lemma 8. *([24]) For a graph $G = (V, E)$ in $G(\mathbf{w})$, if the maximum degree $d_{\max}(V)$ and second order average degree $\tilde{w}(V)$ satisfy $\sqrt{d_{\max}(V)} > \tilde{w}(V) log^2 n$, then the spectral radius of G is almost surely $(1 + o(1))\sqrt{d_{\max}(V)}$.*

Lemma 9. *For a random graph G in $G(\mathbf{w})$ and any $V' \subset V$, $\tilde{w}(V') < \tilde{w}(V)$.*

Proof. We know that,

$$\tilde{w}(V) = \frac{\sum_{v \in V} w(v, V)^2}{\sum_{v \in V} w(v, V)}$$

Therefore, for $G[V']$,

$$\tilde{w}(V') = \frac{\sum_{v \in V'} w(v, V)^2 \frac{\mathrm{Vol}(V')^2}{\mathrm{Vol}(V)^2}}{\sum_{v \in V'} w(v, V) \frac{\mathrm{Vol}(V')}{\mathrm{Vol}(V)}} = \frac{\mathrm{Vol}(V')}{\mathrm{Vol}(V)} \frac{\sum_{v \in V'} w(v, V)^2}{\sum_{v \in V'} w(v, V)}$$

$$= \frac{\mathrm{Vol}(V')}{\mathrm{Vol}(V)} \frac{\sum_{v \in V'} w(v, V)^2}{\mathrm{Vol}(V')} = \frac{\sum_{v \in V'} w(v, V)^2}{\mathrm{Vol}(V)} < \tilde{w}(V)$$

□

Lemma 10. *Let G be a Chung-Lu power law graph with exponent $\beta > 2$ and let $x \leq c w_{\max}$ for a constant $c < 1$. Then, (1) $|V_{\geq x}| = \Theta\left(n/x^{\beta-1}\right)$ and (2) Vol$(V_{\geq x}) = \Theta\left(n/x^{\beta-2}\right)$.*

Proof. We will only prove Case (1); the proof for Case (2) follows on similar lines.

$$|V_{\geq x}| = \sum_{i \geq x}^{w_{\max}} n_i = \Theta(1) \int_x^\infty \frac{n}{z^\beta} dz = \Theta\left(\frac{n}{x^{\beta-1}}\right).$$

□

Lemma 11. *Let G be a Chung-Lu power law graph with exponent $\beta > 2$ and minimum weight a constant, and let $x \leq c w_{\max}$ for a constant $c < 1$. Then, $w_{\max}(V_{\leq x}) = x\left(1 - \theta\left(\frac{1}{x^{\beta-2}}\right)\right)$*

Proof. From Lemma 7, it follows that

$$w_{\max}(V_{\leq x}) = x \frac{\mathrm{Vol}(V_{\leq x})}{\mathrm{Vol}(V)} = x\left(1 - \frac{\mathrm{Vol}(V_{\geq x(1+o(1))})}{Vol(V)}\right).$$

From lemma 10 and the assumption that minimum weight is a constant it follows that, $\frac{\mathrm{Vol}(V_{\geq x(1+o(1))})}{Vol(V)} = \frac{\Theta(n/x(1+o(1))^{\beta-2})}{\Theta(n)} = \Theta(\frac{1}{x^{\beta-2}})$. Therefore the result follows. □

Bounds for Best NE

Lemma 12. *Let $G(\mathbf{w})$ be a Chung-Lu random power law graph on n nodes with power law exponent $\beta > 2$ and $w(V)$ a constant. Let T be the epidemic threshold such that $w_{\max}(V) \geq (1+\delta)T^2$ for any positive constant δ and $T = \Omega(logn)$. The minimum cost of any NE, C_{OPT}, is $\Omega(\frac{n}{T^{2(\beta-1)}})$.*

Proof. Let $S_{\mathrm{OPT}} \subseteq V$ correspond to a minimum cost secured set. Let $b = (1+\epsilon)T^2$ for some positive constant $\epsilon < \min(\delta, 1)$. We will show that $|S_{\mathrm{OPT}}| \geq |V_{>b}|$. From Lemma 10, $|V_{>b}| = \Theta(\frac{n}{T^{2(\beta-1)}})$ and therefore, the result follows.

Suppose $|S_{\mathrm{OPT}}| < |V_{>b}|$; let $x \in V_{>b} \setminus S_{\mathrm{OPT}}$. Also, let $V' = V \setminus S_{\mathrm{OPT}}$. Observing that for any set of size $\leq |V_{>b}|$, $V_{>b}$ has the highest volume,

$$\begin{aligned} w_{\max}(V') &\geq w(x, V) \frac{\mathrm{Vol}(V \setminus V_{>b})}{\mathrm{Vol}(V)} \\ &> b \frac{\mathrm{Vol}(V_{\leq b})}{\mathrm{Vol}(V)} \geq w_{\max}(V_{\leq b}). \end{aligned}$$

Using Lemma 11, and the assumptions that $w(V)$ is a constant and $T^2 = \Omega(\log^2 n)$,

$$w(x, V_{\leq b}) = b\left(1 - \theta\left(\frac{1}{b^{\beta-2}}\right)\right) = (1+\epsilon)T^2(1 - o(1)).$$

Therefore, $w_{\max}(V') \geq (1+\epsilon')T^2$, for some constant ϵ'. From (5.2) and again the fact that $T^2 = \Omega(\log^2 n)$, it follows that the maximum degree of a vertex in V', $d_{\max}(V') \geq w_{\max}(V')(1-o(1)) > T^2$ a.s. and therefore, $\lambda_1(G[V']) \geq \sqrt{d_{\max}(V')} > T$, a contradiction to the initial assumption that S_{OPT} corresponds to a NE. Hence proved. □

Lemma 13. *Let $G(\mathbf{w})$ be a Chung-Lu random power law graph on n nodes with power law exponent $\beta > 2$ and $w(V)$ a constant. Let T be the epidemic threshold such that $w_{\max}(V) \geq (1+\delta)T^2$ and $T \geq (1+\gamma)\tilde{w}(G)log^2 n$ for any positive constants δ and γ. Then the social cost of the NE is $O\left(\frac{n}{T^{2(\beta-1)}}\right)$.*

Proof. Let $a = (1-\epsilon)T^2$ for a suitably small positive constant $\epsilon < 1$. Recall that $V_{\leq a} = \{v \mid w(v, V) \leq a\}$. Using Lemma 11, and the assumptions that $w(V)$ is a constant and $T^2 = \Omega(\log^4 n)$,

$$\begin{aligned} w_{\max}(V_{\leq a}) \leq a\left(1 - \theta\left(\frac{1}{a^{\beta-2}}\right)\right) &= (1-\epsilon)T^2(1-o(1)) \\ &\leq (1-\epsilon')T^2, \end{aligned} \tag{5.3}$$

for some constant ϵ'. From (5.2) and again the fact that $T^2 = \Omega(\log^4 n)$, it follows that the maximum degree of a vertex in $V_{\leq a}$, $d_{\max}(V_{\leq a})$ is bounded as follows:

$$w_{\max}(V_{\leq a})(1-o(1)) \leq d_{\max}(V_{\leq a}) \leq w_{\max}(V_{\leq a})(1+o(1)),$$

and using (5.3), we have $(1-\epsilon'')T^2 \leq d_{\max}(V_{\leq a}) \leq (1-\epsilon''')T^2$ almost surely for some positive constants ϵ'', ϵ'''. Recall that we can have ϵ as small as possible; we choose it such that $(1+\gamma)\sqrt{1-\epsilon''} \geq 1$. This implies that $\sqrt{d_{\max}(V_{\leq a})} \geq \sqrt{1-\epsilon''}T \geq \tilde{w}(G)\log^2 n$. Now applying Lemma 8, it follows that $\lambda_1(G[V_{\leq a}]) \leq (1-\epsilon''')T(1+o(1)) < T$. By Observation 4, it follows that there exists a NE of size $O\left(\frac{n}{T^{2(\beta-1)}}\right)$. □

Bounds for worst NE

Lemma 14. *In a Chung-Lu random power law graph $G(\mathbf{w})$ of n nodes and power law exponent $\beta \geq 2$ and $w(V)$ a constant, if $w_{\max}(G) \geq (1+\delta)T^2 w(V)$ for any positive constant δ and $T = \Omega(log^2 n)$, then the lower bound of the social cost of any NE is $\theta(n)$. Therefore, the size of the largest vaccinated set corresponding to NE is $\Theta(n)$.*

Proof. Recall that $V_{>a} = \{v \mid w(v, V) > a\}$. Let a be such that $w_{\max}(V_{>a}) = (1-\epsilon)T$ for a positive constant $\epsilon < 1$. Using Lemma 7, we have $\mathrm{Vol}(V_{>a}) = \mathrm{Vol}(V)\frac{(1-\epsilon)T}{w_{\max}(V)}$. Since $T = \Omega(\log^2 n)$, using (5.2), $d_{\max}(V_{>a}) \leq (1-\epsilon)T(1-o(1)) < T$ a.s. Hence, $\lambda_1(G[V_{>a}]) < T$ a.s. From Observation 2, it follows that $V_{\leq a}$ contains a subset which corresponds to a NE. Next we show that the smallest such subset is of size $\Theta(n)$.

Let V' be any set which contains $V_{>a}$. This implies that V' contains a vertex with weight $w_{\max}(V)$. Let $\mathrm{Vol}(V') = \mathrm{Vol}(V)\frac{(1+\epsilon')T^2}{w_{\max}(V)}$ for a suitably small positive constant $\epsilon' < \delta$. Then, from Lemma 7, $w_{\max}(V') = (1+\epsilon')T^2$. Since $T = \Omega(\log^2 n)$, it follows from (5.2) that $d_{\max}(V') > T^2$ and therefore, $\lambda_1(G[V']) > T$.

Suppose $V_r \subset V_{\leq a}$ such that $S_{\min} = V_{\leq a} \setminus V_r$ is a smallest secured set corresponding to NE. From the above discussions, it follows that $\text{Vol}(V_r) \leq \text{Vol}(V') - \text{Vol}(V_{>a})$. Recall that $w_{\min}(V) = \min_{v \in V} w(v, V)$. Clearly, $|V_r| \leq \frac{\text{Vol}(V_r)}{w_{\min}(V)}$ and therefore,

$$|V_r| \leq \frac{\text{Vol}(V') - \text{Vol}(V_{>a})}{w_{\min}(V)} \leq \frac{\text{Vol}(V)T}{w_{\max}(V)} \left((1+\epsilon')T - (1-\epsilon)\right)$$
$$\leq cn,$$

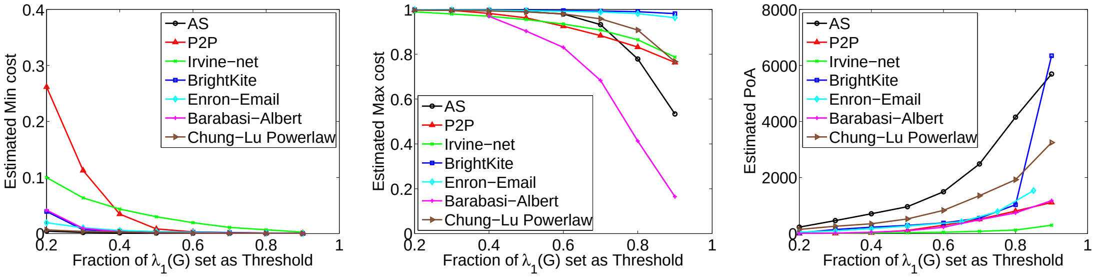
for some constant $c < 1$. This follows from the fact that $\text{Vol}(V) = n \cdot w(V)$ and $w_{\max} > (1+\delta)T^2 w(V)$ and $\epsilon' < \delta$.

From Lemma 10, $\text{Vol}(V_{>a}) = \Theta\left(\frac{n}{a^{\beta-2}}\right)$ and therefore, $a \geq \left(\frac{c_1 w_{\max}(V)}{T}\right)^{\frac{1}{\beta-2}}$ for some constant c_1. Since, $w_{\max}(V) > T^2$ and $T = \Omega(\log^2 n)$, it follows that $a = \Omega\left(\log^{\frac{2}{\beta-2}} n\right)$. Hence, again from Lemma 10, $|V_{>a}| = o(1) \cdot n$. Therefore, $|S_{\min}| = |V_{\leq a}| - |V_r| = (1 - o(1) - c)n = c'n$ for some constant $c' < 1$. Hence proved. □

Combining Lemmas 12 and 14, we have the following tight bound for the price of anarchy of the EC game.

Theorem 3. *Consider a Chung-Lu random power law graph $G(\mathbf{w})$ of n nodes and power law exponent $\beta > 2$ and $w(V)$ a constant, such that $w_{\max}(G) \geq (1+\delta)T^2 w(V)$ for any positive constant δ and $T = \Omega(log^2 n)$. The PoA of the* EC *game in $G(\mathbf{w})$ is $\theta(T^{2(\beta-1)})$ almost surely.*

5.5 Empirical results



(a) Estimated Minimum Cost (b) Estimated Maximum Cost (c) Estimated POA

Figure 5.2: Minimum cost, maximum cost and POA of NE (on y-axis) estimated with the *HDG* strategy and *LDG* strategies, as a function of T (x-axis). (a) and (b) show the minimum and maximum costs respectively normalized by network size, (c) shows the POA values. Results are shown for seven networks(see Table 5.2).

We now study characteristics of Nash equilibria of EC game in six networks, which includes five social/communication networks from the SNAP dataset [2] and from [90], and an instance of the Barabasi-Albert power-law graph model; these are listed in Table 5.2, along with some of their key properties. We study the costs of the cheapest and random NE, PoA, degree distributions in random NE, community structure and effects of Stackelberg strategies. We estimate the maximum and minimum NE cost by two heuristics, that we call the *High Degree*(HDG) and *Low Degree*(LDG) strategies. The HDG strategy is obtained by running the IterativeSecure procedure (Section 5.2) with π and ρ set to be permutations of nodes in non-increasing and non-decreasing degree orders, respectively; the LDG strategy is obtained similarly,by reversing π and ρ. From Section 5.3, it follows that the NE resulting from HDG has cost within a $\Theta(T^{\beta-1})$ factor of the social optimum in general power law graphs. From Section 5.4.2, it follows that the NE resulting from LDG is within a $\Theta(1)$ factor of the maximum cost NE in random power law graphs. Our main observations are:

1. We find that the estimated minimum cost of any NE scales as $\frac{cn}{T^{2c'(\beta-1)}}$ for constants c, c' in scale-free networks where c' is close to 1 (particularly $0.6 \leq c' \leq 1.4$), which is quite close to our theoretical bounds for general power law graphs (Section 5.3) and the Chung-Lu model (Section 5.4.2).

2. In all the networks, the PoA is an increasing function of T. Further, for networks with power law degree distributions, the estimated PoA scales as $cT^{2c'(\beta-1)}$ for constants c, c', where c' is close to 1 (particularly $0.6 \leq c' \leq 1.4$) which is also quite close to our analytical bounds.

3. The estimated size of the biggest NE in large networks turn out to be close to the network size for $0 < T < \frac{1}{2}\lambda_1(G)$.

4. The degree distribution of secured nodes in random NE seems quite close to that of the graph. Further, secured nodes tend to have higher degrees, compared to unsecured nodes.

5. Stackelberg strategies are quite effective in reducing the cost of random NE.

6. To get small equilibria, usually it suffices to secure nodes in a few "important" communities. Because, nodes contributing to high eigenvalue seem to be concentrated only in a few communities.

5.5.1 Estimates of the minimum and maximum NE cost

As shown in Figure 5.3, the minimum cost of NE i.e. the smallest size of secured node set in NE, decreases with T. Further, for the scale-free networks, we find that the best fit for the social optimum cost scales as $\frac{cn}{T^{2c'(\beta-1)}}$ for constants c, c', where c' is close to 1 (the

Graph,G	Description	Nodes,n	$\lambda_1(G)$	$\Delta(G)$	β
AS	Autonomous System peering info graph(Oregon-1 route views)	10670	58.72	2312	2.23
P2P	Peer-to-peer network, Gnutella-6	8717	22.38	115	NA
Irvine-net	Facebook like social network at UC Irvine	1899	48.14	255	1.34
Brightkite	Location based online social network	58228	101.49	1134	2.01
Enron-email	Enron email communication network	36692	118.42	1383	1.86
Barabasi-Albert	Synthetic scale-free graph	5000	12.51	151	2.61
Chung-Lu Powerlaw	Synthetic power law graph	4066	28.95	643	2.5

Table 5.2: Network graphs (five real and two synthetic)[2][90] used for experimenting EC game. For each network the number of nodes, spectral radius, maximum degree and power-law exponent (if a scale free network) are listed.

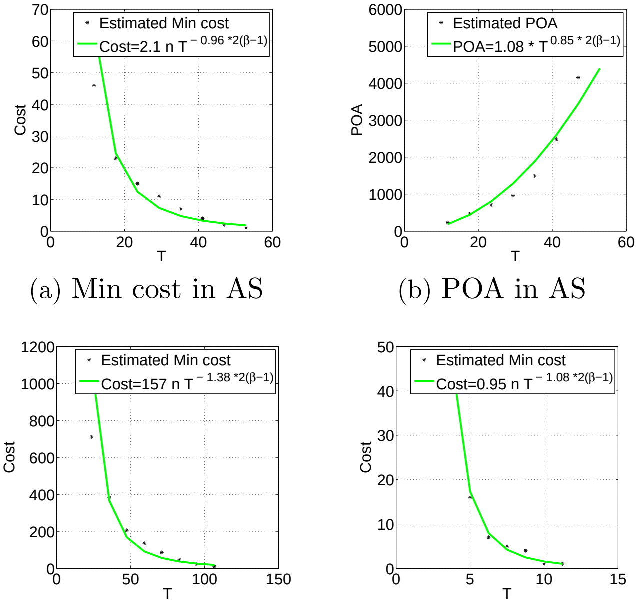
power law exponent β is also estimated by best fit); this is consistent with our analytical bounds for general graphs (Section 5.3) and the Chung-Lu model (Section 5.4.2). Further, the maximum NE cost is close to n for $T < \frac{1}{2}\lambda_1(G)$ and decreases very slowly with T as shown in Figure 5.2b; this suggests that most of the nodes in large networks end up securing themselves in the worst equilibrium for small T.

5.5.2 Estimates of the Price of Anarchy (PoA)

We compute a lower bound on the PoA by taking the ratio of estimated maximum and minimum social cost of NE found from *LDG* and *HDG* strategies respectively. As Figure 5.3 suggests, the PoA (estimated) increases as threshold is increased. Furthermore, for scale free networks, the best fit for the PoA scales as $cT^{2c'(\beta-1)}$, for constants c, c' (c' being close to 1), as illustrated in Figure 5.3.

5.5.3 Degree Distribution in random NE

We compute random equilibria by invoking IterativeSecure with two random node permutations π, ρ. We examine the degree distribution of secure and insecure nodes in random equilibria. As shown in fig 5.4, both the secure and insecure nodes follow roughly similar degree distribution patterns. However, secure nodes are distributed slightly more on higher degrees and slightly less on lower degrees compared to the distribution of insecure nodes. This is an intuitive observation, since higher degree nodes contribute more to the decrease of spectral radius.



(a) Min cost in AS

(b) POA in AS

(c) Min cost in Enron-email

(d) Min cost in Barabasi-Albert

Figure 5.3: Minimum cost of NE and POA (on y-axis) estimated with the *HDG* and *LDG* strategy,as a function of T (x-axis), along with the best fit function of the form $\frac{cn}{T^{2c'(\beta-1)}}$ and $kT^{2k'(\beta-1)}$ respectively, where c' and k' are constants close to 1 (see n, β values in Table 5.2).

5.5.4 Impact of Stackelberg strategies

We now consider Stackelberg strategies [104] for mitigating the effects of decentralized control. Here, a small fraction of nodes are secured in a centralized manner (as part of the Stackelberg strategy), and the rest of the nodes act non-cooperatively. We consider a specific Stackelberg intervention strategy in which different fractions of high degree nodes are secured initially, and study the impact on the worst cost NE and the PoA. The results in Figure 5.5 suggest that PoA is reduced very slowly for high degree node intervention until the intervened node set itself make an efficient NE. However, such intervention has more impact on random NE's and PoA falls comparatively more quickly as high degree nodes are secured with Stackelberg intervention.

5.5.5 Analyis of the community structure in NE

We study the community structure (using the approach of [14]), and find that secured nodes are distributed very unevenly among communities. Figure 5.6 shows the fraction of secure nodes in different communities in the P2P network, for equilibria computed by the *HDG* and *HEC* strategies; these two strategies are obtained by running the IterativeSecure

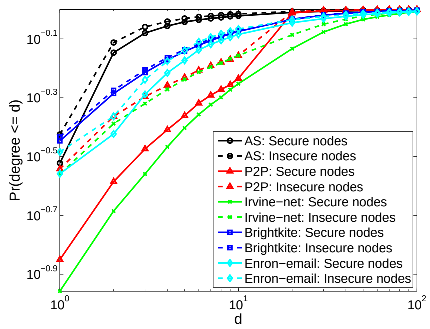
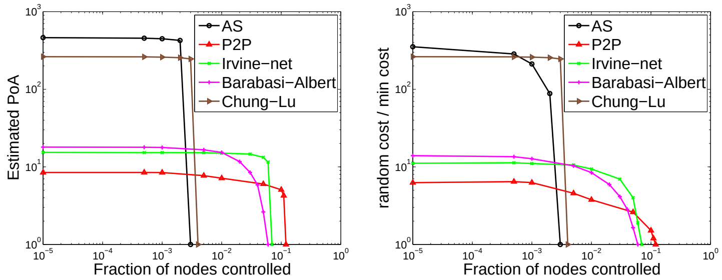
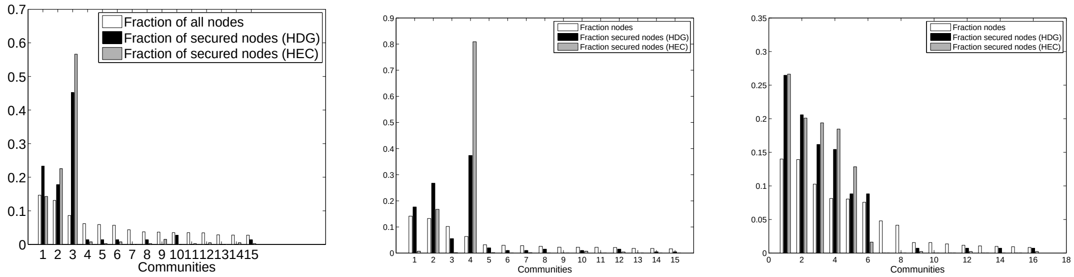


Figure 5.4: Cumulative degree distribution of secure and insecure nodes in random NE. It plots for different networks, the probability (on y-axis) that a secure or insecure node in random NE has degree less than or equal to d (x-axis). The plot is generated from 30 random NE's.



(a) Effects on PoA (b) Effects on $\frac{\text{Random NE cost}}{\text{Min NE cost}}$

Figure 5.5: Effects of Stackelberg strategies on inefficiency metrics. Plots in (a) and (b) show the change in PoA and $\frac{\text{random NE cost}}{\text{Min NE cost}}$ (on y-axis) as different fractions (x-axis) of high degree nodes are secured as part of Stackelberg strategy. Results are shown for four networks and threshold set to $T = 0.3\lambda_1(G)$.



(a) P2P Gnutella-6 (b) BrightKite (c) Enron

Figure 5.6: Secured nodes in the smallest NE (estimated with the *HDG* and *HEC* strategy) as distributed among the communities of P2P, Brightkite, Enron network. For each of the biggest 15 communities, the left bar shows the fraction of nodes that the community has. The other two bars show the fraction of secured nodes each community has. The middle and the right bar show them for NE's computed with the *HDG* and *HEC* strategy respectively. In to both the NE's, secured nodes are concentrated in few "important" communities.

procedure (Section 5.1) with the permutation π being the decreasing order of nodes based on their degree and eigenvector components, respectively. The figure shows that larger communities have disproportionately higher fraction of secure nodes. Similar pattern has been found to hold for other networks as well [105].

5.6 Conclusion

Our EC game formulation allows for a tractable way to incorporate realism in both the network and disease models; this is a natural game-theoretic analogue of the approaches to reduce the spectral radius to control epidemics [115, 79]. Our rigorous results show that the the PoA is high in heterogeneous networks; our experiments suggest that some of the effects of non-cooperative decision making can be mitigated by Stackelberg strategies. The main technical contribution in this chapter is the analysis of the rich network effects in the structure of equilibria, which might give further insights to understanding the incentives for individuals to secure themselves, and to affect it. We find it interesting that our empirical results on several real and random networks corroborate well with our analytical results. The spectral properties of general and random graphs that we find would be useful in future studies of the epidemic processes in these networks.

Chapter 6

Epidemic Containment with Centralized Strategies

In this chapter we focus on the optimum cost of vaccination intervention. In the previous chapter, we have studied the cost of decentralized vaccination in a game theoretic setting. In this chapter we present centralized approximation algorithms for finding the social optimum cost. We show, the same algorithms work for edge removal strategies as well. We show provable bounds on the performance these algorithms.

Given a contact network, which nodes or contacts should we remove to contain the spread of a virus? Equivalently, in a computer network, which connections should we cut to prevent the spread of malware? Designing effective and low cost interventions are fundamental challenges in public health and network security. The spectral characterization of the SIS model of epidemic spread [40, 122, 95], as discussed in chapter 1, motivates the following strategy for controlling an epidemic: remove edges (quarantining) or nodes (vaccinating) to reduce the spectral radius below a threshold T—-we refer to this as the spectral radius minimization (SRM) problems, with variants depending on whether edges are removed (the SRME problem) or whether nodes are removed (the SRMN problem). Van Mieghem et al. [79] and Tong et al. [115] prove that this problem is NP-complete. They also study two heuristics for it, one based on the components of the first eigenvector (EigenScore) and another based on degrees (ProductDegree). However, no rigorous approximations were known for the SRME or the SRMN problems.

Our main contributions.

1. Lower bounds on the worst-case performance of heuristics: We show that the ProductDegree, EigenScore and Pagerank heuristics (defined formally in Section 10.1) can perform quite poorly in general. We demonstrate graph instances where these heuristics give solutions of cost $\Omega(\frac{n}{T^2})$ times the optimal, where n is the number of nodes in the graph.

2. Provable approximation algorithms: We present two bicriteria approximation algorithms for the SRME and SRMN problems, with varying approximation quality and running time tradeoffs. Our first algorithm, GreedyWalk, is based on hitting closed walks in G. We show this algorithm has an approximation bound of $O(\log n \log \Delta)$ times optimal for the cost of edges removed, while ensuring that the spectral radius becomes at most $(1+\epsilon)$ times the threshold, for ϵ arbitrarily small (here Δ denotes the maximum node degree in the graph). We also design a variant, GreedyWalkSparse, that performs careful sparsification of the graph, leading to similar asymptotic guarantees, but better running time, especially when the threshold T is small. We then develop algorithm Primal-Dual, which improves this approximation bound to an $O(\log n)$ using a more sophisticated primal-dual approach, at the expense of a slightly higher (but polynomial) running time.

3. Extensions: We consider two natural extensions of the SRME problem: (i) non-uniform transmission rates on edges and (ii) node version SRMN. We show that our methods extend to these variations too.

4. Empirical analysis: We conduct an extensive experimental evaluation of GreedyWalk, a simplified version of Primal-Dual and different heuristics that have been proposed for epidemic containment on a diverse collection of synthetic and real networks. These heuristics involve picking edges $e = (i, j)$ in non-increasing order of some kind of score; the specific heuristics we compare include: (i) ProductDegree, (ii) EigenScore, (iii) LinePagerank, and (iv) Hybrid, which picks the edge based on either the eigenscore, degree ordering, depending on the maximum decrease in eigenvalue. We find that GreedyWalk performs the better than all the heuristics in all the networks we study. Our analysis of GreedyWalk is for walks of length $k = \Theta(\log n)$; in practice, we find the performance degrades significantly as k is reduced.

Organization. The background and notation are defined in Section 10.1. The GreedyWalk and GreedyWalkSparse algorithms are discussed in Section 6.2, and the Primal-Dual algorithm is discussed in Section 6.4. The lower bounds for some heuristics are discussed in Section 6.6, and experimental results are summarized in Section 6.7. Finally, we conclude in Section 6.8.

6.1 Preliminaries

We consider undirected graphs $G = (V, E)$, and interventions to control the spread of epidemics— vaccination (modeled by removal of nodes) and quarantining (modeled by removal of edges). There can be different costs for the removal of nodes and edges (denoted by $c(v)$ and $c(e)$, respectively), e.g., depending on their demographics, as estimated by [77]. For a set $E' \subset E$, $c(E') = \sum_{e \in E'} c(e)$ denotes the total cost of the set E' (similarly for node subsets).

There are a number of models for epidemic spread; we focus on the fundamental SIS (Susceptible-Infectious-Susceptible) model, which is defined in the following manner. Nodes are in susceptible (S) or infectious (I) state. Each infected node u (in state I) causes each susceptible neighbor v (in state S) to become infected at rate β_{uv}. Further, each infected node u switches to the susceptible state at rate δ. In most of the chapter, we assume a uniform rate $\beta_{uv} = \beta$ for all $(u, v) \in E$; in this case, we define a threshold $T = \beta/\delta$, which characterizes the time to extinction. Let $A = A^G$ denote the adjacency matrix of G, and let $n = |V|$. Let $\lambda_i(G)$ denote the ith eigenvalue of A, and let $\rho(A) = \max_i \lambda_i(A)$ denote the spectral radius of A. Since G is undirected, it follows that all eigenvalues are real, and $\rho(A) > 0$ (see, e.g., [19]). Ganesh et al. [40] showed that the epidemic dies out in time $O(\frac{\log n}{1-\rho(A)/T})$, if $\rho(A) < T$ in the SIS model, with high probability; this threshold was also observed by [122]. Prakash et al. [95] show this condition holds for a broad class of other epidemic models, including the SIR model (which contains the 'Recovered' state).

We consider the following problems in this chapter, which are defined in terms of the type of elements removed (i.e., edges or nodes).

Definition 1. Spectral Radius Minimization (SRME) problem Given an undirected graph $G = (V, E)$, with cost $c(e)$ for each edge $e \in E$, and a threshold T, the goal of the SRME$(G, c(\cdot), T)$ problem is to find the cheapest subset $E' \subseteq E$ such that $\lambda_1(G[E \backslash E']) < T$.

Similarly, we also consider the node version of this problem, denoted by SRMN. We use $E_{\mathrm{OPT}}(T)$ to denote an optimal solution to the SRME$(G, c(\cdot), T)$ problem.

We discuss some notation that will be used in the rest of the chapter. Let $\mathcal{W}_k(G)$ denote the set of closed walks of length k in G; let $W_k(G) = |\mathcal{W}_k(G)|$. For a walk w, let nodes(w) denote the number of distinct nodes in w. A standard result [19] is the following:

$$\sum_{w \in \mathcal{W}_k(G)} \mathrm{nodes}(w) = \sum_i A^k_{ii} = \sum_{i=1}^n \lambda_i(G)^k. \tag{6.1}$$

The number of walks in $\mathcal{W}_k(G)$ containing a node i is A^k_{ii}. For a graph G, let walks(e, G, k) denote the number of closed k-walks in G containing $e = (i, j)$. Similarly, let walks(v, G, k) denote the number of closed k-walks in G containing node v. Then, walks$(e, G, k) = A^{k-1}_{ij}$, and walks$(i, G) = A^k_{ii}$. We say that an edge set E' hits a walk w if an edge participating in w belongs to E'.

The following bounds for the first eigenvalue are used frequently in the chapter (see [78]):

- $\sqrt{\Delta(G)} \leqslant \lambda_1(G) \leqslant \Delta(G)$
- $\lambda_1(G) \leqslant \max_{u \sim v} \sqrt{\mathrm{d}(u, G)\mathrm{d}(v, G)}$, where $u \sim v$ means u is adjacent to v.

6.2 GreedyWalk: $O(\log n \log \Delta)$-approximation

We first describe a bicriteria approximation algorithm for SRME that achieves an $O(\log n \log \Delta)$ approximation in edge removal cost, while exceeding the spectral threshold by at most a factor of $(1+\epsilon)$, for $\epsilon > 0$ that can be made arbitrarily small. We then extend the algorithm and the bounds to the node version, and the versions for labeled graphs. Our algorithms are based on a greedy approach, that exploits the relationship (6.1), and reduces this problem to a partial covering problem.

Algorithm 1: Algorithm GreedyWalk (high level description)

```
input  : $G, T, c(\cdot)$
output: Edge set $E'$
Initialize $E' \leftarrow \phi$
while $W_k(G[E \setminus E']) \geq nT^k$ do
    $r \leftarrow W_k(G[E \setminus E']) - nT^k$
    Pick $e \in E \setminus E'$ that maximizes $\frac{\min\{r, \text{walks}(e, G[E \setminus E'], k)\}}{c(e)}$ $E' \leftarrow E' \cup \{e\}$.
end
```

Lemma 15. *Let E' denote the set of edges found by Algorithm* GreedyWalk*. Given any constant $\epsilon > 0$, for $k = \frac{\log n}{\log(1+\epsilon)}$, we have $\lambda_1(G[E \setminus E']) \leq (1+\epsilon)T$, and $c(E') = O(c(E_{\text{OPT}}(T)) \log n \log \Delta)$.*

Proof. Let $G' = G[E \setminus E']$ denote the graph resulting after the removal of edges in E'. We first prove the bound on $\lambda_1(G')$. Since E' hits at least $W_k(G) - nT^k$, the number of closed k-walks in G', $W_k(G') \leqslant nT^k$. Since $\sum_{i=1}^n \lambda_i(G')^k = \sum_i A_{ii}^k = \sum_{w \in \mathcal{W}(G')} \text{nodes}(w) \leq kW_k(G')$, it follows that $\sum_{i=1}^n \lambda_i(G')^k \leqslant nkT^k$. Therefore, $\lambda_1(G') \leqslant e^{(\log n + \log k)/k} T \leqslant (1+\epsilon)T$ for $k \geqslant (\log n)/\log(1+\epsilon)$ and n sufficiently large.

Next, we derive a bound for $c(E')$. Following the standard analysis of greedy covering algorithms [112], we have $c(E') = O(c(E_{\text{HITOPT}}) \log |H|)$, where H denotes the number of elements in our covering instance, and E_{HITOPT} denotes the size of the optimum solution for this covering instance. Elements correspond to walks in $\mathcal{W}_k(G)$ in our covering instance, and it is easy to see that $H = |W_k(G)| \leqslant n\Delta^k$. We show below that $c(E_{\text{HITOPT}}) \leq c(E_{\text{OPT}}(T))$; it follows that $c(E') = O(E_{\text{OPT}}(T) \log n \log \Delta)$.

Finally, we prove that $c(E_{\text{HITOPT}}) \leq c(E_{\text{OPT}}(T))$. By definition of $E_{\text{OPT}}(T)$, we have $\lambda_1(G[E - E_{\text{OPT}}(T)]) \leq T$. Then,

$$W_k(G') \leq \sum_{i=1}^n \lambda_i(G')^k < nT^k,$$

and therefore, $E_{\text{OPT}}(T)$ hits at least $W_k(G) - nT^k$, which implies $c(E_{\text{HITOPT}}) \leq c(E_{\text{OPT}}(T))$. □

Effect of the walk length k. We set the walk length $k = c \log n$ in Algorithm GreedyWalk; understanding the effect of k is a natural question. From the proof of Lemma 15, it follows that $\lambda_1(G[E \setminus E'])$ can be bounded by $(nk)^{1/k}T$ for any choice of k, as long as it is even. This bound becomes worse as k becomes smaller, e.g., it is $O(\sqrt{n})$ for $k = 2$. This is borne out in the experiments in Section 6.7.

In order to complete the description of GreedyWalk (Algorithm 1), we need to design an efficient method to determine the edge which maximizes the quantity in line 4. We discuss two methods below.

6.2.1 GreedyWalk in small networks using matrix multiplication

Note that walks$(e, G, k) = A_e^{k-1}$. We use matrix multiplication to compute A^{k-1} once for each iteration of the while loop in line 2 of Algorithm 1. In line 4, we iterate over all edges, in order to compute the edge e that maximizes the given ratio. For $k = O(\log n)$, A^{k-1} can be computed in time $O(n^\omega \log \log n)$, where $\omega < 2.37$ is the exponent for the running time of the best matrix multiplication algorithm [124]. Therefore, each iteration involves $O(n^\omega \log \log n + m) = O(n^\omega \log \log n)$ time. This gives a total running time of $O(n^\omega \log \log n \mathrm{OPT} \log^2 n)$, since only $O(\mathrm{OPT} \log^2 n)$ edges are removed. One drawback with this approach is the high (superlinear) space complexity, even with the best matrix multiplication methods, in general.

6.2.2 GreedyWalk in large sparse networks using a dynamic programming approach

When the graphs are very sparse, with $\Theta(n)$ edges, we adapt a dynamic programming approach to compute walks(e, G, k) more efficiently and compute the the edge that maximizes walks$(e, G[E \setminus E'], k)/c(e)$ in line 4 of Algorithm 1. Let $H_{\overrightarrow{uv}}(G, x, l)$ denote the number of walks of length l from node u and edge (u, v) as the first edge to node x in G. This means that $H_{\overrightarrow{uv}}(G, u, k) = $ walks(e, G, k). As a first step towards this, Algorithm 2 describes how to compute $H_{\overrightarrow{uv}}(G, x, l)$.

```
Algorithm 2: ClosedWalkE($G, k$)
input  : $G, (u, v), k \geq 2$
output: Number of closed walks of length $k$ in $G$ containing $(u, v)$
Initialize $H_{\overrightarrow{uv}}(G, v, 1) = 1$, $H_{\overrightarrow{uv}}(G, x, 1) = 0, \forall x \in V \setminus \{v\}$
for $l = 2$ to $k$ do
    $H_{\overrightarrow{uv}}(G, x, l) = \sum_{y \in N_G(x)} H_{\overrightarrow{uv}}(G, y, l-1), \forall x \in V(G)$
end
return $H_{\overrightarrow{uv}}(G, u, k)$
```

Next, we describe in Algorithm 3 how the greedy edge choice in line 4 of Algorithm 1 is implemented efficiently. We make use of the fact that walks$(e, G', k) \leq$ walks(e, G, k) for any $G' \subset G$. In every iteration of Algorithm 3, potentially, we need to update $f(\cdot)$ for all edges in $E \setminus E'$. However, in practice, we observe that the number of such updates is very small compared to $|E \setminus E'|$.

Algorithm 3: GreedyWalk with edge updates.

```
input  : $G, T, c(\cdot)$
output: Edge set $E'$
Initialize $E' \leftarrow \phi$ and $\forall e \in E$, let $f(e) = \text{walks}(e, G, k)$
while $W_k(G[E \setminus E']) \geq nT^k$ do
    Order edges of $E \setminus E'$ as $e_1, e_2, \cdots$, in the decreasing order of $f(e_i)$.
    $E' \leftarrow E' \cup \{e_1\}$
    for $j = 2, \ldots, |E \setminus E'|$ do
        Update $f(e_j) = \text{walks}(e_j, G[E \setminus E'], k)$.
        if $f(e_j) \geq f(e_{j+1})$ then
            Exit from the for loop.
        end
    end
end
```

Running time and space complexity: Let $V = V(G)$, $E = E(G)$ and $n = |V|$, $m = |E|$. Note that, ClosedWalkE(k) takes $2mk$ time; it computes walks(e, G, k). Therefore, computing walks(e, G, k) for all the edges takes $2m^2k = O(n^2k)$, assuming $m = \Theta(n)$ in real world networks. For computing $H_{uv}(G, x, l)$, $\forall x \in V$, ClosedWalkE(k) needs to look only at $H_{uv}(G, x, l-1)$, $\forall x \in V$. Therefore, the space complexity is $O(n)$. Using this method, walks(e, G, k) can be computed one edge at a time or in parallel.

6.3 Using sparsification for faster running time: Algorithm GreedyWalkSparse

The efficiency of Algorithm GreedyWalk can be improved if the number of edges in the graph can be reduced. We now discuss Algorithm GreedyWalkSparse that combines two pruning steps with GreedyWalk, leading to sparser graphs, without affecting the asymptotic approximation guarantees. The algorithm is given below. It refers to the T-core of a graph which denotes the the maximal subgraph of G with minimum degree T (see, e.g., [3]).

Lemma 16. *Let E_1 and E_2 denote the set of edges removed in the pruning steps* MaxDegreeReduction *and* DensityReduction*, respectively. Then, $c(E_1)$ and $c(E_2)$ are both*

Algorithm 4: Algorithm GreedyWalkSparse

```
input  : $G, T, c(\cdot)$
output: Edge set $E'$
Initialize $G_r = G$.
begin pruning step 1: MaxDegreeReduction
    Let $V_{T^2} = \{v : \mathrm{d}(v, G) \geqslant T^2\}$.
    for $v \in V_{T^2}$ do
        if $\mathrm{d}(v, G_r) \geq T^2$ then
            Let $e_{v,1}, \ldots, e_{v,\mathrm{d}(v,G_r)}$ be the edges incident
             on $v$ ordered so that $c(e_{v,1}) \leq \ldots c(e_{v,\mathrm{d}(v,G_r)})$. Let
             $E_v = \{e_{v,1}, \ldots, e_{v,\mathrm{d}(v,G_r)-T^2+1}\}$.
            $E_1 \leftarrow E_1 \cup E_v$ and $E(G_r) \leftarrow E(G_r) \setminus E_v$.
        end
    end
end
begin pruning step 2: DensityReduction
    Let $C_T$ denote the $T$-core of $G_r$.
    Order the edges $e_1, \ldots, e_{|E(C_T)|}$ in non-decreasing order of cost.
    $E_2 \leftarrow \{e_i \mid i \leq |E(C_T)| - T|V(C_T)|/2 + 1\}$
end
$E(G_r) \leftarrow E(G_r) - E_2$
$E_3 = \text{GreedyWalk}(G_r, T, c(\cdot))$
$E' \leftarrow E_1 \cup E_2 \cup E_3$
```

at most $2c(E_{\mathrm{OPT}}(T))$.

Proof. We have $\sqrt{\Delta(G')} \leqslant \lambda_1(G')$, as mentioned in Section 10.1, which implies $\Delta(G[E - E_{\mathrm{OPT}}(T)]) \leq T^2$. Therefore, $c(\{e \in N(v) \cap E_{\mathrm{OPT}}(T)\}) \geq \sum_{j=1}^{\mathrm{d}(v,G)-T^2+1} c(e_{v,j})$, where the sum is the minimum cost of edges that can be removed to ensure that the degree of v becomes at most T^2. Therefore,

$$\begin{aligned} c(E_1) &= \sum_{v \in V_{T^2}} \sum_{j=1}^{\mathrm{d}(v,G)-T^2+1} c(e_{v,j}) \\ &\leq \sum_{v \in V_{T^2}} c(\{e \in N(v) \cap E_{\mathrm{OPT}}(T)\}) \\ &\leq c(E_{\mathrm{OPT}}(T)) \end{aligned}$$

Recall that the second pruning step is applied on G_r. For bounding $c(E_2)$, we use another lower bound for λ_1: for any induced subgraph H of G_r, $\sum_{v \in V(H)} \frac{d(v,H)}{|V(H)|} \leq \lambda_1(G_r)$. Therefore, the existence of a T-core C_T implies that $\lambda_1(G_r) \geq T$. Since the average degree of C_T in the residual graph must be at most T, at least implies $|E(C_T)| - T|V(C_T)|/2 + 1$ edges must be removed from C_T. Therefore,

$$c(E_2) = \sum_{j=1}^{|E(C_T)|-T|V(C_T)|/2+1} c(e_j) \leq c(E_{\text{OPT}}(T) \cap E(C_T)) .$$

Hence proved. □

By Lemma 16, it follows that the approximation bounds of Lemma 15 still hold. However, the pruning steps reduce the number of edges, thereby speeding the implementation of GreedyWalk. We discuss the empirical performance of pruning in Section 6.7. Now we show that pruning also improves the approximation factor marginally from $O(\log n \log \Delta)$ to $O(\log n \log T)$ which could be significant when n is large and $T \ll \Delta$.

Lemma 17. *Let E' denote the set of edges found by Algorithm* GreedyWalkSparse*. Given any constant $\epsilon > 0$, for $k = \frac{\log n}{\log(1+\epsilon)}$, we have $\lambda_1(G[E \setminus E']) \leq (1+\epsilon)T$, and $c(E') = O(c(E_{\text{OPT}}(T)) \log n \log T)$.*

Proof. From Lemma 16, the number of edges removed is at most $2c(E_{\text{OPT}})$. The residual graph G_r has maximum degree less than T^2. Therefore, applying Lemma 15 on G_r, it follows that the number of edges removed is $O(c(E_{\text{OPT}}(T)) \log n \log T)$. Hence, the total number of edges removed by GreedyWalkSparse is at most $2c(E_{\text{OPT}}(T) + O(c(E_{\text{OPT}}(T)) \log n \log T) = O(c(E_{\text{OPT}}(T)) \log n \log T)$. □

6.4 Primal-Dual: $O(\log n)$-approximation

We use the approach of [39] to improve the approximation factor. The algorithm of [39] is a primal-dual approach, which gives an f-approximation for the partial covering problem, where f denotes the maximum number of sets that contain any element in the set system. In our reduction from the SRM problem to partial covering, elements correspond to all the closed walks of length $k = O(\log n)$, while sets correspond to edges; for an edge e, the corresponding set S_e consists of all the walks w that are hit by e. In this reduction, each walk w lies in k sets; therefore, $f = O(\log n)$ for this set system. Unfortunately, our set system has size $n^{O(\log n)}$, so that the algorithm of [39] cannot be used directly in polynomial time.

We now discuss how to adapt this algorithm to run in polynomial time. We only discuss the polynomial time implementation of the Primal-Dual subroutine of [39] in detail here. However, we also present the set cover algorithm HitWalks for completeness. This algorithm iterates over all edges and invokes Primal-Dual in each iteration to obtain a candidate set of edges to remove and finally chooses that set with minimum cost. $\mathcal{T}'$, $\mathcal{S}'$, c' and k' denote the set of elements (walks) to be covered, the sets (corresponding to edges that can be chosen), the costs corresponding to the sets/edges and the number of elements (walks) that need to be covered, respectively. A subset $C \subseteq \mathcal{S}'$ is k'-feasible if $|\cup_{S_e \in C} S_e| \geq k'$. Let $u(w)$ denote the dual variables corresponding to the walks w; these are not maintained in the algorithm explicitly, but assigned in the comments, for use in the analysis. This algorithm does not explicitly update the dual variables, but the edges are picked in the same sequence as in [39].

Algorithm 5: Primal-Dual$(\mathcal{T}', \mathcal{S}', c', k')$

```
output: Edge set $E''$
Initialize $z_e = 0$ for all $S_e \in \mathcal{S}'$, $C \leftarrow \phi$.
// $u(w) = 0$ for all walks $w$ in $G'$.
while $C$ is not $k'$-feasible do
    $x = \min_{e \in E \setminus E''}\{\frac{c_e - z_e}{\text{walks}(e,G')}\}$; let $e$ be an edge for which the minimum is reached.
    $C \leftarrow C \cup \{S_e\}$
    For each $e' \in E \setminus E''$: $z_{e'} = z_{e'} + x \cdot \text{walks}(e', G')$
    // $u(w) = u(w) + x$ for all walks $w$ in $G'$ that pass through $e'$
    $E' \leftarrow E'' \cup \{e\}$
end
```

Algorithm 6: HitWalks$(\mathcal{T}, \mathcal{S}, c, k)$

```
input  : Set of all $k$-closed walks $\mathcal{T}$, walks corresponding to edges $\mathcal{S}$, edge cost set $c$,
         number of walks to hit $k$
output: Edge set $E'$
Sort the edges of $G$ in increasing order of their costs.
Initialize $\forall j$, $c'(e_j) \leftarrow \infty$
for $j \leftarrow 1$ to $m$ do
    $c'(e_j) \leftarrow c(e_j)$ and compute $\text{walks}(e_j, G)$
    $cs_j \leftarrow \infty$. // cost of edge set in this iteration
    if $|S_1 \cup S_2 \cup \cdots S_j| \geqslant k$ then
        $E'_j = \{e_j\} \cup$ Primal-Dual$\big(\mathcal{T} \setminus S_j, \mathcal{S} \setminus S_j, c', k - \text{walks}(e_j, G)\big)$
        $cs_j = c(E'_j)$
    end
    $i = \min_j cs_j$
    $E' = E_i$
end
```

Lemma 18. *The dual variables $u(w)$ in algorithm* Primal-Dual *are maintained and updated as in [39], and the edge e picked in each iteration is the same. We have* $c(C) =$

$O(c(E_{OPT})\log n)$ and $\lambda_1(G[E-C]) \leq (1+\epsilon)T$, where, as before, $k = \log(n)/\epsilon$, and $\epsilon > 0$ can be set arbitrarily small.

Proof. Instead of updating the dual variable $u(w)$ for each element (walk) w, as done in [39], the variable z_e corresponding to each set (edge) e is updated in algorithm Primal-Dual at the end of each iteration. We will prove that, the following is an invariant at the end of each iteration, $z_e = \sum_{w \in S_e} u(w)$.

Then we will show that, a set e is picked into the cover in Primal-Dual, whenever $z_e = c_e$.

Therefore, increasing the $u(w)$'s has the same effect as increasing the z_e's in terms of picking the sets into the cover and both the algorithm Primal-Dual and the one in [39] chooses the same set in each iteration. □

6.5 Extensions

Our discussion so far has focused on the SRME problem. We now consider two extensions which capture two kinds of issues arising in practice.

1. *Non-uniform transmission rates*: In general, the transmission rate β is not constant for all the edges. The transmission rate β_{ij} for edge (i,j) depends on individual properties, especially the demographics of the end-points i and j, such as age, e.g., [77]. This gives us the SRME-nonuniform problem, which is defined later. We extend the spectral radius characterization of [40, 122, 95] to handle this setting, and show that GreedyWalk can also be adapted for solving SRME-nonuniform, with the same guarantees.
2. *The node removal version (*SRMN *problem)*: We extend the GreedyWalk algorithm in a natural manner to work for SRMN, with the same approximation guarantees.

6.5.1 Non-uniform transmission rates

Let $B = (\beta_{ij})$ denote the matrix of the transmission rates. We assume the rates are symmetric, i.e., $\beta_{ij} = \beta_{ji}$. In this case, the sufficient condition for the epidemic to die out is slightly different, and is stated below.

Lemma 19. *Let B be the matrix of transmission rates, and let δ be the recovery rate in the SIS model. If $\rho(B) < \delta$, the time to extinction, τ satisfies*

$$\text{Exp}[\tau] \leq \frac{\log n + 1}{\delta - \rho(B)}$$

For the case of uniform costs, i.e., $c(e) = 1$ for all edges e, this motivates the following problem:

Definition 2. SRME-NONUNIFORM problem Given an undirected graph $G = (V, E)$, with transmission rate β_{ij} for each $(i, j) \in E$ and recovery rate δ, find the smallest set $E' \subseteq E$ such that $\rho(B(G[E - E'])) \leq \delta$.

In this section, we use E_{OPT} to denote the optimum solution to SRME-NONUNIFORM(G, B, δ). Our algorithm GREEDYWALK-NONUNIFORM adapts GREEDYWALK to a weighted covering problem. We need to refine the definitions used earlier. For walk $w \in \mathcal{W}_k(G)$, let $f(w) = \prod_{e=(ij)\in E(w)} \beta_{ij}^{\text{count}(e,w)}$ denote its weight, where count(e, w) is the number of occurrences of edge e in walk w; for a set W' of walks, let $f(W') = \sum_{w\in W'} f(w)$ denote the total weight of W'. In the algorithm, we will need to compute $f(W_k(G))$, which is done by modifying the recurrence used in Algorithm COUNTWALKS(G) to compute $W_k(G)$:

$$f(W_k(G)) = B^k_{nn} + f(W_k(G[V - \{n\}]).$$

Let $f(e, G) = \sum_{w:e\in w} f(w)$ denote the total weight of walks containing edge e; $f(e, G) = B^k_e$. Algorithm GREEDYWALK-NONUNIFORM involves the following steps:

- $E' = \phi$
- while $f(W_k(G[E - E'])) \geq n\delta$:
  - Pick the $e \in E \setminus E'$ that maximizes $(\min\{n\delta - f(W_k(G[E - E'])), f(e, G[E \setminus E'])\})/c(e)$.
  - $E' \leftarrow E' \cup \{e\}$

Lemma 20. *Let E' denote the set of edges found by Algorithm* GREEDYWALK-NONUNIFORM*. Given any constant $\epsilon > 0$, for $k = \log n/\epsilon$, we have $\rho(B(G[E \setminus E'])) \leq (1 + \epsilon)\delta$ and $|c(E')| = O(|E_{\text{OPT}}| \log n \log \Delta)$.*

Proof. The bound on $\rho(B(G[E \setminus E']))$ follows on the same lines as the proof of Lemma 15. The main difference is that the proof of [113] does not consider the case of weights associated with elements. But, as we argue now, the same approach for analyzing greedy algorithms extends to our case, and we show $c(E') = O(c(E_{\text{HITOPT}}) \log n)$.

We partition the iterations of Algorithm GREEDYWALK-NONUNIFORM into $O(\log n)$ phases. Each phase, ends at the first iteration when the total weight that needs to be further covered goes down by a factor of at least 2. So if F is the weight that needs to be covered at the start of the phase, in every iteration of the phase, there exists an edge e (which is in an optimum solution) such that $f(e, G[E\setminus E'])/c(e) \geq F/(2c(E_{\text{HITOPT}}))$. Thus, the total cost of the edges selected in the phase is at most $2c(E_{\text{HITOPT}})$. Since the ratio of $n\delta$ over the minimum weight of a walk is polynomial in n, the total number of phases is $O(\log n)$. Adding over all phases then yields the desired bound on $c(E')$. Putting this together with the rest of the proof of Lemma 15 yields the desired bound. □

6.5.2 Node version: SRMN problem

Recall the definition of walks(v, G, k) from Section 10.1. Let $G[V'']$ denote the subgraph of $G = (V, E)$ induced by subset $V'' \subset V$. We modify Algorithm 1 to work for the SRMN problem in the following manner:

```
1. Initialize $V' \leftarrow \phi$
2. while $W_k(G[V \setminus V']) \geq nT^k$:
    (a) $r \leftarrow W_k(G[E \setminus E']) - nT^k$
    (b) Pick $v \in V \setminus V'$ that maximizes $\frac{\min\{r, \text{walks}(v, G[V \setminus V'], k)\}}{c(v)}$
    (c) $V' \leftarrow V' \cup \{v\}$
```

It can be shown on the same lines as Lemma 15 that this gives a solution of cost $O(c(E_{\text{OPT}}(T)) \log n \log \Delta)$, where $c(E_{\text{OPT}}(T))$ denotes the cost of the optimal solution to SRMN problem. Further, the same running time bounds as in Sections 6.2.1 and 6.2.2 hold.

6.6 Popular heuristics and lower bounds

A number of heuristics have been developed for controlling the spread of epidemics– these are discussed below. All these heuristics involve ordering the edges based on some kind of score, and then selecting the top k based on this score. We describe the score function in each heuristic.

1. ProductDegree ([79]): The score for edge $e = (u, v)$ is defined as $\deg(u) \times \deg(v)$. Edges are removed in non-increasing order of this score.

2. EigenScore ([79, 115]): Let $\mathbf{x}$ be the eigenvector corresponding to the first eigenvalue of the graph. The score for edge $e = (u, v)$ is defined as $|x(u) \times x(v)|$.

3. LinePagerank: This method uses the linegraph $L(G) = (E, F)$ of graph $G = (V, E)$, where $(e, e') \in F$ if $e, e' \in E$ have a common endpoint. We define the score of edge $e \in E$ as the pagerank of the corresponding node in $L(G)$.

As we find in Section 6.7.2, these heuristics work well for different kinds of networks. We design another heuristic, Hybrid, which picks the best of the EigenScore and ProductDegree methods. The edges are ordered in the following manner:

- Let $\pi_1, \ldots, \pi_m$ and $\mu_1, \ldots, \mu_m$ be orderings of edges in the Eigenscore and ProductDegree algorithms, respectively.

- Initialize $i = 0$ and $j = 0$.
- From the edges $\pi(i)$ and $\rho(j)$, remove the one which decreases the max eigenvalue of the residual graph more. Increment the corresponding index.

We now examine the worst case performance of these heuristics. Two of these, namely, EigenScore and ProductDegree, have been used specifically for reducing the spectral radius, e.g., [79, 115]. No formal analysis is known for any of these heuristics in the context of the SRME or SRMN problems; some of them seem to work pretty well on real world networks. We show that the worst case performance of these heuristics can be quite poor, in general.

Theorem 4. *Given any sufficiently large positive integer n, there exists a threshold $T < c\sqrt{n}$, for some constant $c < 1$ and a graph of size n for which the number of edges removed by* ProductDegree*,* EigenScore*,* Hybrid *and* LinePagerank *is $\Omega\big(\frac{n}{T^2}\big)$ OPT.*

Proof. **Construction:** We construct a graph G for which the statement holds. For convenience let us assume that T' is a positive integer. G contains (1) a clique G_1 on $T' + 1$ nodes; (2) a caterpillar tree G_2, which comprises of a path $v_1 v_2 \cdots v_{q-1}$ with v_i adjacent to T' leaves each and (3) G_3, a star graph with $(T' + 1)^2$ leaves and central vertex denoted by v_q. We connect G_1 to G_2 by (v_0, v_1) where, v_0 is some node in G_1 and G_2 is connected to G_3 by the edge (v_q, v_{q-1}). Note that $q = \frac{n-(T'+1)^2-T'}{T'}$ and $\lambda_1(G) \geq \lambda_1(G_3) = T' + 1$. Again, here we assume that q is an integer.

Bound on $c(E_{\text{OPT}})$: We will show that $c(E_{\text{OPT}}) \leq 2T' + 3$. Removing the edges (v_0, v_1) and (v_{q-1}, v_q) isolates the components G_1, G_2 and G_3. G_1 is a clique on $T' + 1$ nodes and on removing one edge, its spectral radius decreases below T'. G_2 is a star with $(T' + 1)^2$ leaves and therefore, on removing at most $(T' + 1)^2 - (T'^2 + 1)$ edges, its spectral radius decreases below T'. It can be shown that $\lambda_1(G_2) \leq \sqrt{T'} + 2$.

Now we will demonstrate that all the four algorithms score the edges (v_i, v_{i+1}), $i = 0, \ldots, q-2$ above any edge belonging to the clique G_1. However, the spectral radius cannot be brought down below T' until at least one edge in G_1 is removed. Therefore, at least q edges will be removed by all the algorithms. By the initial assumption that $T' < c\sqrt{n}$, it follows that $q = \Omega\big(\frac{n}{T'}\big)$, while, by $c(E_{\text{OPT}}) = O(T')$, hence completing the proof. Now we analyze each algorithm separately.

ProductDegree: For all $u \in V(G_1)$, $d(u) \leq T'+1$ while, for each $i = 1, \ldots, q$, $d(v_i) \geq T'+2$. Therefore, (v_i, v_{i+1}), $i = 0, \ldots, q-2$ has higher score than any edge in G_1.

EigenScore: Let x denote the unit eigenvector corresponding to $\lambda_1(G)$ and for any $v \in V(G)$, let $x(v)$ denote the vth component of x. We will show that $x(v_{q-1}) > x(v_{q-2}) > \cdots > x(v_0) > x(v')$ where v' is any vertex in G_1 other than v_0. This implies that all the edges (v_i, v_{i+1}), $i = 0, \ldots, q-2$ have eigenscore greater than the edges in G_1.

Let $\lambda := \lambda_1(G)$. By symmetry, all $v' \in V(G_1) \setminus \{v_0\}$ have the same eigenvector component $x(v')$ and all leaves of v_i have the same component $x(l_i)$. Let A be the adjacency matrix of G. Since $Ax = \lambda x$, we have

$$\begin{aligned}
\lambda x(v') &= (T'-1)x(v') + x(v_0) && \text{(6.2a)}\\
\lambda x(v_0) &= T'x(v') + x(v_1) && \text{(6.2b)}\\
\lambda x(v_i) &= x(v_{i-1}) + x(v_{i+1}) + T'x(l_i),\ 1 \le i \le q-1 && \text{(6.2c)}\\
\lambda x(v_q) &= x(v_{q-1}) + (T'+1)^2 x(l_q) && \text{(6.2d)}\\
\lambda x(l_i) &= x(v_i),\ 1 \le i \le q\,. && \text{(6.2e)}
\end{aligned}$$

From (6.2a) and the fact that $\lambda \ge T'+1$,

$$x(v_0) = (\lambda - T' + 1)x(v') \ge 2x(v')\,. \tag{6.3}$$

By induction on i, we will show that $x(v_i) \ge \frac{T'}{2}x(v_{i-1})$ for $i = 1, \ldots, q-1$. The base case is $i = 1$. Using (6.2b), (6.3) and the bound $\lambda \ge T'+1$,

$$x(v_1) = \lambda x(v_0) - T'x(v') \ge > \frac{T'}{2}x(v_0)\,. \tag{6.4}$$

Assuming $x(v_i) \ge \frac{T'}{2}x(v_{i-1})$ and applying (6.2c), (6.2e) and again $\lambda \ge T'+1$,

$$\begin{aligned}
x(v_{i+1}) &= \lambda x(v_i) - x(v_{i-1}) - T'x(l_i) && \text{(6.5)}\\
&\ge \left(T' + 1 - \frac{2}{T'} - \frac{T'}{\lambda}\right)x(v_i) && \text{(6.6)}\\
&\ge \left(T' + 1 - \frac{2}{T'} - \frac{T'}{T'+1}\right)x(v_i) > \frac{T'}{2}x(v_i). && \text{(6.7)}
\end{aligned}$$

From (6.3) and (6.5), it follows that $x(v_{q-1}) > x(v_{q-2}) > \cdots > x(v_0) > x(v')$.

Hybrid: Since both ProductDegree and EigenScore rate edges (v_i, v_{i+1}), $i = 0, \ldots, q-2$, higher than any edge in G_1, it follows that the same holds for Hybrid as well.

LinePagerank: Let $\pi(e)$ denote the pagerank of edge e. We will show that $\pi(v_{q-1}v_q) = \pi(v_{q-2}v_{q-1}) = \cdots = \pi(v_1v_2) > \pi(v_0v_1) > \pi(e^c_{v_0}) > \pi(e^c)$ where $\pi(e^c_{v_0})$ (by symmetry) is the pagerank of every edge in clique G_1 incident with v_0 while $\pi(e^c)$ (again by symmetry) is the pagerank of every other edge in the clique. Let l_i denote the leaf edges incident with v_i for $i = 1, \ldots, q$. Pagerank of each edge is computed as follows: $\pi(e) = \sum_{e' \in N(e)} \frac{\pi(e')}{d(e')}$ where, $N(e)$ and $d(e)$ denote the set of neighbors and degree respectively of e in the line graph.

In the line graph, the degrees of each edge of G are as follows: $d(e^c) = 2(T'-1)$; $d(e^c_{v_0}) = 2T'-1; d(v_0v_1) = 2T'+1$; $d(v_{q-1}v_q) = (T'+1)^2+1; d(v_iv_{i+1}) = 2(T'+1)$, $i = 1, \ldots, q-$

$2; d(l_i) = T' + 1$, $i = 1, \ldots, q-1$. The pageranks of the relevant edges are as follows:

$$\pi(e^c) = \frac{2(T'-1)-2}{2(T'-1)}\pi(e^c) + \frac{2\pi(e^c_{v_0})}{2(T'-1)+1} \tag{6.8a}$$

$$\pi(e^c_{v_0}) = \frac{\pi(e^c)}{2} + \frac{T'-1}{2T'-1}\pi(e^c_{v_0}) + \frac{\pi(v_0v_1)}{2T'+1} \tag{6.8b}$$

$$\pi(v_0v_1) = \frac{T'\pi(e^c_{v_0})}{2T'-1} + \frac{\pi(v_1v_2)}{2(T'+1)} + \frac{T'\pi(l_1)}{T'+1} \tag{6.8c}$$

$$\pi(v_1v_2) = \frac{\pi(v_0v_1)}{2T'+1} + \frac{\pi(v_2v_3)}{2(T'+1)} + \frac{T'(\pi(l_1)+\pi(l_2))}{T'+1} \tag{6.8d}$$

$$\pi(v_iv_{i+1}) = \frac{\pi(v_{i-1}v_i)+\pi(v_{i+1}v_{i+2})}{2(T'+1)} + \frac{T'(\pi(l_i)+\pi(l_{i+1}))}{T'+1}, \quad i = 2, \ldots, q-2 \tag{6.8e}$$

$$\pi(l_1) = \frac{T'-1}{T'+1}\pi(l_1) + \frac{\pi(v_0v_1)}{2T'+1} + \frac{\pi(v_1v_2)}{2(T'+1)} \tag{6.8f}$$

$$\pi(l_i) = \frac{T'-1}{T'+1}\pi(l_1) + \frac{\pi(v_{i-1}v_i)+\pi(v_iv_{i+1})}{2(T'+1)} \quad i = 2, \ldots, q-2\,. \tag{6.8g}$$

Using (6.8), we have the following:

$$(6.8a) \Rightarrow \pi(e^c_{v_0}) = \frac{2(T'-1)+1}{2(T'-1)}\pi(e^c), \tag{6.9a}$$

$$(6.8b) \text{ and } (6.9a) \Rightarrow \pi(v_0v_1) = \frac{2T'+1}{2T'-1}\pi(e^c_{v_0}), \tag{6.9b}$$

$$(6.8c), (6.8f) \text{ and } (6.9b) \Rightarrow \pi(v_1v_2) = \frac{2(T'+1)}{2T'+1}\pi(v_0v_1), \tag{6.9c}$$

$$(6.8d), (6.8f), (6.8g) \text{ and } (6.9c) \Rightarrow \pi(v_2v_3) = \pi(v_1v_2)\,. \tag{6.9d}$$

Now, by induction on i we can show that $\pi(v_iv_{i+1}) = \pi(v_{i-1}v_i)$, for $i = 2, \ldots, q-2$. The base case $i = 1$ is covered in (6.9d). For any $k \geq 2$, applying $\pi(v_kv_{k+1}) = \pi(v_{k-1}v_k)$ in (6.8d) (with $i = k$) and (6.8g), it follows that $\pi(v_{k+1}v_{k+2}) = \pi(v_kv_{k-1})$.

Hence, proved.

□

6.7 Experiments

6.7.1 Methods and Dataset

We evaluate the algorithms developed in the chapter[1] – GreedyWalk, GreedyWalkSparse and Primal-Dual – and compare their performance with the heuristics from literature (described in Section 6.6) EigenScore, ProductDegree, LinePagerank and Hybrid (described in Section 6.6), as a more sophisticated baseline. The networks which we considered in our empirical analysis are listed in Table 6.1 spanning infrastructure networks, social networks and random graphs.

Table 6.1: Networks and their sizes

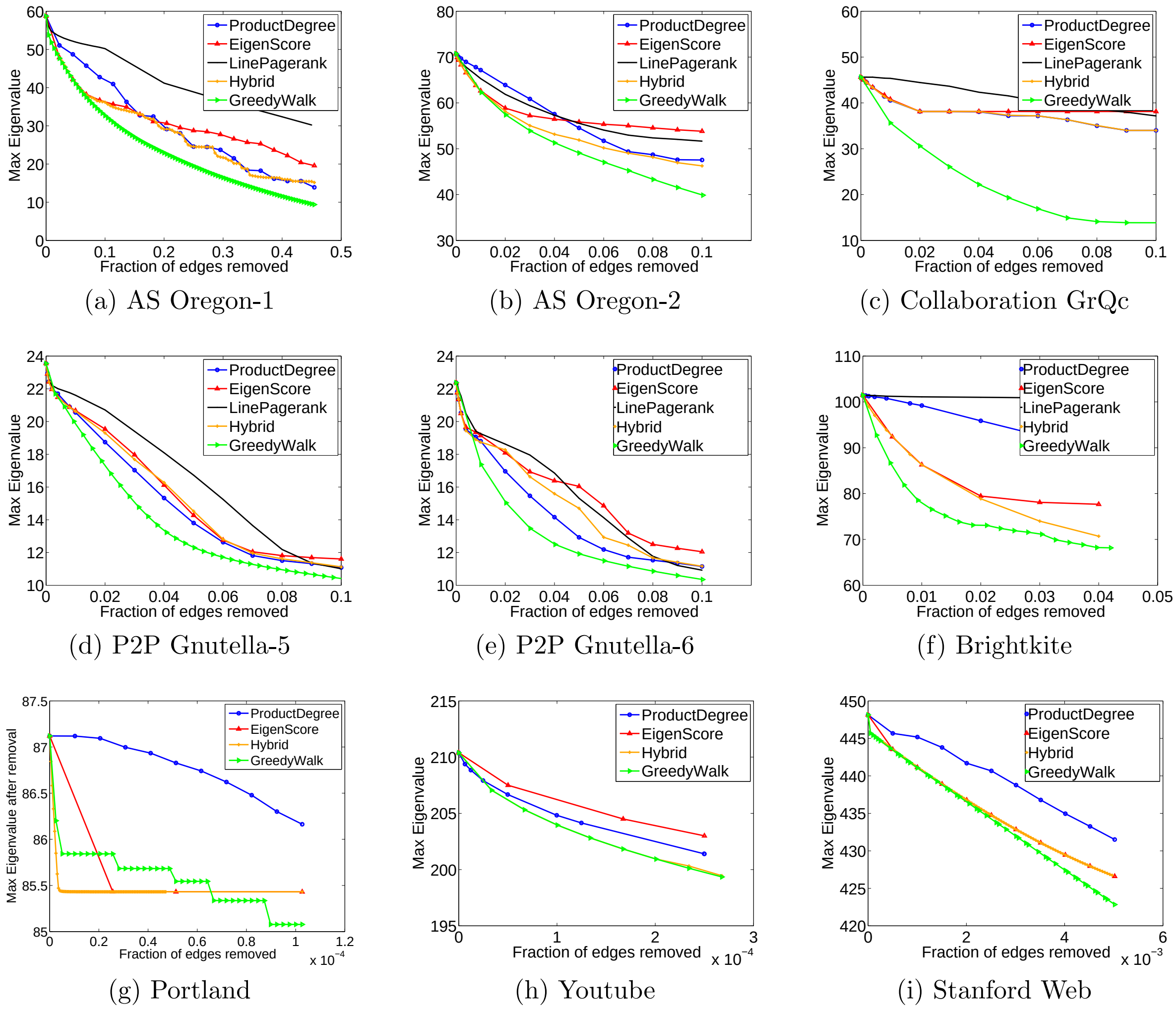
Network	nodes	edges	λ_1	Source
Barabasi-Albert Graph	1000	1996	11.1	Synthetic
Erdos-Renyi Graph	994	2526	6.38	Synthetic
P2P (Gnutella05)	8846	31839	23.55	[2]
P2P (Gnutella06)	8717	31525	22.38	[2]
Collab. Net (HepTh)	9877	25998	31.03	[2]
Collab. Net (GrQc)	5242	14496	45.62	[2]
AS (Oregon 1)	10670	22002	58.72	[2]
AS (Oregon 2)	10900	31180	70.74	[2]
Brightkite Net	58228	214078	101.49	[2]
Portland Cont. Net.	1575861	19481626	87.12	[2]
Youtube Network	1134890	2987624	210.4	[2]
Stanford Web graph	281903	1992636	448.13	[2]

6.7.2 Experimental results

Performance of our algorithms and comparison with other heuristics We first compare the quality of solution from our algorithms with the EigenScore, ProductDegree, LinePagerank and Hybrid heuristics in Figure 6.1. We note that GreedyWalk is consistently better than all other heuristics, especially as the target threshold becomes smaller. Compared to the EigenScore, ProductDegree and LinePagerank heuristics, the spectral radius for the solution produced by GreedyWalk, as a function of the fraction of edges removed, is lower by at least 10-20%; for the collaboration network (Figure 6.1c), the solution produced by GreedyWalk is smaller by more than a factor of two. Our improved baseline, the Hybrid heuristic, works better than the other heuristics, and comes somewhat close the GreedyWalk in many networks. Although, for the collaboration network, Hybrid remains significantly worse than GreedyWalk.

[1] All code at: https://www.dropbox.com/s/j6e5yj0fzyv7vuk/code.zip.

Though Primal-Dual gives a significantly better approximation guarantee, compared to GreedyWalk, it has a much higher running time. Therefore, we only evaluate it for one iteration of Algorithm 6. Figure 6.2 shows that Primal-Dual is quite close to GreedyWalk after just one iteration; we expect running this algorithm fully would further improve the performance, but additional work is needed to improve the running time.



(a) AS Oregon-1

(b) AS Oregon-2

(c) Collaboration GrQc

(d) P2P Gnutella-5

(e) P2P Gnutella-6

(f) Brightkite

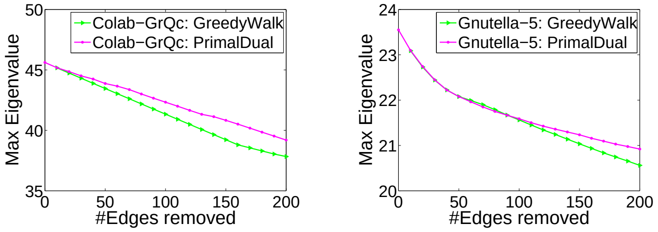
(g) Portland

(h) Youtube

(i) Stanford Web

Figure 6.1: Comparison between the GreedyWalk, ProductDegree, Eigenscore, LinePagerank and Hybrid algorithms for different networks. Each plot shows the spectral radius (y-axis) as a function of the fraction of edges removed (x-axis). The LinePagerank heuristic has not been evaluated in 6.1g, 6.1h and 6.1i because of the scale of these networks.

Running time and effect of sparsification. Figure 6.3 shows the total running time of GreedyWalk for three networks. The time reduces with the value of T, because the while loop in Algorithm 1 needs to be run for fewer iterations. The high running time motivates faster methods. We evaluate the performance of the GreedyWalkSparse algorithm. As



(a) Collaboration GrQc (b) P2P Gnutella-5

Figure 6.2: GREEDYWALK vs PRIMAL-DUAL. Each plot shows the spectral radius (y-axis) as a function of the number of edges removed (x-axis) using the two methods.

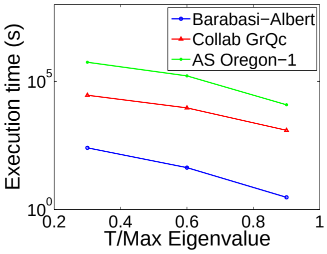
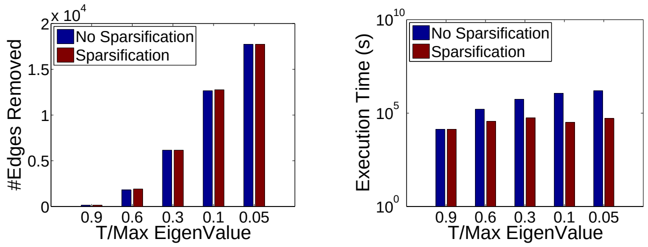


Figure 6.3: Total running time of GreedyWalk method (y-axis) as a function of $T/\rho(G)$ (x-axis), where T is the threshold and $\rho(G)$ is the spectral radius of the initial graph, without any edges removed.

shown in Figure 6.4, GREEDYWALKSPARSE gives almost the same quality of approximation as GREEDYWALK, but improves the running time by up to an order of magnitude, particularly when T is small.

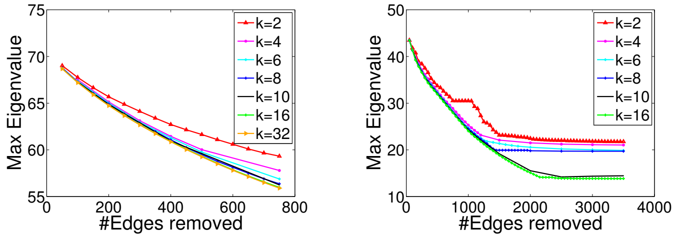
Effect of varying walk lengths As discussed in Section 6.2, the walk length parameter k is critical for the performance of the performance of GREEDYWALK. Figure 6.5 shows the approximation quality in the Oregon-2 and collaboration networks. We find that as k becomes smaller, the approximation quality degrades significantly, and the best performance occurs at k close to $2 \log n$.

Extensions. For the SRME-NONUNIFORM problem, we compare the adaptation of GREEDYWALK, as discussed in Section 6.5, with the EIGENSCORE heuristic run on the matrix B of transmission rates. As shown in Figure 6.6, we find that GREEDYWALK performs much better. Next we consider the SRMN problem, and compare the GREEDYWALK, as adapted in Section 6.5, with the node versions of the DEGREE and EIGENSCORE heuristic [115]. As shown in Figure 6.7, GREEDYWALK performs consistently better.



(a) Edges removed (b) Execution time

Figure 6.4: Impact of sparsification on GreedyWalk. The plots show (a) the number of edges removed and (b) the execution time on the y-axis, as a function of $T/\rho(G)$ (x-axis), where T is the threshold and $\rho(G)$ is the spectral radius of the initial graph, without any edges removed.

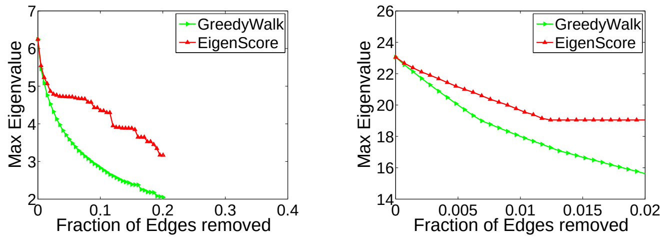


(a) AS Oregon-2 (b) Collaboration GrQc

Figure 6.5: Impact of the value of k on the performance of GreedyWalk. Each plot shows the spectral radius after edge removal (y-axis) vs the number of edges removed (x-axis), for different values of k, ranging from 2 to $2\log n$, for the corresponding networks.

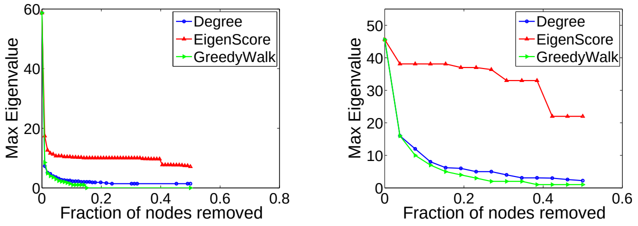
6.8 Conclusions

We study the problem of reducing the spectral radius of a graph to control the spread of epidemics by removing edges (the SRME problem) or nodes (the SRMN problem). We have developed a suite of algorithms for these problems, which give the first rigorous bounds for these problems. Our main algorithm GreedyWalk performs consistently better than all other heuristics for these problems, in all networks we studied. We also develop variants that improve the running time by sparsification, and improve the approximation guarantee using a primal dual approach. These algorithms exploit the connection between the graph spectrum and closed walks in the graph, and perform better than all other heuristics. Improving the running time of these algorithms is a direction for further research. We expect these techniques could potentially help in optimizing other objectives related to spectral properties, e.g., *robustness* [20], and in other problems related to the design of interventions to control the spread of epidemics.



(a) Barabasi-Albert (b) Collaboration GrQc

Figure 6.6: GreedyWalk vs EigenScore for the SRME-nonuniform problem. Each plot shows the spectral radius after edge removal (y-axis) vs the fraction of edges removed (x-axis). The GreedyWalk method was adapted to this setting as in Section 6.5, and the EigenScore method was run on the matrix B of transmission rates.



(a) Collaboration GrQc (b) AS Oregon-1

Figure 6.7: Comparison between GreedyWalk, Degree and EigenScore for the SRMN problem. Each plot shows the spectral radius after node removal (y-axis) vs the fraction of nodes removed.

Chapter 7

Comparison among Different Epidemic Thresholds

In this chapter we will compare spectral radius, λ_1 with other epidemic parameters and investigate their effectiveness as epidemic thresholds. The most popular epidemic parameter is the basic reproduction number which is established as an epidemic threshold in homogeneous mixing models. We will also provide comparisons with the "critical transmissibility based parameter" proposed by Newman [84].

7.1 Epidemic Parameters

7.1.1 Basic reproduction number, R_0

The basic reproduction number, R_0 is defined by Anderson and May [6] as "the average number of secondary infections produced when one infected individual is introduced into a host population where everyone is susceptible." In homogeneous mixing models (or, in complete graphs) it is established as the epidemic threshold parameter. However, in network models the parameter does not exactly correspond to an epidemic threshold parameter, as discussed by Breban et al. [17] and Jing et al. [73]. Estimating the value of R_0 in real outbreaks is not straightforward. We use the approach discussed by Breban et al. [17]. According to this approach, the average number of secondary infections are counted over the first few days of infections. Figure 7.1 shows that the estimated R_0 in this approach converges in a few days.

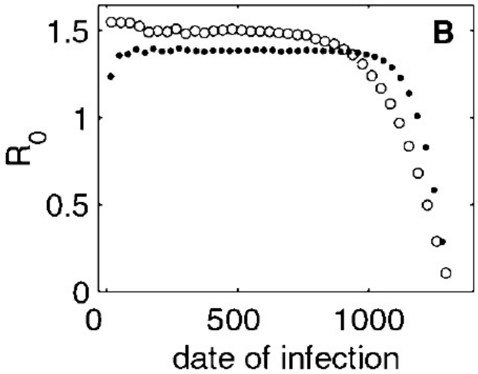


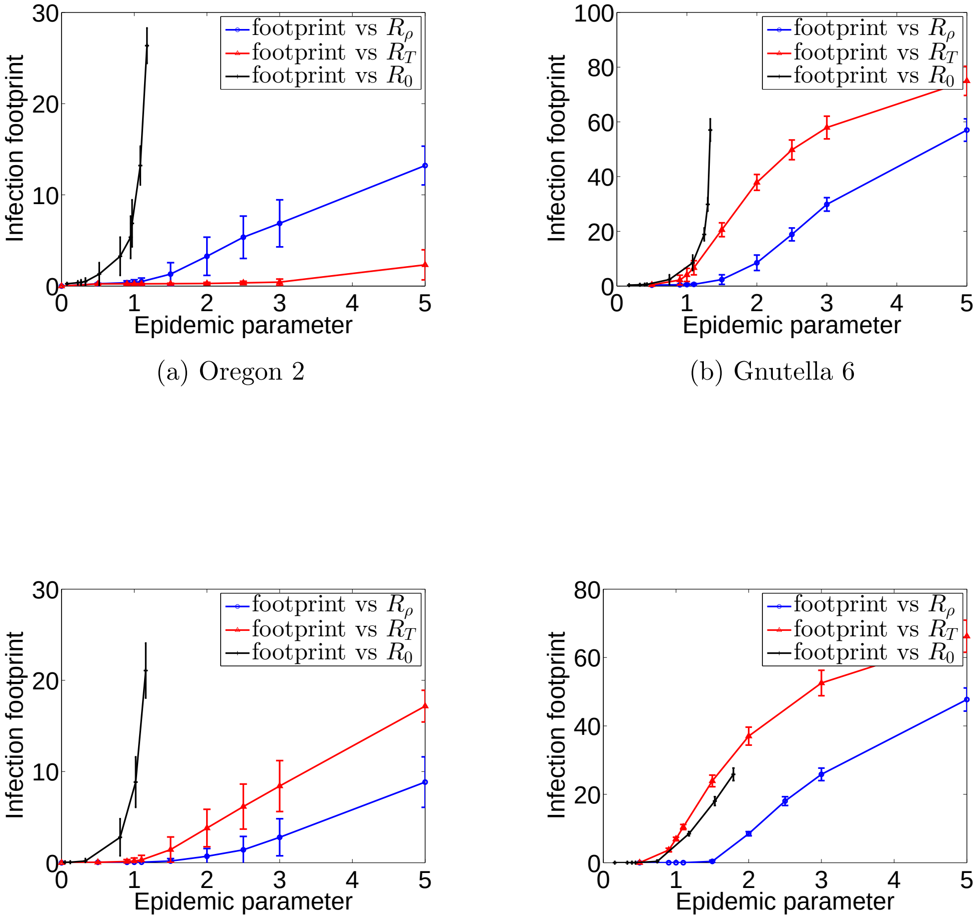
Figure 7.1: The estimated value of R_0 converges in a few days in the approach of Breban et al. [17].

7.1.2 Spectral radius based epidemic parameter, R_ρ

As discussed in earlier chapters, Ganesh et al. [40] and Prakash et al. [95] have shown a characterization of the SIS/SIR models in terms of the spectral radius of the adjacency matrix of the network. Therefore, this can also act as an epidemic summary parameter similar to the basic reproduction number. Ganesh et al. [40] have shown that an epidemic will die out soon if $\rho \leq \frac{\gamma}{\beta}$. However, the opposite has not been established analytically. We consider the spectral radius based epidemic parameter $R_\rho = \rho\frac{\beta}{\gamma}$ and compare its effectiveness with R_0 as an epidemic threshold parameter.

7.1.3 Critical transmissibility based epidemic parameter, R_T

Another epidemic parameter has been discussed for network models by Newman [85]. This is based on a quantity called "critical transmissibility" per edge. Newman demonstrated that the critical transmissibility per edge that results in epidemic outbreak in a configuration type random network (which has no clustering and degree correlation) is $T_c = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}$. Here $\langle k \rangle$ and $\langle k^2 \rangle$ are the average and second moment of node degree distribution. We compare with R_0 and R_ρ the effectiveness of critical transmissibility based epidemic parameter $R_T = \frac{\beta}{\gamma}\left(\frac{\langle k^2 \rangle}{\langle k \rangle} - 1\right)$, i.e., we experiment whether there is an epidemic outbreak depending on $R_T > 1$ or not.



(a) Oregon 2

(b) Gnutella 6

(c) Brightkite

(d) Portland

Figure 7.2: Comparison of the epidemic parameters R_0, R_ρ and R_T as epidemic threshold in different networks - AS Oregon 2, P2P Gnutella 6, Brightkite and Portland social network. The epidemic parameters are inconsistent with each other.

7.2 Comparison among the epidemic parameters

We experimentally investigate how effectively these epidemic parameters work as epidemic thresholds in the SIR model. We use a fast epidemic tool called “Epifast” for epidemic evolution on a contact network. Given a contact network, the initial conditions of an outbreak and its characteristics such as infection probability and recovery probability, EpiFast rapidly computes realizations of stochastic propagation process according to the SIR model. For different networks and for SIR model, we plot the average epidemic footprint for variation in the epidemic parameters. The epidemic footprint refers to the total fraction of nodes that are infected in the network. For a given network, we vary the epidemic parameters by varying the infection probability β but keeping the recovery probability γ fixed. We consider an epidemic parameter as a good epidemic threshold if the following holds: the footprint is very small if the epidemic parameter is less than 1, and it is large if the epidemic parameter is more than or equal to 1.

Figure 7.2 shows experimental comparison among the epidemic parameters in their effectiveness as epidemic thresholds. The figure shows that the parameters are inconsistent with each other. As figure 7.2 shows, R_ρ performs better than the other two parameters in Oregon 2 and Gnutella 6 networks; but as the figure shows for Brightkite and Portland networks, any noticeable increase in the footprint does not happen before R_ρ exceeds 1.5. The basic reproduction number, R_0 is not an epidemic threshold either, since the outbreak does not necessarily dies out when $R_0 < 1$, as can been seen in all four networks. Similarly, R_T works well in the case of Brightkite and Gnutella 6 network, but not in the case of the other two networks. Therefore, the epidemic parameters are inconsistent with each other and their performance depends on the network.

7.3 Conclusions

We find that the epidemic footprint is small for $\rho < \frac{\gamma}{\alpha}$ which corroborates the spectral radius characterization of Ganesh et al. [40]; but for $\rho > \frac{\gamma}{\alpha}$, the footprint is not necessarily large. Our experimental study shows that the epidemic parameters are not consistent with each other. Their performance as epidemic threshold depends on the network. Studying the relation among the epidemic parameters and characterizing their performance remain compelling open problems.

Part II

Containing Cascading Failures in Attack Graphs

Chapter 8

Introduction

In this part, we consider the cascading scenarios which can be modeled by attack graph models. In the network security literature, prior works in [88], [109], [93], [50], [5], [99], [30] have introduced the concept of attack graphs, in which nodes represent some kind of vulnerability configurations (e.g., open port in a machine, unsafe firewall configuration), and edges (directed) represent dependencies among the vulnerabilities. This model captures the cascading dynamics across a range of computer and infrastructure networks, e.g., power transmission networks, smart grids, enterprise networks etc. Here, we address the cascade containment problem using attack graph models.

First, we study the attack cascades and defense strategies in attack graphs with the game framework of FlipIt, proposed by Van Dijk et al. [119]. This game framework allows us to study an important but less studied aspect of cybersecurity - the stealth and persistence of advanced cyber attacks. The existing literature has studied this game in simple scenarios: the game involves either single or multiple but disconnected resources. We propose a generalized FlipIt game for network that we call "FlipNet". In this game model, we study the strategic interaction of a defender and an attacker in stealthily and persistently making moves and controlling networked resources. In chapter 9, we formulate this model, develop analytical results and algorithms, and present experimental results.

Then, we try to understand the inherent vulnerabilities of a network against known exploits and their cascading effects using attack graph models. We analyze the maximum damage that can be mounted by a malicious attacker on a network with given vulnerability configurations and dependencies. We develop approximation algorithms for efficiently hardening the network to control potential cascades. The algorithms and analytic results are presented in chapter 10.

Chapter 9

FlipNet: Modeling Covert and Persistent Attacks on Networked Resources

Persistent and zero-day attacks have increased considerably in the recent past in terms of scale and impact. Security experts can no longer rely only on known defenses and thereby protect their resources permanently. It is now increasingly common to observe attackers capable of repeatedly breaking systems, exploiting new vulnerabilities, and defenders hardening systems with new measures. To model this phenomenon of the repeated takeover of computing resources by system administrators and malicious attackers, a novel game framework, FlipIt, has been proposed by Van Dijk et al. [119] for a system consisting of a single resource.

In this chapter we extend this framework and develop FlipNet, a repeated game framework for a networked system of multiple resources. This game involves two players—a defender and an attacker. Each player's objective is to maximize his gain (i.e., his control over the nodes in the network with stealthy moves), while minimizing the cost for making those moves. This leads to a novel and natural game formulation, with a very complex strategy space, that depends on the network structure. We show that finding the best response strategy for both the defender and attacker is NP-hard. In a key result in this study, we show that the attacker's gain for an instance of the game has a type of diminishing marginal return property, which leads to a near-optimal algorithm for maximizing the attacker's gain. We examine the impact of network structure on the strategy space using simulations.

9.1 Introduction

Defending systems with known defenses, i.e., antivirus signatures, patches etc., is no longer sufficient in the current cyber-security landscape, especially when the targets are high-value systems. This fact is highlighted by several recent cyber attack incidents on systems of national importance. For example, the sophisticated malware StuxNet penetrated Iranian nuclear facilities and destroyed twenty percent of their centrifuges [36]. Another famous incident was the "Aurora" attack, in which the Hydraq trojan compromised important information of Google and other companies [71]. In these cases, the targets of the attacks were of high value and were highly protected. However, the attackers were sophisticated and well funded. These incidents demonstrate that even the tightest security does not guarantee protection in the face of persistent attacks. One common means of launching such attacks is to exploit zero-day vulnerabilities, i.e., the vulnerabilities which are yet to be discovered by the security community. Two important traits of such attacks are their stealthy and persistent nature. *Therefore, it is of practical significance to study what strategies a security expert should take to guard against zero-day and persistent attacks.*

Zero day vulnerabilities in a system are those yet to be identified by the corresponding security vendors. New vulnerabilities are routinely discovered by hackers and hundreds of companies are covertly intruded when these vulnerabilities are exploited [108]. These vulnerabilities generally arise from coding flaws in the software. Once they are discovered in the attacker community, they are exploited in the wilderness for an average of 312 days, as studies have pointed out [13]. While there exists a significant body of work on modeling exploitation of known vulnerabilities [88] [109] [5] [30], there are few that address unknown vulnerabilities. There is thus a clear need to analyze systems and networks in the face of zero-day attacks. Wang et al. [121] have proposed a graph based model of unknown vulnerabilities called the "zero-day attack graph"—while this work proposes a method for computing the security metric of a system, the work does not recommend on the defensive means. Common means such as anti-virus, firewalls etc. do not work against zero-day attacks. However, the defender may take certain steps that can make such attacks more difficult, or can deter them temporarily; examples of such steps include refreshing a virtual machine instance that hosts a service, cleaning machines or rebooting servers, changing passwords or rotating encryption keys, cloud auditing etc [119]. Bowers et al. [16] have discussed how such defensive strategies can be used in system security, particularly password reset policy and cryptographic key rotation. *A natural question to ask for these defensive steps is – how often should such steps be taken?*

One of the most novel approaches for addressing these questions is that of Van Dijk et al. [119], who have proposed a novel game framework called "FlipIt" to model the interaction of an attacker and a defender in repeatedly and stealthily taking over a resource. This game involves a resource which can be controlled by two players - a defender and an attacker. Both the players make security moves to gain control of the resource. For an attacker, one possible move would be to launch an attack, whereas a defender's move could be, for example, changing encryption keys. *However, their moves are often covert, as a player does not know when the other player has moved; this is mostly true in the case of the attackers, whose moves typically remain undetected for a while.* Both the players want to control the resource for as long as possible; however, moves have costs associated. Therefore, a natural problem from both the players' point of view is – how often should they make the moves? Van Dijk et al. [119] have analyzed the dynamics of FlipIt for a single resource, and characterize optimal defender strategies. However, critical resources are often networked. Therefore, analyzing such interactions in a network is an obvious and interesting extension. In this chapter, we extend FlipIt to systems with networked resources. Our contributions are the following.

- We propose FlipNet, a novel game framework on network for modeling the persistent and covert interactions between a defender and an attacker in controlling networked resources. This problem involves many dimensions, including how an attacker can move from one resource to the next, how the move intervals are distributed, and how the move costs are assigned, all of which make it very challenging.

- We study the complexity of finding defender and attacker strategies, and show that maximizing their gains is NP-hard, even under very simplistic conditions. We also find that pure Nash equilibria might not exist, in general, for some of the regimes and models.

- For a model of stochastic moves, we show that the attacker gain has a marginal diminishing return property that allows for a greedy method to compute a $(1 - \frac{1}{e})$ factor approximate solution for the best attacker strategy.

- We use simulations to explore the properties of best response in different kinds of networks. We consider a notion of "critical defender strategies", which make the attacker strategy unsustainable. We find experimentally that critical defender strategies and best defender strategies depend crucially on graph parameters, such as density and "depth". Finally, we study the properties of the most vulnerable nodes, with respect to the attacker's strategy.

9.2 Formal Definitions and Notations

Attack Graph: FlipNet involves two players - a defender (player D) and an attacker (player A) and an attack graph $G = (S \cup V, E)$ where V is the set of vulnerable nodes or security states that can be controlled by either the defender (D) or the attacker (A) (i.e., protected or compromised respectively), S is the set of seed nodes (alternatively called "source" nodes) controlled by the attacker where the attacks start, and E is the set of directed edges that represent security dependencies or atomic attacks. For an edge (u, v), nodes u and v are called the head node and the tail node respectively. Since the graph is

Table 9.1: Notations

D	Defender.
A	Attacker.
G	Attack Graph, $G = (V \cup S, E)$.
S	Set of seed nodes where attack starts.
V	Set of nodes or security states controlled by either D or A.
E	Set of directed edges or security dependencies.
$C_v(t)$	Player (D or A) who controls $v \in V$ at time t.
$C_v^i(t)$	=1 if player $i \in \{D, A\}$ controls $v \in V$ at time t; 0 otherwise.
κ_v^D	Defender's Move cost in unit time at $v \in V$.
κ_e^A	Attacker's Move cost in unit time along $e \in E$.
α_v	Defender move strategy at $v \in V$.
β_e	Attacker move strategy along $e \in E$.
$\Gamma_v^i(t)$	Player i's gain at $v \in V$ up to time t.
$\gamma_v^i(t)$	Player i's average gain rate at $v \in V$ up to time t.
$U^i(t)$	Player i's utility up to time t.
$u^i(t)$	Player i's average utility rate up to time t.

directed in FlipNet, we also call it FlipNetDir. In this work we also consider a variant of the model in which the graph is undirected which represent the scenarios in which attacks can cascade both ways between two nodes; we call this model FlipNetUndir which is discussed in section 9.7.

Moves and Controls: Both the players make repeated security moves at the vulnerable nodes. The defender can choose to move at any node $v \in V$ independently of other nodes. A move of the defender at a node represents a defensive measure that gives the defender control over that node. On the other hand, attacks are propagated through the network by repeated attacker moves according to an independent cascade model. From a node which the attacker controls, he makes repeated moves to its neighboring nodes. Whenever a player makes a move at a node, he gains control over it; if both the players make moves at the same time, the tie is broken in the defender's favor.

The flipping game of control between the two players starts at time $t = 0$ and continues indefinitely as $t \to \infty$. The players can make moves at each discrete time steps $t = 1, 2, ..., \infty$. The state of a node $v \in V \cup S$ at time t is denoted by $C_v(t) \in \{D, A\}$; If the defender controls v at t then $C_v(t) = D$, otherwise $C_v(t) = A$. We also use the binary indicator variable $C_v^i(t) \in \{0, 1\}$ to indicate if player $i \in \{D, A\}$ controls $v \in V$ at time $t \geq 0$. We assume that $C_v(0) = D$ for all $v \in V$ and $C_s(t) = A$ for all $s \in S$ and all $t \geq 0$.

We model the defender and attacker moves with Bernoulli processes; we call this class of move strategy the Bernoulli move strategy. The move strategies of the defender and the attacker are specified by probability vectors, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively. At each of the time steps $t = 1, 2, ..., \infty$, the defender makes a move at $v \in V$ with probability α_v. We assume that the defender can protect any node independently and so we associate the defender strategies with the nodes. On the other hand, the attacker moves are made from a compromised node to a neighboring node by exploiting the corresponding vulnerability dependency; therefore, we associate the attacker move strategies with the edges. For an edge $e = (u, v) \in E$, the attacker moves from u to v along edge $e = (u, v) \in E$ with probability β_e, if the attacker controls the head node u at time $t-1$. Therefore, unlike the defender, the attacker has a constraint on the moves he can make; he has to build on the already made attack moves, which reflects many cyber attack scenarios. Note that, the attacker can move or attack along one direction of an edge as long as it controls the corresponding head node. In FlipNetUndir, however, the attacker can move or attack along both directions of an edge as long as the starting node along the direction is controlled by the attacker.

The move costs, K^D, K^A are specified by cost parameter vectors $\boldsymbol{\kappa^D}$ and $\boldsymbol{\kappa^A}$. Here, κ_v^D denotes the cost of each defender move at node $v \in V$ and κ_e^A denotes the cost of each attacker move along edge $e \in E$. Therefore, the defender's expected cost of moves in unit time at $v \in V$ is given by $\kappa_v^D \alpha_v$; similarly the attacker's expected move cost in unit time along $e \in E$ is $\kappa_e^A \beta_e$. So the expected total move cost of the players up to time t are,

$$K^D(\boldsymbol{\alpha}, t) = \sum_{v \in V} \kappa_v^D \alpha_v t$$

$$K^A(\boldsymbol{\beta}, t) = \sum_{e \in E} \kappa_e^A \beta_e t$$

And, the expected average cost rates are,

$$k^D(\boldsymbol{\alpha}, t) = \frac{K^D(\boldsymbol{\alpha}, t)}{t}$$

$$k^A(\boldsymbol{\beta}, t) = \frac{K^A(\boldsymbol{\beta}, t)}{t}$$

A FlipNet instance is defined by a tuple $\mathcal{F} = (G = (V \cup S, E), \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\kappa^D}, \boldsymbol{\kappa^A})$.

Gains and Utilities: For a given defender and attacker strategy $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and given length of time T, the gain of player i, $\Gamma_v^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)$ denotes the expected amount of time that i controls $v \in V$ up to T , i.e.,

$$\Gamma_v^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \sum_{t=1}^{T} E(C_v^i(t))$$

And, the gain of i in the network is given by,

$$\Gamma^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \sum_{v \in V} \Gamma_v^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)$$

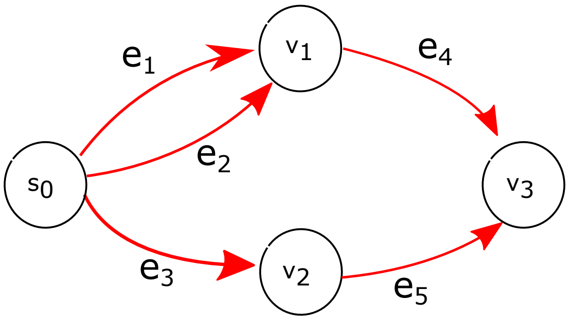


Figure 9.1: Example attack graph, G of four nodes and five edges. The nodes represent security states. Node s_0 is a seed node or attack start node; the other nodes are vulnerable nodes. The edges represent attacks. The defender makes moves at the vulnerable nodes (e.g., changes passwords), whereas the attacker makes moves along the edges (e.g., exploits vulnerabilities).

Thus,

$$\Gamma_v^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) + \Gamma_v^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = T$$

,

$$\Gamma^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) + \Gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = |V|T$$

The average gain rates γ_v^i and γ^i at $v \in V$ and in the whole network respectively are defined as,

$$\gamma_v^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \frac{\Gamma_v^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)}{T}$$
$$\gamma^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \frac{\Gamma^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)}{T}$$

The utility of player i is equal to its gain in the network, minus the expected cost of moves, i.e.,

$$U^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \Gamma^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) - K^D(\boldsymbol{\alpha}, T)$$
$$U^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \Gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) - K^A(\boldsymbol{\beta}, T)$$

And the average utility rates are,

$$u^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \frac{U^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)}{T}$$

$$u^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T) = \frac{U^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)}{T}$$

The objective of player i is to maximize $u^D(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)$ when the game is played up to time T. If the game is played indefinitely, the objective is to maximize the asymptotic utility rate, which is defined as follows,

$$u^i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \lim_{t \to \infty} \inf u^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, t)$$

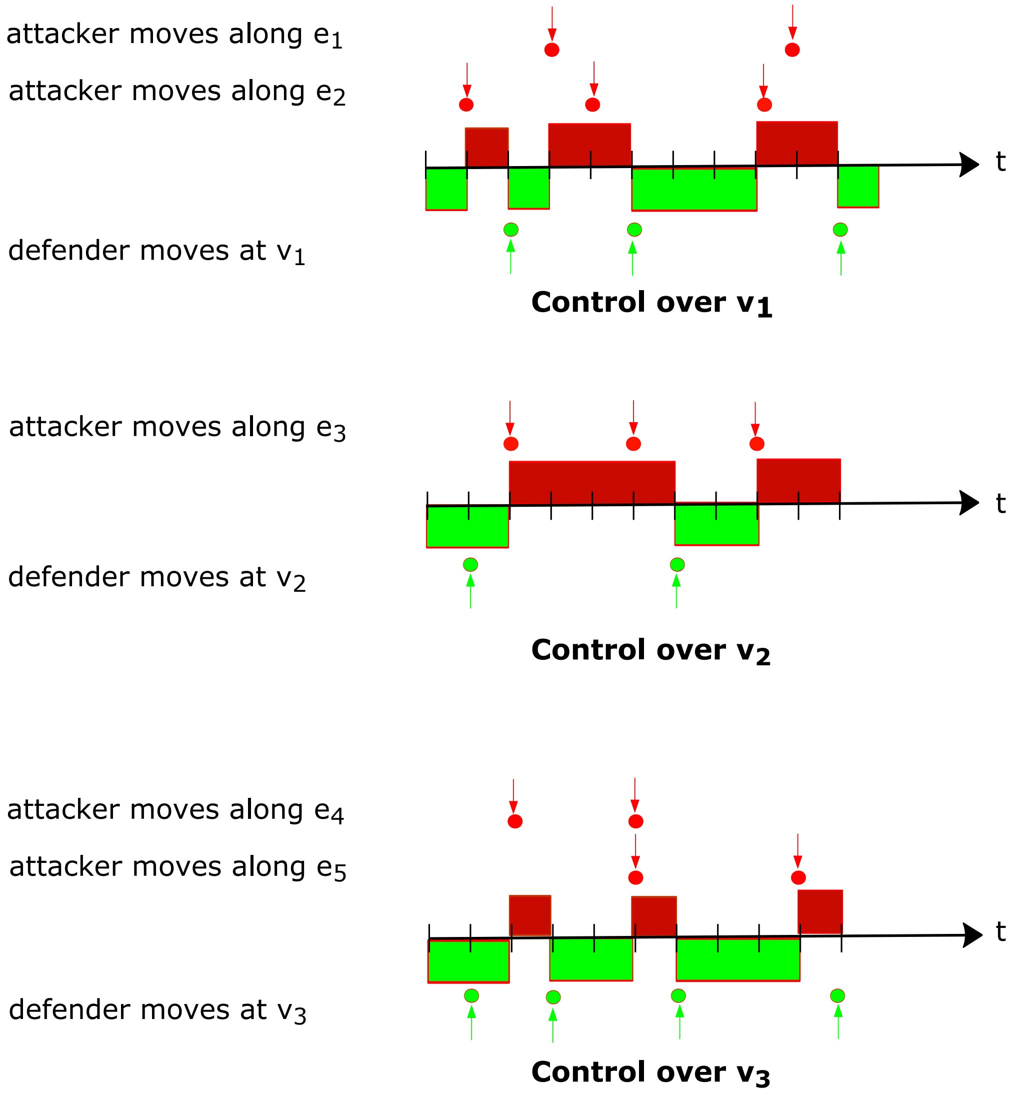

attacker moves along e_1
attacker moves along e_2
t
defender moves at v_1
Control over v_1
attacker moves along e_3
t
defender moves at v_2
Control over v_2
attacker moves along e_4
attacker moves along e_5
t
defender moves at v_3
Control over v_3

Figure 9.2: Example FlipNet illustration on the attack graph of figure 9.1 for Bernoulli move strategies. The figure illustrates the flipping of control over the nodes (i.e., security states) over a period of time. The green arrows represent the defender moves. The attacker's move attempts are shown by red arrows.

When the strategy vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$ can be implied from the context, we omit them in the gain and cost variables. For example, we use $\Gamma^i(t)$ instead of $\Gamma^i(\boldsymbol{\alpha}, \boldsymbol{\beta}, t)$; similarly we use $\Gamma^i_v(t), \gamma^i_v(t), \gamma^i(t), K^i(t), U^i(t), u^i(t)$

9.3 The complexity of finding optimal strategies

The network structure makes analyzing FlipNet very challenging. We can show that computing the optimal strategy of the defender or the attacker is NP-hard even when the time of opponent's moves are known.

Lemma 21. *Given an attacker strategy $\boldsymbol{\beta}$ which is known to the defender, and a budget B on the total move cost, computing a maximum gain strategy $\boldsymbol{\alpha}$ of cost at most B for the defender is NP-hard.*

Proof. The proof is by a reduction from the Knapsack problem. Consider an instance of Knapsack with elements with weights $\{b_1, \ldots, b_n\}$, corresponding values $\{a_1, \ldots, a_n\}$ and a budget B. We construct an instance of FlipNet in the following manner. G is a star graph with source s and directed edges to nodes $v_1, \ldots, v_n$ and the game is played for just one time step, i.e., $T = 1$. We consider an attacker move strategy in which it moves on all the edges (s, v_i) at $t = 1$. The defender move strategy corresponds to a choice of a subset of $\{v_1, \ldots, v_n\}$ to move at time $t = 1$. We set the defender move cost for v_i to be equal to b_i, and the corresponding utility to be a_i per time unit for controlling a node. It is easy to see that, if there is a knapsack solution of value A, then that corresponds to a defender gain of A in FlipNet and vice versa. Since the knapsack problem is NP-hard and this is a polynomial time reduction, therefore the lemma follows. □

Lemma 22. *Given a defender strategy $\boldsymbol{\alpha}$ which is known to the attacker, and a budget B on the total move cost, computing a maximum gain strategy $\boldsymbol{\beta}$ of cost at most B for the attacker is NP-hard.*

Proof. We use a similar reduction as in the proof of Lemma 21. We consider a defender strategy which involves no moves. Therefore, the best attacker strategy involves picking a subset of edges (s, v_i) to move that maximizes its gain. The costs and utilities for the attacker are set in a similar manner. Since, the defender does not move at any node, so for any move the attacker controls the corresponding node. It is easy to see that, a knapsack solution of value A corresponds to the attacker's gain of A and vice versa. Therefore, the lemma follows. □

9.4 Existence of Nash equilibria

The strategy space in the FlipNet game is much more complex than the FlipIt model. We find that, in general, pure Nash equilibria (NE) do not exist.

Lemma 23. *There exist instances of* FlipNet *with deterministic moves, in which no pure NE exists.*

Proof. Consider a case in which either player's cost of making a move is less than 1 (which is equal to his utility per unit time for controlling each node). In that case, no choice of defender and attacker moves is in equilibrium for which the attacker controls at least one node for at least one time unit; because, for such a case, the defender would simply add a move at the corresponding node and at the corresponding time and thereby increase his utility. On the other hand, if the attacker makes no move at all, the defender also does not find any incentive to make any move; if the defender makes no move, the attacker can increase his utility by making moves. Therefore, no pure Nash equilibrium exists in this setting. □

Extending this to the case of stochastic strategies seems very challenging. Following the approach of Laszka et. el. [69], we use simulations for computing best response strategies to study equilibria in these games. We find that no pure NE exists in general for the case of uniform strategies, as will be discussed later.

9.5 Results on optimizing attacker's gain

It is generally difficult to reason about the gain and utility of the players in FlipIt when extended to multiple resources. The network structure in FlipNet makes the task even more difficult. However as it turns out, we can work with an alternate view of the control and gain of the players in FlipNet. We can show that, the sequence of attack moves within the network over time can be represented by a time-expanded graph. From the graph $G = (V \cup S, E)$, the corresponding time-expanded graph $G^T = (V^T \cup S^T, E^T)$ can be constructed as follows for time $t = 0, 1, ..., T$. For each node $v \in V \cup S$, we create $T+1$ copies $v_0, v_1, ..., v_T$ in G^T. And for each edge $e = (u, v) \in E$, we create copies $e_i = (u_{i-1}, v_i) \in E^T$ for $1 \leq i \leq T$. Additionally we create self edges $(v_{i-1}, v_i) \in E^T$ for each node $v \in G$ and $1 \leq i \leq T$. We call edge e_i as the i^{th} copy of e. The set of all copies of $e \in E$ is denoted by E^T_e where $E^T_e \subset E^T$. A simple path graph G and its corresponding time-expanded graph are shown in figure 9.3.

The sequence of attacker and defender moves and control in graph G over time can be represented as activation of edges and nodes in G^T as follows. By default, none of the nodes and edges in G^T are active. Node and edge activations represent defender and attacker moves respectively. If the defender makes a move at node $v \in V$ at time step t, we represent this by the activation of $v_t \in V^T$; in this case, we refer to the node as blocked. If v_t is not activated, then it is live. An attacker move, on the other hand, along edge (u, v) at time time t is represented by the activation of $(u_{t-1}, v_t) \in E^T$; we say the edge is live if it is activated by the attacker. If (u_{t-1}, v_t) is not activated, then it is blocked. The self edges $(v_t, v_{t+1}) \in E^T$ and source nodes $s_t \in S^T$ are always live. A node $v_t \in V^T$ is reachable by the attacker, if there is a path consisting of only live nodes and live edges from a source node $s \in S^T$ to v_i.

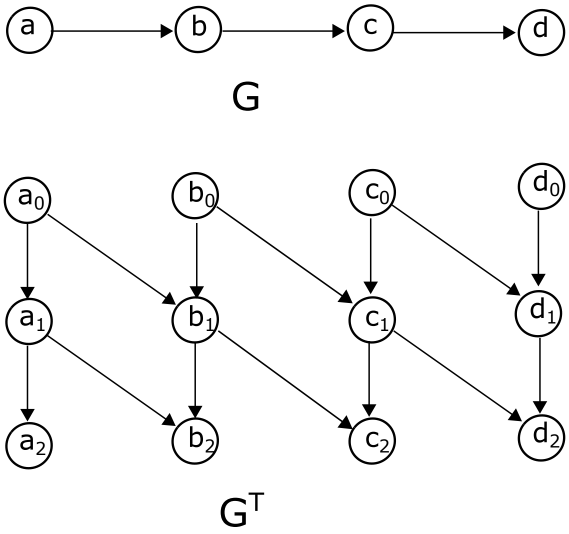


Figure 9.3: Example graph G and its time-expanded graph for $T = 2$ time steps.

Lemma 24. *The gain of the attacker, $\Gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)$ in G is equal to the expected number of reachable nodes in G^T.*

Proof. Let $M^A \subseteq \{\cup_{e \in E} E_e^t\}$ denote a schedule of moves of the attacker up to time t; a schedule of defender moves, M^D is defined similarly. First we show that for a given schedule of defender and attacker moves, the total amount of time that the attacker controls any node $v \in V$ in G is equal to the number of reachable nodes in G^T. In other words, we will show that the number of reachable nodes in G^T is equal to $\sum\limits_{v \in V} \sum\limits_{t \leq T} C_v^T(t)$. Since both the expected control time or gain, and the expected reachable nodes is computed over the probability space of all such schedules, it will follow that, the gain is equal to the expected number of reachable nodes.

We can show that the total amount of time that the attacker controls any node is equal to the total number of reachable nodes in G^T by showing that, for any $v \in V$ and $t \leq T$, the following holds: $C_v^A(t) = 1$ if and only if v_t is reachable in G^T. We can show this by mathematical induction on the time steps.

$C_v^A(t) = 1$ at time $t = 1$, if and only if the attacker makes a move from a source $s \in S$ to v along an edge $(s, v) \in E$ and the defender does not make a move at v at $t = 1$. For attacker's move along (s, v) at $t = 1$, the edge (s_0, v_1) is activated in the corresponding time-expanded graph; Since the defender does not move at v at $t = 1$, the node v_1 remains live. Therefore, there is a path of live nodes and edges from the source to v_1 in G^T. Thus, v_1 is reachable.

Now suppose for time steps up to $t = i$ the following is true, $C_v^A(t) = 1$ if and only if v_t is reachable in G^T. Now, $C_v^A(i + 1) = 1$, if the defender does not move at v at time $t = i + 1$ and either (i) the attacker attempts a move along an incoming edge $(u, v) \in E$ at $t = i + 1$

and $C_u^A(i) = 1$, or, (ii) $C_v^A(i) = 1$. If the defender does not move at v at $t = i + 1$, then v_{i+1} is live. Now if (i) is true, then u_i is reachable (because of the inductive hypothesis) and edge (u_i, v_{i+1}) is activated; therefore, v_{i+1} is reachable. If (ii) is true then, v_i is reachable due to the inductive hypothesis. Since each self-edge is live, so (v_i, v_{i+1}) is live. Therefore, since v_{i+1} is not blocked, v_{i+1} is also reachable.

□

We use the following notations to prove our results in this section. An activation sample $\sigma \in 2^{|X|}$ on the set $X \subseteq \{V^T \cup E^T\}$ denotes a sample set of activated nodes and edges in X (Note that, seed nodes S^T are always activated or live, i.e., in attacker's control). Let $E_e^T \subset E^T$ denote the set of copies of $e \in E$. For $X \subseteq E_e^T$ and for a given fixed activation sample σ of edges $E^T - E_e^T$ and nodes V^T, let $g_e^\sigma(X)$ denote the number of reachable nodes if edges X are activated and $E_e^T - X$ are not activated.

We will show a submodularity property of $g_e^\sigma(X)$. A set function $f : 2^\Omega \to \mathbb{R}$, where Ω is the set of elements and 2^Ω is the power set of Ω, is submodular if the following is true: for any $X \subseteq Y \subset \Omega$ and every $x \in \Omega - Y$, $f(Y \cup x) - f(Y) \leq f(X \cup x) - f(X)$. Therefore, submodular set functions have a diminishing return property.

Lemma 25. *For edge $e = (u, v) \in E$ in a DAG G and for any activation sample σ on $V^T \cup (E^T - E_e^T)$, the function $g_e^\sigma : E_e^T \to \mathbb{N}$ is submodular.*

Proof. Let B denote the set of nodes that become reachable for the activation of the nodes and edges in sample σ and the activation of no edge in E_e^T, i.e., $g_e^\sigma(\{\}) = |B|$. Consider a copy edge $e_i = (u_{i-1}, v_i) \in E_e^T$. If this edge is activated and either u_{i-1} or v_i is blocked, then nothing happens and no new node is reached for this activation. But, if e_i is activated and both u_{i-1} and v_i are live as well, then v_i becomes reachable, which in turn potentially makes other nodes reachable. Given the reachable nodes B, let the set of nodes that become newly reachable due to the activation of e_i be X_i.

Note that X_i does not include any u_{j-1} (or, the head node of any e_j) where $i \neq j$. Firstly, it is obvious for $j < i$, since no edge goes back in time from v_i to u_{j-1}. It is also obvious for $j = i$ due to the construction of the time-expanded graph. If $j > i$, still u_{j-1} cannot be reached by any sequence of edges from v_i, since G is a DAG.

So for a fixed σ, X_i is not affected by the activation state (or the absence of it) of another $e_j \in E_e^T, j \neq i$. In other words, for a given σ, if a node is reachable due to the activation of e_i, then the node remains reachable whether or not any other $e_j \in E_e^T, j \neq i$ is activated. Now consider $Y \subseteq Z \subset E_e^T$ and $e_j \notin Z$. Note that,

$$g_e^\sigma(Y) = |\bigcup_{i:v_i \in Y} X_i \cup B|$$

and

$$g_e^\sigma(Z) = |\bigcup_{i:v_i \in Z} X_i \cup B|.$$

Since $Z \supseteq Y$, therefore,

$$\bigcup_{i:v_i \in Z} X_i \cup B \supseteq \bigcup_{i:v_i \in Y} X_i \cup B.$$

So,

$$X_j - (\bigcup_{i:v_i \in Z} X_i \cup B) \subseteq X_j - (\bigcup_{i:v_i \in Y} X_i \cup B).$$

Note that,

$$|X_j - \bigcup_{i:v_i \in Z} X_i \cup B| = g_e^\sigma(Z \cup \{e_j\}) - g_e^\sigma(Z).$$

and,

$$|X_j - \bigcup_{i:v_i \in Y} X_i \cup B| = g_e^\sigma(Y \cup \{e_j\}) - g_e^\sigma(Y).$$

Hence,

$$g_e^\sigma(Z \cup \{e_j\}) - g_e^\sigma(Z) \leq g_e^\sigma(Y \cup \{e_j\}) - g_e^\sigma(Y).$$

□

For a defender and attacker strategy vector $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively, let $f_{\boldsymbol{\alpha}}(\boldsymbol{\beta})$ denote the gain of the attacker up to time T, i.e., $f_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) = \Gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta}, T)$. Let $\chi_e \in \{0,1\}^{|E|}$ denote the characteristic vector of $e \in E$; i.e., this is a vector of $|E|$ components where each component is 0 except the one that corresponds to edge $e \in E$.

We define the following reachability variables which we will use to describe our results and proofs in this section. For $X \subseteq E_e^T$ and for a fixed sample of node and edge activation on $V^T \cup E^T - X$, let $R_p^\sigma(X)$ denote the expected number of nodes reachable in G^T if edges in X are each activated independently with probability p. Note that, $R_p^\sigma(X)$ is defined under fixed sample activation σ in the rest of the graph except X. So if Z denotes activated edges in $E_e^T - X$, i.e., $Z = \sigma \cap E_e^T$ then, $R_p^\sigma(X) = \sum\limits_{Y \subseteq X} p^{|Y|}(1-p)^{|X-Y|} g_e^\sigma(Y \cup Z)$.

Now we will define two similar variables $R_p^\sigma(X|e_j)$ and $R_p^\sigma(X|\overline{e_j})$. For $X \subsetneq E_e^T$ and $e_j \in E_e^T - X$, these variables are defined for a fixed activation sample σ on $V^T \cup (E^T - X - \{e_j\})$. That is, $\sigma \subseteq V^T \cup (E^T - X - \{e_j\})$ is a fixed sample of activated nodes in V^T and edges in

$E^T - X - \{e_j\}$. Let $R_p^\sigma(X|e_j), e_j \in E^T - X$ denote the expected number of nodes reachable in G^T if edge e_j is activated, edges in X are each activated with probability p, and the rest of the edges in $E^T - X - \{e_j\}$ and nodes in V^T are activated according to fixed sample σ. Similarly $R_p^\sigma(X|\overline{e_j})$ denote the same thing as $R_p^\sigma(X|e_j)$ except e_j is not activated.

Let p be an activation probability and $p' = p + \epsilon$ and $p'' = p + 2\epsilon$ for some constant $\epsilon > 0$.

Lemma 26. *For an edge $e \in E$ and $X \subsetneq E_e^T$ and $e_j \in E_e^T - X$ and for a fixed activation sample σ on $V^T \cup (E^T - X - \{e_j\})$, the following is true,*

$$R_{p'}^\sigma(X|e_j) - R_p^\sigma(X|e_j) \leq R_{p'}^\sigma(X|\overline{e_j}) - R_p^\sigma(X|\overline{e_j}).$$

Proof. Let $B \subset E_e^T - X - \{e_j\}$ denote the set of activated edges in σ, i.e., $B = \sigma \cup (E_e^T - X - \{e_j\})$.

We are considering the difference between the case when e_j is activated versus the case when it is not. First, sample the edges in X with probability p each and let $X_p \in 2^X$ occurs as the set of activated edges with probability $Pr_p(X_p) = p^{|X_p|}(1-p)^{|X - X_p|}$. Note that, the number of reachable nodes due to the activation of $X_p \in 2^X$, when e_j is not activated, is given by $g_e^\sigma(X_p \cup B)$; therefore,

$$R_p(X|\overline{e_j}) = \sum_{X_p \in 2^X} Pr_p(X_p) g_e^\sigma(X_p \cup B).$$

For each such sample X_p, if the unselected edges $X - X_p$ are resampled with probability $\epsilon' = \frac{\epsilon}{1-p} \leq 1$, then some edges $X_p' \in 2^{X - X_p}$ become newly activated with probability $Pr_{\epsilon'}(X_p') = \epsilon'^{|X_p'|}(1-\epsilon')^{|X - X_p - X_p'|}$ which results in $g_e^\sigma(X_p \cup X_p' \cup B)$ nodes reachable. Note that, this process of sampling with p first and resampling the rest with probability $\epsilon' = \frac{\epsilon}{1-p}$ has the same effect of sampling X with probability $p + \epsilon$, because the probability that an edge is selected in this two step process is $p + (1-p)\frac{\epsilon}{(1-p)} = p + \epsilon$. Therefore,

$$R_{p'}(X|\overline{e_j}) = \sum_{X_p \in 2^X} \sum_{X_p' \in 2^{X - X_p}} Pr_p(X_p) Pr_{\epsilon'}(X_{p'}) g_e^\sigma(X_p \cup X_p' \cup B).$$

So, the difference in reachable nodes for sampling with p and $p' = p + \epsilon$, when e_j is not activated, is,

$$\begin{aligned}
&R_{p'}(X|\overline{e_j}) - R_p(X|\overline{e_j}) \\
&= \sum_{X_p \in 2^X} Pr_p(X_p) \left((\sum_{X'_p \in 2^{X-X_p}} Pr_{\epsilon'}(X'_p) g_e^{\sigma}(X_p \cup X'_p \cup B)) - g_e^{\sigma}(X_p \cup B) \right) \\
&= \sum_{X_p \in 2^X} Pr_p(X_p) \left((\sum_{X'_p \in 2^{X-X_p}} Pr_{\epsilon'}(X'_p) g_e^{\sigma}(X_p \cup X'_p \cup B)) - (\sum_{X'_p \in 2^{X-X_p}} Pr_{\epsilon'}(X'_p)) g_e^{\sigma}(X_p \cup B) \right) \\
&= \sum_{X_p \in 2^X} \sum_{X'_p \in 2^{X-X_p}} Pr_p(X_p) Pr_{\epsilon'}(X'_p) \left(g_e^{\sigma}(X_p \cup X'_p \cup B) - g_e^{\sigma}(X_p \cup B) \right). \qquad (9.1)
\end{aligned}$$

The third line above follows since $\sum_{X'_p \in 2^{X-X_p}} Pr_{\epsilon'}(X'_p) = 1$. Similarly, if e_i is included in the activated set, then the difference in reachable nodes is,

$$\begin{aligned}
&R_{p'}(X|e_i) - R_p(X|e_i) \\
&= \sum_{X_p \in 2^X} \sum_{X'_p \in 2^{X-X_p}} Pr_p(X_p) Pr_{\epsilon'}(X'_p) \left(g_e^{\sigma}(X_p \cup X'_p \cup B \cup \{e_i\}) - g_e^{\sigma}(X_p \cup B \cup \{e_i\}) \right). \quad (9.2)
\end{aligned}$$

From the submodularity result in lemma 25, it follows that for any $X_p \in 2^X$ and $X'_p \in 2^{X-X_p}$, the following inequality is true, $g_e^{\sigma}(X_p \cup X'_p \cup B) - g_e^{\sigma}(X_p \cup B) \geq g_e^{\sigma}(X_p \cup X'_p \cup B \cup \{e_i\}) - g_e^{\sigma}(X_p \cup B \cup \{e_i\})$. Therefore, from eq 9.1 and 9.2, for any probability distribution $Pr(X_p)$ and $Pr(X'_p)$, the following is true: $R_{p'}(X|\overline{e_i}) - R_p(X|\overline{e_i}) \geq R_{p'}(X|e_i) - R_p(X|e_i)$.

□

Theorem 5. *For given defender and attacker strategies $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and a constant $\epsilon > 0$ the following is true, $f_{\boldsymbol{\alpha}}(\boldsymbol{\beta} + 2\epsilon\chi_e) - f_{\boldsymbol{\alpha}}(\boldsymbol{\beta} + \epsilon\chi_e) \leq f_{\boldsymbol{\alpha}}(\boldsymbol{\beta} + \epsilon\chi_e) - f_{\boldsymbol{\alpha}}(\boldsymbol{\beta})$.*

Proof. Note that, for an edge $e \in E$, there are corresponding T copies $E_e^T = \{e_1, e_2, ..., e_T\}$ in G^T. So, when β_e increases by ϵ (i.e., $\epsilon\chi_e$ is added to $\boldsymbol{\beta}$), the probability of activation of each $e_i \in E_e^T$ is increased by ϵ. Let $\mathbb{S}_{\boldsymbol{\beta}}$ denote the sample space over $V^T \cup (E^T - E_e^T)$ where each sample $\sigma \in \mathbb{S}_{\boldsymbol{\beta}}$ is a sample of activation of nodes V^T and edges $E^T - E_e^T$; let the probability distribution of this sample space be denoted by $Pr_{\boldsymbol{\beta}}(\sigma)$ for each $\sigma \in \mathbb{S}_{\boldsymbol{\beta}}$ which follows from the strategy $\boldsymbol{\beta}$. Note that, $f_{\boldsymbol{\alpha}}(\boldsymbol{\beta}) = \sum_{\sigma \in \mathbb{S}_{\boldsymbol{\beta}}} Pr_{\boldsymbol{\beta}}(\sigma) R^{\sigma}_{\beta_e}(E_e^T)$. Therefore, we can prove the theorem by proving the following for any sample activation $\sigma \in \mathbb{S}_{\boldsymbol{\beta}}$,

$$R^{\sigma}_{p''}(E_e^T) - R^{\sigma}_{p'}(E_e^T) \leq R^{\sigma}_{p'}(E_e^T) - R^{\sigma}_{p}(E_e^T),$$

where $p = \beta_e, p' = p + \epsilon, p'' = p + 2\epsilon$. In the following, we will show this by mathematical induction for a fixed activation sample σ. Here is a set of activated nodes and edges from

$V^T \cup (E^T - E_e^T)$. We will also use other activation samples σ' and σ'' which are superset of σ and includes some elements of E_e^T. In the proof we will use the notation $\mathbb{H}_i \subseteq E_e^T$ to denote the edge set $\{e_1, e_2, ..., e_i\}$.

Base Case: Let $\mathbb{H}_2 = \{e_1, e_2\}$. And let σ' be an activation sample on $V^T \cup (E^T - \mathbb{E}_2)$ where $\sigma' \supseteq \sigma$. That is, the activation sample σ' includes all the activated nodes and edges in σ, plus zero or more elements from $E_e^T - \mathbb{H}_2$. First we show for any such σ' that,

$$R_{p''}^{\sigma'}(\mathbb{H}_2) - R_{p'}^{\sigma'}(\mathbb{H}_2) \leq R_{p'}^{\sigma'}(\mathbb{H}_2) - R_{p}^{\sigma'}(\mathbb{H}_2)$$

There are four possible scenarios involving the activation of e_1 and e_2. Let B_1, B_2, B_3, B_4 denote the number of reachable nodes under sample σ' when both are activated, only e_1 is activated, only e_2 is activated and neither is activated, respectively. Then the expected number of reachable nodes is,

$$R_p^{\sigma'}(\mathbb{E}_2) = p^2 B_1 + p(1-p)B_2 + (1-p)pB_3 + (1-p)^2 B_4$$

Similarly,

$$R_{p'}^{\sigma'}(\mathbb{H}_2) = (p+\epsilon)^2 B_1 + (p+\epsilon)(1-(p+\epsilon))B_2 + (1-(p+\epsilon))(p+\epsilon)B_3 + (1-(p+\epsilon))^2 B_4$$

$$R_{p''}^{\sigma'}(\mathbb{H}_2) = (p+2\epsilon)^2 B_1 + (p+2\epsilon)(1-(p+2\epsilon))B_2 + (1-(p+2\epsilon))(p+2\epsilon)B_3 + (1-(p+2\epsilon))^2 B_4$$

So,

$$R_{p'}^{\sigma'}(\mathbb{H}_2) - R_{p}^{\sigma'}(\mathbb{H}_2) = (2p\epsilon + \epsilon^2)B_1 + (\epsilon - 2p\epsilon - \epsilon^2)(B_2 + B_3) + (2p\epsilon - 2\epsilon + \epsilon^2)B_4$$

$$R_{p''}^{\sigma'}(\mathbb{H}_2) - R_{p'}^{\sigma'}(\mathbb{H}_2) = (2p\epsilon + 3\epsilon^2)B_1 + (\epsilon - 2p\epsilon - 3\epsilon^2)(B_2 + B_3) + (2p\epsilon - 2\epsilon + 3\epsilon^2)B_4$$

Therefore,

$$\begin{aligned}
&\Big(R_{p''}^{\sigma'}(\mathbb{H}_2) - R_{p'}^{\sigma'}(\mathbb{H}_2)\Big) - \Big(R_{p'}^{\sigma'}(\mathbb{H}_2) - R_{p}^{\sigma'}(\mathbb{H}_2)\Big) \\
=&2\epsilon^2 B_1 - 2\epsilon^2 B_2 - 2\epsilon^2 B_3 + 2\epsilon^2 B_4 \\
=&2\epsilon^2(B_1 + B_4) - 2\epsilon^2(B_2 + B_3) \\
\leq&0
\end{aligned}$$

The last line follows from lemma 25. Therefore the base case is true.

Inductive Case: Now we show that, if

$$R_{p''}^{\sigma'}(\mathbb{H}_i) - R_{p'}^{\sigma'}(\mathbb{H}_i) \leq R_{p'}^{\sigma'}(\mathbb{H}_i) - R_{p}^{\sigma'}(\mathbb{H}_i) \tag{9.3}$$

for $i < T$ and any activation sample σ' on $V^T \cup (E^T - \mathbb{H}_i)$ where $\sigma' \supseteq \sigma$, then,

$$R_{p''}^{\sigma''}(\mathbb{H}_{i+1}) - R_{p'}^{\sigma''}(\mathbb{H}_{i+1}) \leq R_{p'}^{\sigma''}(\mathbb{H}_{i+1}) - R_{p}^{\sigma''}(\mathbb{H}_{i+1}). \tag{9.4}$$

for activation sample σ'' on $V^T \cup (E^T - \mathbb{H}_{i+1})$ where $\sigma'' =\supseteq \sigma$.

Since for any arbitrary fixed sample σ' on $V^T \cup E^T - \mathbb{H}_i$ the relationship 9.3 is true, the following two relationships are also true for any arbitrary fixed sample $\sigma'' \supseteq \sigma$ on $V^T \cup E^T - \mathbb{H}_{i+1}$. Considering a σ' where $e_{i+1} \in \sigma'$, we get from 9.3,

$$R^{\sigma''}_{p''}(\mathbb{H}_i|e_{i+1}) - R^{\sigma''}_{p'}(\mathbb{H}_i|e_{i+1}) \leq R^{\sigma''}_{p'}(\mathbb{H}_i|e_{i+1}) - R^{\sigma''}_{p}(\mathbb{H}_i|e_{i+1}) \tag{9.5}$$

And considering a σ' where $e_{i+1} \notin \sigma'$, we get from 9.3,

$$R^{\sigma''}_{p''}(\mathbb{H}_i|\overline{e_{i+1}}) - R^{\sigma''}_{p'}(\mathbb{H}_i|\overline{e_{i+1}}) \leq R^{\sigma''}_{p'}(\mathbb{H}_i|\overline{e_{i+1}}) - R^{\sigma''}_{p}(\mathbb{H}_i|\overline{e_{i+1}}) \tag{9.6}$$

Note that,

$$\begin{aligned}
R^{\sigma''}_{p}(\mathbb{H}_{i+1}) &= pR^{\sigma''}_{p}(\mathbb{H}_i|e_{i+1}) + (1-p)R^{\sigma''}_{p}(\mathbb{H}_i|\overline{e_{i+1}}) && (9.7)\\
R^{\sigma''}_{p'}(\mathbb{H}_{i+1}) &= p'R^{\sigma''}_{p'}(\mathbb{H}_i|e_{i+1}) + (1-p')R^{\sigma''}_{p'}(\mathbb{H}_i|\overline{e_{i+1}}) && (9.8)\\
R^{\sigma''}_{p''}(\mathbb{H}_{i+1}) &= p''R^{\sigma''}_{p''}(\mathbb{H}_i|e_{i+1}) + (1-p'')R^{\sigma''}_{p''}(\mathbb{H}_i|\overline{e_{i+1}}) && (9.9)
\end{aligned}$$

Let,

$$\begin{aligned}
R^{\sigma''}_{p''}(\mathbb{H}_i|e_{i+1}) - R^{\sigma''}_{p'}(\mathbb{H}_i|e_{i+1}) &= B_5 && (9.10)\\
R^{\sigma''}_{p''}(\mathbb{H}_i|\overline{e_{i+1}}) - R^{\sigma''}_{p'}(\mathbb{H}_i|\overline{e_{i+1}}) &= B_6 && (9.11)\\
R^{\sigma''}_{p'}(\mathbb{H}_i|e_{i+1}) - R^{\sigma''}_{p}(\mathbb{H}_i|e_{i+1}) &= B_7 && (9.12)\\
R^{\sigma''}_{p'}(\mathbb{H}_i|\overline{e_{i+1}}) - R^{\sigma''}_{p}(\mathbb{H}_i|\overline{e_{i+1}}) &= B_8 && (9.13)
\end{aligned}$$

Since $p' = p + \epsilon$ and $p'' = p' + \epsilon$, so from eq 9.8 and 9.9,

$$\begin{aligned}
&R^{\sigma''}_{p''}(\mathbb{H}_{i+1}) - R^{\sigma''}_{p'}(\mathbb{H}_{i+1})\\
=&p'\left(R^{\sigma''}_{p''}(\mathbb{H}_i|e_{i+1}) - R^{\sigma''}_{p'}(\mathbb{H}_i|e_{i+1})\right) + \epsilon R^{\sigma''}_{p''}(\mathbb{H}_i|e_{i+1})\\
&+ (1-p')\left(R^{\sigma''}_{p''}(\mathbb{H}_i|\overline{e_{i+1}}) - R^{\sigma''}_{p'}(\mathbb{H}_i|\overline{e_{i+1}})\right) - \epsilon R^{\sigma''}_{p''}(\mathbb{H}_i|\overline{e_{i+1}})\\
=&p'B_5 + (1-p')B_6 + \epsilon R^{\sigma''}_{p''}(\mathbb{H}_i|e_{i+1}) - \epsilon R^{\sigma''}_{p''}(\mathbb{H}_i|\overline{e_{i+1}})\\
=&pB_5 + (1-p)B_6 + \epsilon(B_5 - B_6) + \epsilon\left(R^{\sigma''}_{p''}(\mathbb{H}_i|e_{i+1}) - R^{\sigma''}_{p''}(\mathbb{H}_i|\overline{e_{i+1}})\right) && (9.14)
\end{aligned}$$

Similarly from eq 9.7 and 9.8,

$$\begin{aligned}
&R^{\sigma''}_{p'}(\mathbb{H}_{i+1}) - R^{\sigma''}_{p}(\mathbb{H}_{i+1})\\
=&pB_7 + (1-p)B_8 + \epsilon\left(R^{\sigma''}_{p'}(\mathbb{H}_i|e_{i+1}) - R^{\sigma''}_{p'}(\mathbb{H}_i|\overline{e_{i+1}})\right) && (9.15)
\end{aligned}$$

Now, according to lemma 26,

$$\begin{aligned}
& B_5 \leq B_6 \\
\implies & R_{p''}^{\sigma''}(\mathbb{H}_i|e_{i+1}) - R_{p'}^{\sigma''}(\mathbb{H}_i|e_{i+1}) \leq R_{p''}^{\sigma''}(\mathbb{H}_i|\overline{e_{i+1}}) - R_{p'}^{\sigma''}(\mathbb{H}_i|\overline{e_{i+1}}) \\
\implies & R_{p''}^{\sigma''}(\mathbb{H}_i|e_{i+1}) - R_{p''}^{\sigma''}(\mathbb{H}_i|\overline{e_{i+1}}) \leq R_{p'}^{\sigma''}(\mathbb{H}_i|e_{i+1}) - R_{p'}^{\sigma''}(\mathbb{H}_i|\overline{e_{i+1}}) \qquad (9.16)
\end{aligned}$$

Also, $B_5 \leq B_7$ and $B_6 \leq B_8$ according to eq 9.5 and 9.6 respectively. Therefore from eq 9.14 and 9.15 and using eq 9.16, we have, $R_{p''}^{\sigma''}(\mathbb{H}_{i+1}) - R_{p'}^{\sigma''}(\mathbb{H}_{i+1}) \leq R_{p'}^{\sigma''}(\mathbb{H}_{i+1}) - R_{p}^{\sigma''}(\mathbb{H}_{i+1})$.

□

Algorithm 7: GreedyMaxAttackerStrategy

```
input  : Defender strategy $\boldsymbol{\alpha}$,
         Attacker budget $B$
output: Attacker strategy $\boldsymbol{\beta}$
Initialize $\beta_e = 0, \forall e \in E$
While $\sum \beta_e \leq B$:
    • Find $e$ maximizing $f_{\boldsymbol{\alpha}}(\boldsymbol{\beta} + \epsilon\chi_e) - f_{\boldsymbol{\alpha}}(\boldsymbol{\beta})$ among all $e \in E$
    • $\boldsymbol{\beta} = \boldsymbol{\beta} + \epsilon\chi_e$.
end While
```

Theorem 6. *Suppose the attacker has the following discrete strategy space along each edge e, $\beta_e \in \{0, \delta, 2\delta, ..., q\delta\}$, where $q\delta \leq 1$ for some constant $0 < \delta \leq 1$. Then given a defender strategy $\boldsymbol{\alpha}$ and attacker budget B, the greedy method* GreedyMaxAttackerStrategy *to compute the best attacker strategy that maximizes its gain has approximation factor $(1 - \frac{1}{e})$.*

The theorem follows from the submodular function maximization work of Soma et al. [114] and theorem 5.

9.6 Computing gains

Computing gains in FlipNet becomes challenging due to the network structure. In this section we will discuss methods to compute gains for a special case – when the graph is a tree graph and the players play uniform move strategies across the network.

Let the defender's uniform move probability be α and the attacker's uniform move probability be β. In the rest of the discussion in this subsection, we assume a DAG for the network.

Let $Z_i^D(t)$ denote the waiting time of the defender computed at time t till his last move at node i; Let $Z_i^A(t)$ be defined similarly. Note that, not all attacker moves are successful (some attacker moves fail because of no control at the dependent node at the time of the move). Let $\mathcal{Z}_i^A(t)$ denote the waiting time of the attacker computed at t till his last successful move at node i. It is easy to see that for any node $v_i \in V$, time t and waiting time $x, 0 \leq x \leq t$, the probabilities of waiting time are the following,

$$Pr\left(Z_i^D(t) = x\right) = \alpha(1-\alpha)^x \tag{9.17}$$

$$Pr\left(Z_i^D(t) > x\right) = (1-\alpha)^{\alpha+1} \tag{9.18}$$

$$Pr\left(Z_i^A(t) = x\right) = \beta(1-\beta)^x \tag{9.19}$$

It is easy to that, for a node i which is exactly one hop away from the attack source (i.e., edge $(s,i) \in E$ where $s \in S$) and for $t = 0, 1, 2, ...\infty$,

$$Pr\left(C_i^A(t) = 1\right) = \sum_{x=0}^{t} \beta(1-\beta)^x(1-\alpha)^{\alpha+1} \tag{9.20}$$

Now consider a node j which is exactly two hops away from the source and let i be the parent of j. The time elapsed at t since the last successful attacker move at v_j, $\mathcal{Z}_j^A(t)$ takes the value x, where $0 \leq x \leq t$ with the following probability,

$$\begin{aligned} Pr\left(\mathcal{Z}_j^A(t) = x\right) &= Pr\left(Z_j^A(t) = 0\right) Pr\left(C_i^A(t) = 0\right) Pr\left(\mathcal{Z}_j^A(t-1) = x-1\right) \\ &\quad + Pr\left(Z_j^A(t) = 1\right) Pr\left(C_i^A(t-1) = 0\right) Pr\left(\mathcal{Z}_j^A(t-2) = x-2\right) \\ &\quad . \\ &\quad . \\ &\quad + Pr\left(Z_j^A(t) = x\right) Pr\left(C_i^A(t-x) = 1\right) \end{aligned}$$

In general, for any node v_j and its parent node v_i and $0 \leq x \leq t$, the following is true,

$$\begin{aligned} Pr\left(\mathcal{Z}_j^A(t) = x\right) &= Pr\left(Z_j^A(t) = 0\right) Pr\left(C_i^A(t) = 0\right) Pr\left(\mathcal{Z}_j^A(t-1) = x-1\right) \\ &\quad + Pr\left(Z_j^A(t) = 1\right) Pr\left(C_i^A(t-1) = 0\right) Pr\left(\mathcal{Z}_j^A(t-2) = x-2\right) \\ &\quad . \\ &\quad . \\ &\quad + Pr\left(Z_j^A(t) = x\right) Pr\left(C_i^A(t-x) = 1\right) \end{aligned} \tag{9.21}$$

Finally,

$$Pr\left(C_j^A(t) = 1\right) = \sum_{x=0}^{t} Pr\left(\mathcal{Z}_j^A(t) = x\right).Pr\left(Z_j^D(t) > x\right) \tag{9.22}$$

Therefore, given the defender and attacker move probabilities α and β respectively, we can use dynamic programming to compute the gain of the attacker; this in turn allows us to compute the defender's gain. For instance, the attacker's total gain in a path graph $s, v_1, \ldots, v_n$ up to time $t = T$ can be computed as shown in algorithm 8; note that, the attacker's tatal gain is $C^A = \sum_{i=1}^{n} \sum_{t=0}^{T} Pr\left(C_i^A(t) = 1\right)$.

Algorithm 8: Compute Attacker Gain, C^A

```
Initialize $C^A = 0$
for $t = 0$ to $t = T$ do
  Compute $C_1^A(t)$ according to 9.20
  $C^A = C^A + C_1^A(t)$
end for
for $i = 2$ to $i = n$ do
  for $t = 0$ to $t = T$ do
    for $x = 0$ to $x = t$ do
      Compute $Pr\left(\mathcal{Z}_i^A(t) = x\right)$ according to 9.21
    end for
    Compute $Pr\left(C_i^A(t) = 1\right)$ according to 9.22
    $C^A = C^A + Pr\left(C_i^A(t) = 1\right)$
  end for
end for
```

9.7 Extension: FlipNetUndir

9.7.1 FlipNetUndir

We now study the FlipNetUndir problem. In this problem setting, the attack graph is undirected and the attacks can propagate both ways along an edge. If we consider the constraint on the attacker's part that he can attack a seed node only once, then the following lemma follows from the characterization of [40] for the SIS model of epidemics.

Lemma 27. *For an instance $\mathcal{F} = (G, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\kappa^D}, \boldsymbol{\kappa^A})$ of* FlipNetUndir *for exponential move strategy class, with the attack starting at a node $s \in V_S$, if the attack is not repeated at the source, then the defender has control of the entire network in the limiting distribution.*

We can also show that the duration of time for which the attacker has control over any node can be characterized in terms of the spectral properties, building on the characterization of [40] for the SIS model of epidemics.

Lemma 28. *For an instance $\mathcal{F} = (G, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\kappa^D}, \boldsymbol{\kappa^A})$ of* FlipNetUndir*, let τ denote the number of time steps for which the attacker has control of at least one node in the network. Define matrix $M = (\max\{\beta_{(i,j)}, \beta_{(j,i)}\})$. Let $\delta = \min_x \alpha_x$. Let $\rho(M)$ denote the spectral radius of the matrix M (i.e., the largest eigenvalue of M). Then, if $\rho(M) < \delta$, we have $E[\tau] = O(\frac{\log n}{\delta - \rho(M)})$.*

9.7.2 FlipNetCumulative

In this problem setting, both the defender and attacker make moves independently at each node, as they do in the basic model. However, unlike FlipNet, the attacker controls node i at any point of time t, if and only if, it is in control of i and all its predecessor nodes as well at t.

We show a submodularity property of the players' gains when the graph is a path graph and players play exponential strategies. Let the strategy of the defender and the attacker be the rate vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively. In a path graph of n nodes, the probability that the attacker controls the i^{th} node is as follows,

$$Pr\left(C_i^A(t) = 1\right) = Pr\left(Z_i^A(t) < Z_i^D(t) \wedge C_{i-1}^A(t) = 1 \wedge ... \wedge C_1^A(t) = 1\right)$$

According to this definition, it is easy to see that if $C_{i-1}^A(t) = 1$, then it infers $C_j^A(t) = 1$ for any $1 \leq j < i$. Therefore,

$$Pr\left(C_i^A(t) = 1\right) = Pr\left(Z_i^A(t) < Z_i^D(t) \wedge C_{i-1}^A(t) = 1\right)$$

Therefore, the control of the node i can be considered as a FlipThem system as described in [69], whose control depends on nodes $1, 2, ..., i$ in AND control model. So, if both the defender and the attacker move with exponential move strategies with rate vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively, then according to [69], the attacker's gain at node i is,

$$\gamma_i^A = \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j}$$

Therefore, the total gain of the attacker is,

$$\gamma^A = \sum_{i=1}^{n} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j}$$

We can show the following submodularity property of the players' gains.

Lemma 29. *The gain of the defender and the attacker in* FLIPNETCUMULATIVE *in a path graph $s, v_1, v_2, ..., v_n$ is submodular w.r.t. α_k and β_k respectively for any $1 \leq k \leq n$ when $\alpha_i > 0, \forall i$.*

Proof. First we show the property for the attacker. Note that,

$$\begin{aligned}
\gamma^A &= \sum_{i=1}^{n} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} \\
&= \sum_{i=1}^{k-1} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} + \sum_{i=k}^{n} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} \\
&= \sum_{i=1}^{k-1} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} + \left(\prod_{i=1}^{k-1} \frac{\beta_i}{\alpha_i + \beta_i} \right) \frac{\beta_k}{\alpha_k + \beta_k} \left(1 + \sum_{i=k+1}^{n} \prod_{j=k+1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} \right)
\end{aligned}$$

Suppose,

$$\left(\prod_{i=1}^{k-1} \frac{\beta_i}{\alpha_i + \beta_i} \right) \left(1 + \sum_{i=k+1}^{n} \prod_{j=k+1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} \right) = X$$

Then, taking first and second order derivative of γ^A,

$$\begin{aligned}
\frac{\partial \gamma^A}{\partial \beta_k} &= \frac{1}{\alpha_k + \beta_k} X - \frac{\beta_k}{(\alpha_k + \beta_k)^2} X \\
\frac{\partial^2 \gamma^A}{\partial {\beta_k}^2} &= -\frac{1}{(\alpha_k + \beta_k)^2} X - \frac{1}{(\alpha_k + \beta_k)^2} X + \frac{2\beta_k}{(\alpha_k + \beta_k)^3} X \\
&= -\frac{2}{(\alpha_k + \beta_k)^2} X + \frac{2}{(\alpha_k + \beta_k)^2} \times \frac{\beta_k}{\alpha_k + \beta_k} X \\
&\leq 0
\end{aligned}$$

Therefore, γ^A is submodular w.r.t. β_k for any $1 \leq k \leq n$.

Similarly we can prove the lemma for the defender as follows.

$$\begin{aligned}
\gamma^D &= n - \sum_{i=1}^{n} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} \\
&= n - \sum_{i=1}^{k-1} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} - \sum_{i=k}^{n} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} \\
&= n - \sum_{i=1}^{k-1} \prod_{j=1}^{i} \frac{\beta_j}{\alpha_j + \beta_j} - \sum_{i=k}^{n} \left(\prod_{j=1}^{k-1} \frac{\beta_j}{\alpha_j + \beta_j} \right) \frac{\beta_k}{\alpha_k + \beta_k} \left(\prod_{l=k+1}^{i} \frac{\beta_l}{\alpha_l + \beta_l} \right) \\
\frac{\partial \gamma^D}{\partial \alpha_k} &= \sum_{i=k}^{n} \left(\prod_{j=1}^{k-1} \frac{\beta_j}{\alpha_j + \beta_j} \right) \frac{\beta_k}{(\alpha_k + \beta_k)^2} \left(\prod_{l=k+1}^{i} \frac{\beta_l}{\alpha_l + \beta_l} \right) \\
\frac{\partial^2 \gamma^D}{\partial {\alpha_k}^2} &= -\sum_{i=k}^{n} \left(\prod_{j=1}^{k-1} \frac{\beta_j}{\alpha_j + \beta_j} \right) \frac{2\beta_k}{(\alpha_k + \beta_k)^3} \left(\prod_{l=k+1}^{i} \frac{\beta_l}{\alpha_l + \beta_l} \right) \\
&\leq 0
\end{aligned}$$

□

The following corollary follows from this lemma.

Corollary 7. Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be the vectors of integers representing the exponential move rates of the defender and the attacker respectively at the nodes of a path graph in FlipNetCumulative game; let X_i be the characteristic vector of i where i represents a node. Then, for any node i in the path graph, their individual gains have the diminishing marginal return property:

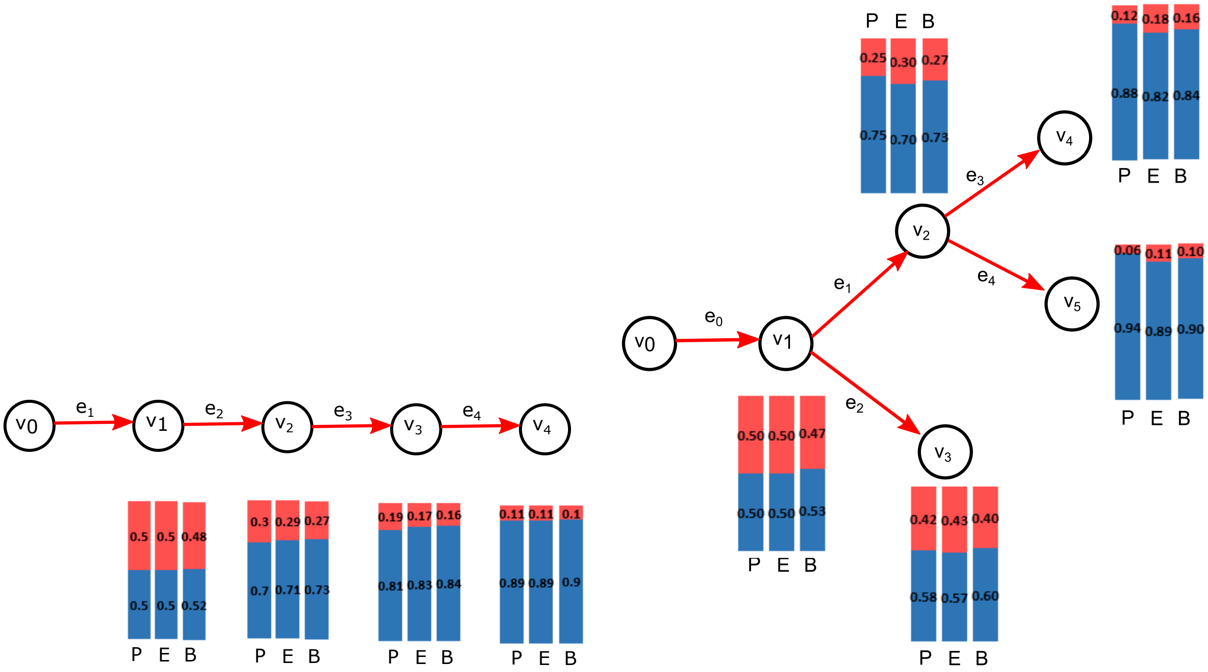
$$\gamma^D(\boldsymbol{\alpha} + X_i, \boldsymbol{\beta}) - \gamma^D(\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq \gamma^D(\boldsymbol{\alpha} + 2X_i, \boldsymbol{\beta}) - \gamma^D(\boldsymbol{\alpha} + X_i, \boldsymbol{\beta})$$

and,

$$\gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta} + X_i) - \gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq \gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta} + 2X_i) - \gamma^A(\boldsymbol{\alpha}, \boldsymbol{\beta} + X_i).$$

9.8 Experimental Results

We empirically investigate the structure of the best response strategies of the players, the existence of equilibria and the impact of various network parameters on the effective defender strategies. The players' utility on different nodes of the networks depends on the network



(a) Path graph

(b) Tree graph

Figure 9.4: Example graphs and gains of the defender and attacker. The players' gains are simulated for (i) periodical (P), (ii) exponential (E), (iii) Bernoulli move strategies. In path graph, both the attacker and the defender play at each node v_i with average move parameter (i.e., move rate or move probability) 0.1. In the tree graph, the defender and the attacker play with average move parameter 0.1 at each node except in v_3 and v_5 where it plays with move parameter 0.2 and 0.05 respectively. The gain values are calculated by simulating FlipNet for 200 instances of 4000 time units each. The attacker's gain keeps decreasing at subsequent hops. For different move strategies, the change in the gains are similar.

structure. Due to attacker's control restrictions in the model, the attacker gets comparatively lesser control over the nodes that are farther from the source. However, depending on the network structure, there can be multiple ways for the attacker to reach a node. These notions, coupled with the move cost of the players, decide the players' utilities and best move strategies. In the experiments we have used simulations to compute utilities.

Figure 9.4 also shows that the gains are similar for playing different strategies, i.e., periodic, Bernoulli and exponential moves, with the average move rates being the same. Hence, we experiment with Bernoulli move strategies, since gains are less expensive to compute in Bernoulli move games.

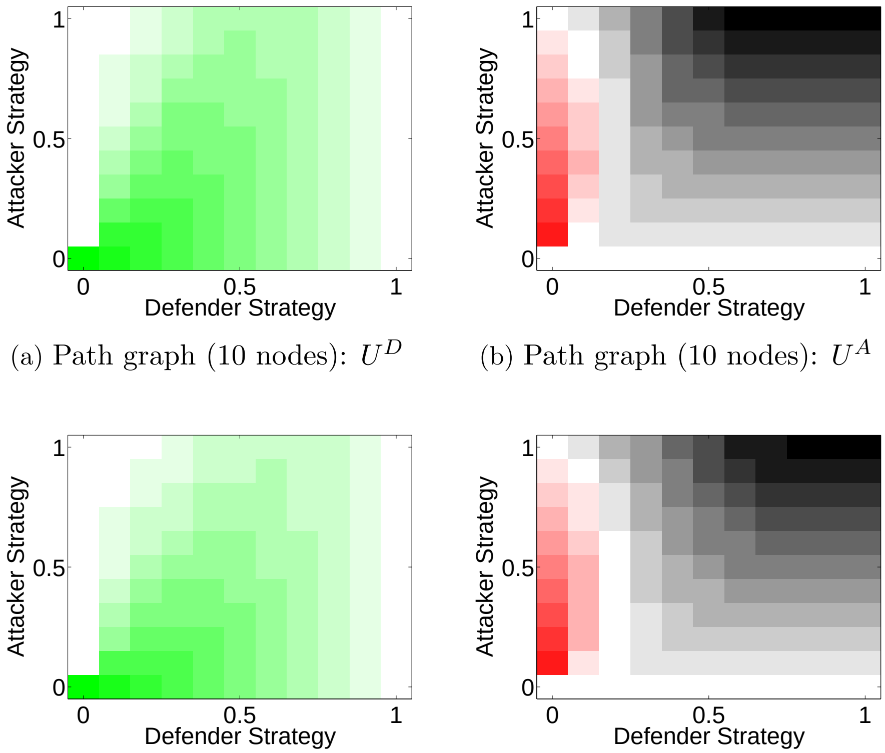
Given an attack strategy, two quantities of interest from a system owner's point of view are the "best" and the "critical" defender strategy. The "best" defender strategy is defined as the one that maximizes the defender's utility. We define the "critical" defender strategy as the cheapest move strategy that makes the attacker utility zero and thereby force him to drop out.

A natural question to ask is what network properties affect the effective defender strategies. As we have found, two graph parameters - graph density and "graph depth" impact the best and the critical defender strategies. The graph density is defined as $\frac{|E|}{|V|(|V|-1)}$ in directed graphs and $\frac{2|E|}{|V|(|V|-1)}$ in undirected graphs. We define the "graph depth" as the average shortest distance of a node from a source $s \in S$. In the following, we show experimental results that aims to answer the following questions:

1. What can we say about the existence of Nash equilibria in simple versions of FlipNet?
2. How graph depth and density affect defender's best and critical strategies?
3. What is the structure of optimal distribution of attacker's budget?

9.8.1 Dominant Strategies and Nash Equilibria

The dominant strategy of a player is the strategy that is better than any other strategy regardless of what the opponent plays. In FlipNet we have investigated the existence of dominant strategies in uniform discrete move games. Figure 9.5 shows the best defender and attacker strategy given the opponent's strategy in a grid for uniform strategy games in which both the defender and attacker strategies are chosen from a discrete strategy space - their move probabilities are chosen from $\{0, 0.1, 0.2, ..., 0.9, 1\}$. As the figure shows, the defender does not have a dominant strategy (there is no single best defender strategy against any attacker strategy); the same is true for the attacker. Since there is no dominant strategy in this game, therefore no unique pure strategy Nash equilibrium exists in general in discrete uniform strategy games. Also, simulation of repeated best response of the players show that the best response strategy oscillates and does not converge; figure 9.6 shows it for path graph. Therefore, a pure NE also does not exist in a uniform discrete move FlipNet game.



(a) Path graph (10 nodes): U^D

(b) Path graph (10 nodes): U^A

(c) CBT (15 nodes): U^D

(d) CBT (15 nodes): U^A

Figure 9.5: Defender's (left) and the attacker's (right) utility in (i) path graph of 10 nodes and (ii) complete binary tree (CBT) of 15 nodes. Utilities are shown as a function of their uniform move probabilities α and β. Darker shades of green and red correspond to higher defender and attacker utility respectively; white squares correspond to zero utility, and black shades correspond to negative utility.

9.8.2 Critical Strategies

Given a strategy of the attacker, we define the "critical defender strategy" as the cheapest defender strategy that makes the attacker utility zero. In other words, critical defender strategy is the cheapest defender strategy that makes an attacker strategy unsustainable. We investigate the structure of critical defender strategy in different networks and against varying attacker strength in discrete uniform strategy games. Figure 9.7 shows that as the attacker move probability increases, the critical move probability of the defender increases and reaches a peak before decreasing. Also note that the critical defender strategy depends on the networks.

Therefore, one natural question to ask is what network properties affect the critical defender strategy. We have observed in our experiments that graph density and "graph depth" are two network properties that affect the critical defender strategy. The graph density is defined as

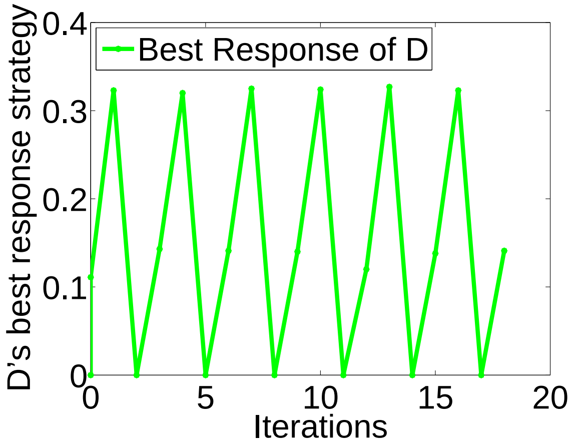


Figure 9.6: Iterative Best response plot of the defender – the plot shows that there is no pure Nash equilibrium in uniform move discrete strategy games.

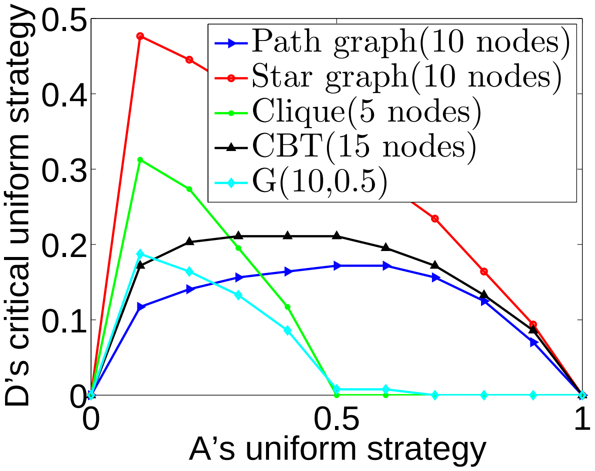
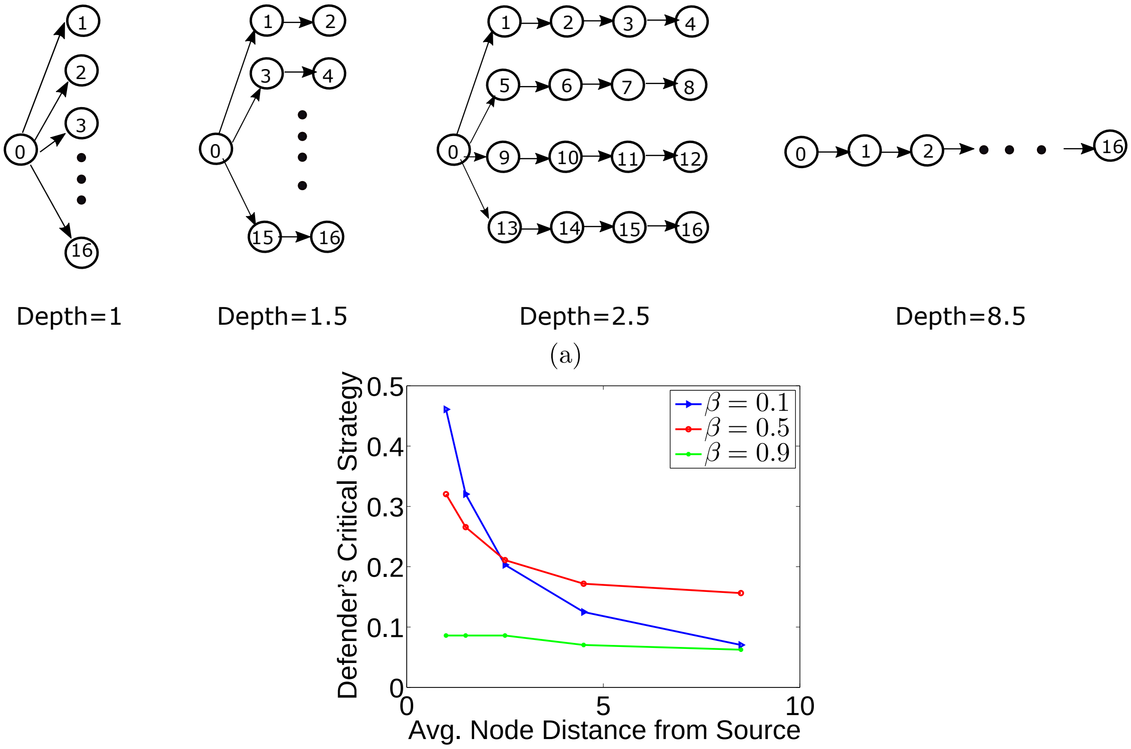


Figure 9.7: Plot of the critical uniform strategy of the defender (D) given a uniform strategy of the attacker (A) in different networks - path graph, start graph, complete graph, complete binary tree (CBT) and Erdős-Rényi graph with parameters $n = 10$ and $p = 0.5$. For a given attacker strategy, the critical defender strategy is the one that makes the attacker drop out, i.e., the attacker finds no incentive to play that strategy.

$\frac{|E|}{|V|(|V|-1)}$ in directed graphs and $\frac{2|E|}{|V|(|V|-1)}$ in undirected graphs. We define the "graph depth" as the average shortest distance of a node from a source $s \in S$.

Consider the tree graphs in figure 9.8a; they have the same number of nodes and the same graph density but varying graph depth. As figure 9.8b shows, the critical defender strategy decreases with the increase in graph depth. We see the same pattern in random Barabasi-albert trees. Barabasi-Albert trees are basically the Barabasi-Albert graphs with $m = 1$. As



(a)

(b)

Figure 9.8: Critical defender strategy on trees of varying depth. The critical strategy decreases with increased depth. (a) shows trees of the same number of nodes but varying depth and (b) shows the variation in critical strategy.

figure 9.9 shows, the same relationship between critical defender strategy and graph depth holds in random Barabasi-Albert trees.

Similarly, we experiment the effect of graph density on critical defender strategy. Consider the graph of figure 9.10a. As the dotted edges are added, the density of the graph increases, but the number of nodes and the depth of the graph remain the same. Figure 9.10b shows that, as the graph density increases the critical defender strategy increases. Similar results are found in the case of Erdős-Rényi graphs (figure 9.11)as the density parameter, p is varied.

Therefore from the defender's point of view, higher graph depth and lower graph density is good.

9.8.3 Best Response Strategy of the Defender

Given a fixed uniform move strategy, β of the attacker, the defender's gain increases monotonically with its increased move probability, α which is obvious from the model. However,

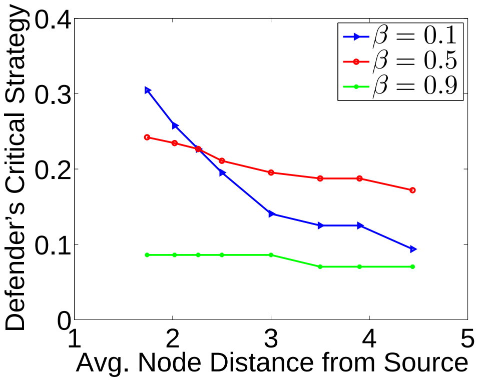
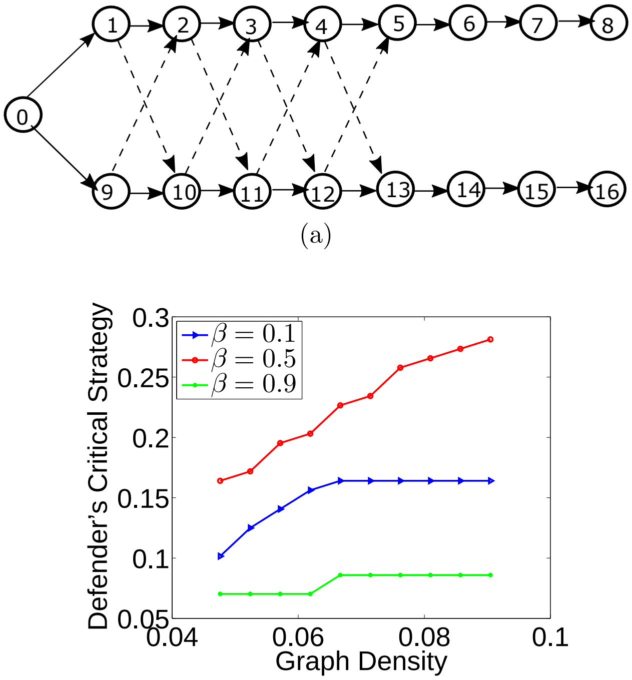
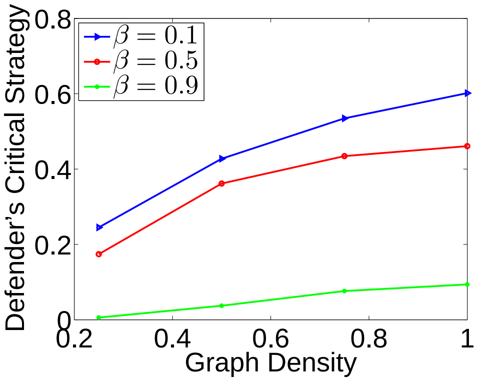


Figure 9.9: Critical defender strategy on Barabasi-Albert trees of 50 nodes and varying depth and for different attacker strategies. For each depth, 10 randomly generated graphs were considered and the average critical strategy is shown in the plots.



(b)

Figure 9.10: Critical defender strategy on graphs of varying density but fixed depth. The critical strategy increases with increased density. (a) shows the graphs; as the dotted edges are added the graph density increases, but graph depth remains the same. Plots in (b) show the variation in critical strategy.



(a) $G(50, p)$

Figure 9.11: Critical defender strategy on $G(n, p)$ of $n = 50$ and varying p (density) and for different attacker strategies. For each density, 25 randomly generated graphs were considered and the average critical strategy is shown in the plots.

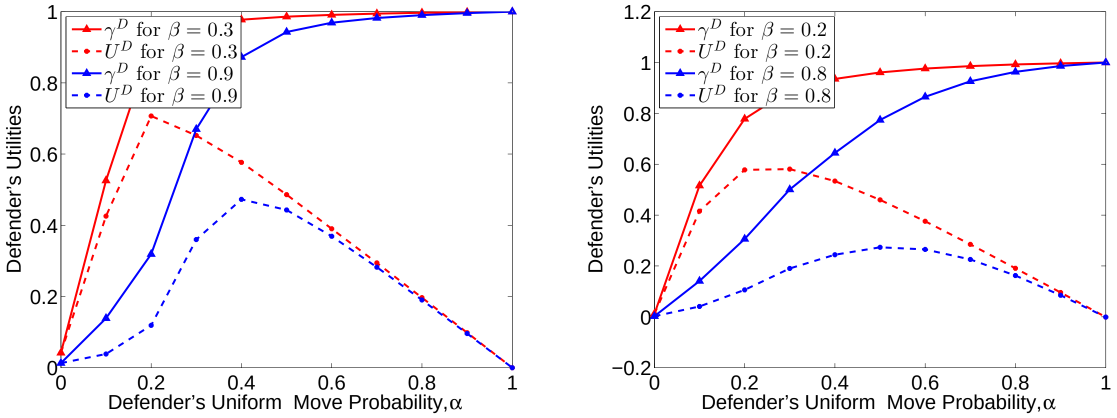
the utility does not increase monotonically with increased α due to the cost that the defender pays for making moves. This is seen in figure 9.12 for various graphs. So, there exists a best defender strategy against a fixed attacker strategy. We study how the best defender strategy varies with varying attacker strategy in different graphs.

As figure 9.13 shows, the defender's best utility always decreases with increased attacker move probability. However, this also depends on the network. In path graphs and complete binary trees, the defender always has to move faster for best utility against a faster attacker. However, in complete graphs and Erdős-Rényi graphs, there exists a peak value for the defender's best move probability. That is, beyond a certain attacker strength the defender cannot afford to play faster to achieve his best utility. This is because, after that peak point, the extent of marginal gain for moving faster becomes less than the increased cost that the defender pays. This results in the slowing down of the defender for the best utility.

Similar to critical defender strategy, we observe that the graph depth and graph density affects the best defender strategies. In Barabasi-Albert trees, as the graph depth increases the best defender strategy decreases and the best defender utility increases (figure 9.14). Similarly in Erdős-Rényi graphs, as the graph density increases, the defender's best strategy increases and the best utility decreases (figure 9.15).

9.8.4 Distribution of Attacker Budget

We use the GreedyMaxAttackerStrategy as described in section 9.5 to approximate the best strategy of a budgeted attacker against a given defender strategy. In this case, the budget refers to the total expected move cost over the edges in the network that the attacker



(a) Path graph - 25 nodes

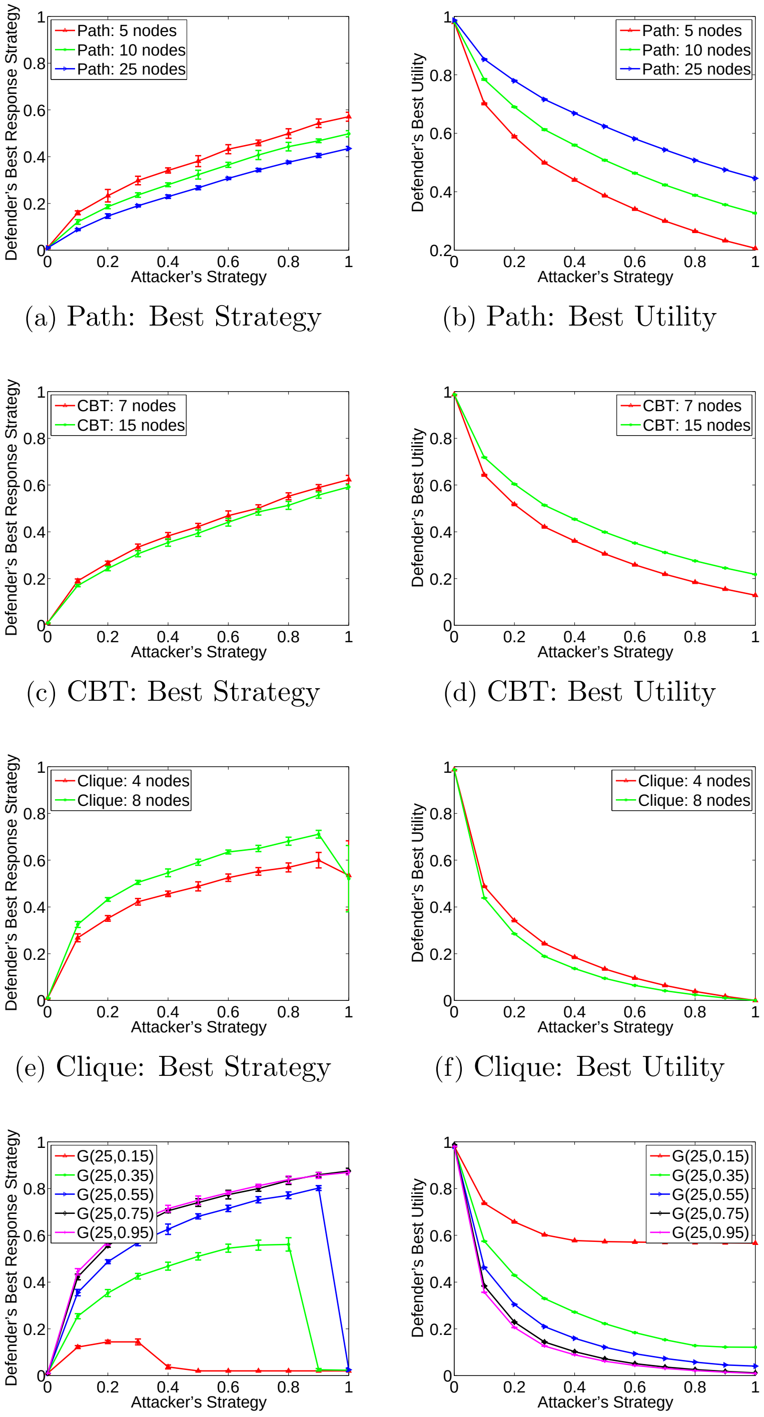
(b) Complete Binary Tree - 15 nodes

Figure 9.12: Impact of defender's uniform move probability, α on the defender's utilities - gain, γ^D and benefit,U^D. Although the gain monotonically increases with the increase in α, the benefit decreases after some point when the move cost becomes more significant compared to the the added gain. In the plots, the move cost of both the players are uniform across the network.

can afford; remember that, if the cost per move is 1, then the expected move cost is equal to the move probability. We study the structure of the attacker's budget allocation for a given defender strategy in different graphs. We observe that, if the move costs are uniform across the network, the attacker tends to move more frequently near the source nodes. Figure 9.16 shows that, as the edge distance from the source becomes greater, the attacker distributes lesser budget (or moves with lesser probability). The same pattern is observed for different graphs - path, complete binary tree, Erdos-Renyi graph and star graph. This suggests that the defender also should move more frequently near the source node to minimize attacker's control. For example, while the tightest security is usually deployed in the back end of an enterprise network, the defender should monitor more in the front end (or near the perimeter) to minimize attack footprint.

9.9 Conclusion

Studying persistent and covert attacks in networks is a fundamental problem in network security. We have presented FlipNet as a novel security game, which provides a very rich framework for these problems. We study the structure and complexity of best response strategy and NE in such games using analytical as well as simulation based techniques. Developing more efficient algorithms for computing properties of these games, and understanding the effects of network properties and parameters of the game can help in obtaining better insights into the security of networked systems. This includes identifying the most



(a) Path: Best Strategy

(b) Path: Best Utility

(c) CBT: Best Strategy

(d) CBT: Best Utility

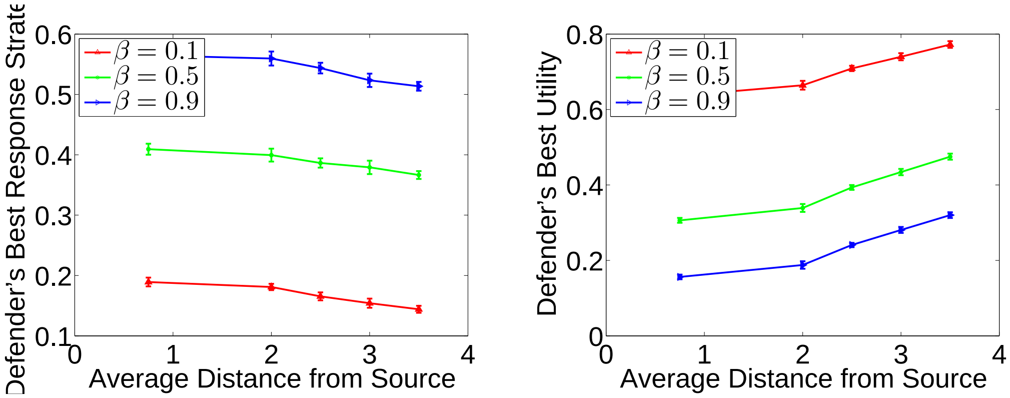
(e) Clique: Best Strategy

(f) Clique: Best Utility

(g) $G(25, p)$: Best Strategy

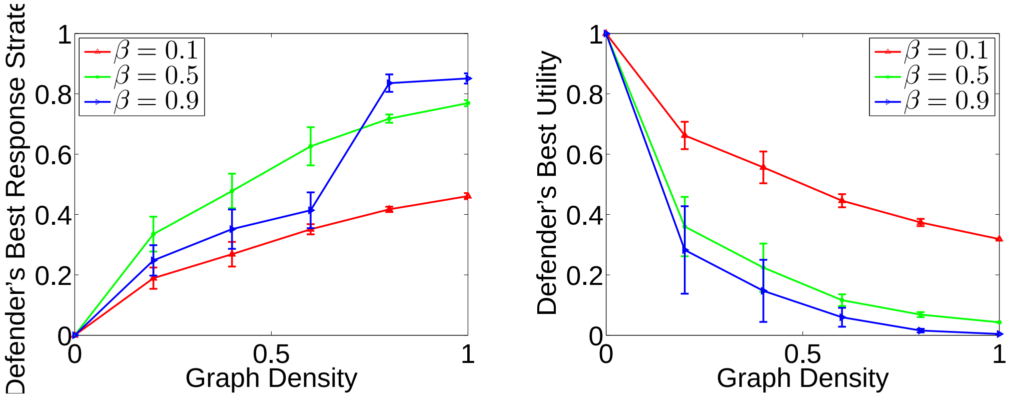
(h) $G(25, p)$: Best Utility

Figure 9.13: Defender's best response (uniform) strategy, i.e., best move probability and corresponding best utility for given attacker strategy (uniform) in path, complete binary tree (CBT), clique and Erdős-Rényi graphs of varying size and parameter. The move cost of both the players are 1 in all the plots.



(a) Barabasi-Albert Tree: Best Strategy (b) Barabasi-Albert Tree: Best Utility

Figure 9.14: Defender’s best response (uniform) strategy and corresponding best utility for different depths of nodes in Barabasi-Albert tree of 50 nodes. Defender’s best strategy decreases and best utility increases for increased depth of the graph.



(a) $G(25, p)$: Best Strategy (b) $G(25, p)$: Best Utility

Figure 9.15: Defender’s best response (uniform) strategy and corresponding best utility for different density of Erdős-Rényi graphs of 25 nodes. Defender’s best strategy increases and best utility decreases for increased density of the graph.

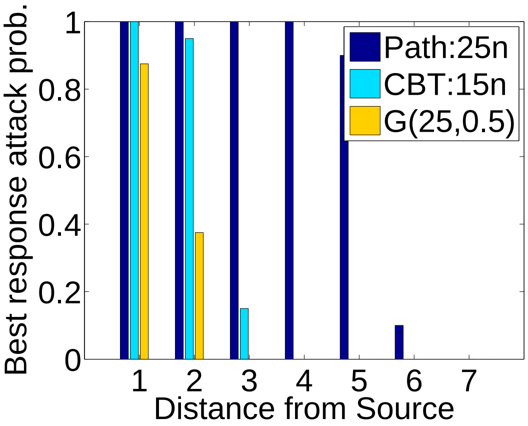


Figure 9.16: Attacker's best move probabilities along the edges of different graphs constrained by a budget and computed with greedy method. The figure shows that, for uniform move cost parameters in the network, the attacker finds it best to allocate most of the budget near the source nodes. The attacker allocates less and less budget at the edges further from the source of attack.

critical elements, and effective strategies for monitoring such systems.

Chapter 10

Vulnerability Analysis and Hardening Solutions

We investigate efficient security control methods for protecting against vulnerabilities in networked systems. A large number of interdependent vulnerabilities typically exist in the computing nodes of a cyber-system; as vulnerabilities get exploited, starting from low level ones, they open up the doors to more critical vulnerabilities. These cannot be understood just by a topological analysis of the network, and we use the attack graph abstraction of [30] to study these problems. In contrast to earlier approaches based on heuristics and evolutionary algorithms, we study rigorous methods for quantifying the inherent vulnerability and hardening cost for the system. We develop algorithms with provable approximation guarantees, and evaluate them for real and synthetic attack graphs.

10.1 Preliminaries

Formally, an attack graph, $\mathcal{A}$ is denoted by a tuple $(G = (V, E), \mathcal{F}, P, C)$, where: (i) $G = (V, E)$ is a directed acyclic graph, where the set V of nodes represent vulnerable system and network configurations (referred to as "attributes in [30]). An edge $e = (v_i, v_j) \in E$ represents an attack for which $v_i = pre(e)$ is the pre-condition and $v_j = post(e)$ is the post-condition attribute; (ii) $\mathcal{F}$ is a vector of node functions, with F_v being either AND or OR, (iii) $P(v)$ denotes the "potential damage", which quantifies the damage inflicted if vulnerability v is exploited; and (iv) $C(v)$ denotes the cost of securing external attribute $v \in L$. See Figure 10.2 and 10.2 for an example.

We describe some other notations that will be used in the rest of the paper. Let $d_{in}(v)$ and $d_{out}(v)$ denote the in-degree and out-degree of a node v in G. The set of attributes/nodes is divided into two subsets– external attributes (which correspond to the set L of leaves in G) and internal attributes (corresponding to the non-leaf nodes in $V - L$). We assume that each

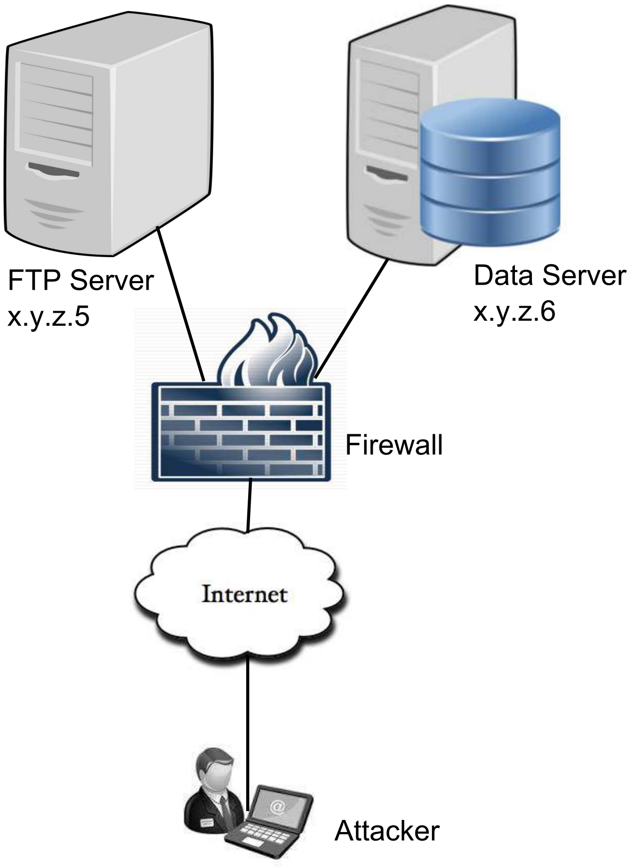


Figure 10.1: An example network: the data center, which is not directly accessible to the user can be attacked through the ftp server.

internal attribute is a post-condition of at least one attack. On the other hand, external attributes represent intrinsic vulnerabilities of the system which are not post-conditions of any attack, rather they act only as pre-conditions of other attacks. We use external attributes and leaf node interchangeably in the rest of the paper. We say that attribute v is true when the vulnerability represented by v is successfully compromised by the attacker; we use $I(v) = 1$ to denote this event. If $F_v = AND$, then $I(v) = 1$ if for all its pre-conditions u (i.e., for all $(u, v) \in E$), the attribute u holds (i.e., $I(u) = 1$). On the other hand, if $F_v = OR$, then $I(v) = 1$ if any one of its pre-conditions holds (i.e., $I(u) = 1$ for some $(u, v) \in E$). If $I(v) = 1$, i.e., if the vulnerability corresponding to v is exploited, there is a potential damage of $P(v) \geq 0$ for the defender. Finally, $C(v)$ denotes the defender's cost of securing external attribute $v \in L$. We use $C_a(v)$ to denote the adversary's cost for attacking leaf v. In most of the paper, we will assume $C_a(v) = C(v)$, for simplicity. A list of notations is given in table .

A defender wants to prevent attacks by securing some of the leaves. We use $\mathbf{a} : L \to \{0, 1\}$ to describe the defender's strategy. $\mathbf{a}(v) = 0$ denotes that the leaf node v is not protected;

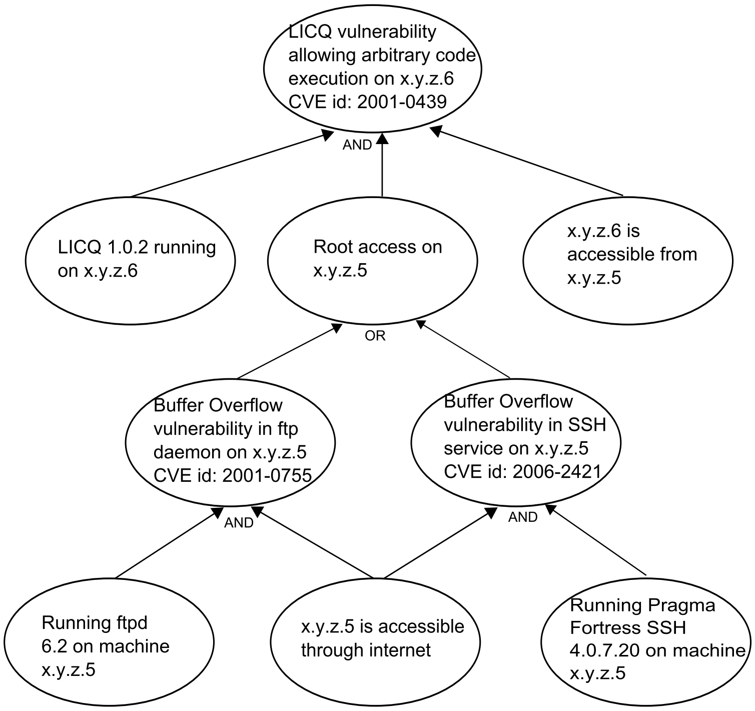

LICQ vulnerability allowing arbitrary code execution on x.y.z.6 CVE id: 2001-0439
AND
LICQ 1.0.2 running on x.y.z.6
Root access on x.y.z.5
x.y.z.6 is accessible from x.y.z.5
OR
Buffer Overflow vulnerability in ftp daemon on x.y.z.5 CVE id: 2001-0755
Buffer Overflow vulnerability in SSH service on x.y.z.5 CVE id: 2006-2421
AND
AND
Running ftpd 6.2 on machine x.y.z.5
x.y.z.5 is accessible through internet
Running Pragma Fortress SSH 4.0.7.20 on machine x.y.z.5

Figure 10.2: Attack graph representation for the network in Figure 10.1

1 means it is protected. The resulting cost for the defender is $\sum_v a(v)C(v)$. Similarly the attacker's strategy is described by $\mathbf{b} : L \to \{0,1\}$, with $b(v) = 1$ if leaf $v \in L$ is attacked. The cost for the attacker is $\sum_{v \in L} b(v)C(v)$. Let $I^{(\mathbf{a},\mathbf{b})}$ denote the truth value of attribute v when the defender and the attacker's strategies are $\mathbf{a}$ and $\mathbf{b}$, respectively. $I^{(\mathbf{a},\mathbf{b})}(v) = 1$ if the vulnerability associated with node v is realized. Given a defender's strategy $\mathbf{a}$ and attacker's strategy $\mathbf{b}$ in an attack, the total potential damage for the defender equals

$$U(\mathbf{a},\mathbf{b}) = \sum_{v \in V} I^{(\mathbf{a},\mathbf{b})}(v)P(v).$$

We now address two problems from the defender and the attacker's point of view. Given a limited budget, the defender wants to maximize its utility. This problem formulation relates to the defender's optimization problem discussed in [30]. Similarly we address the attacker's optimization problem as well where it wants to maximize its utility given a limited budget on the number of nodes that it can attack.

10.1.1 Characterizing the inherent vulnerability of the system

We formalize the vulnerability of the network to attackers with resource constraints as the maximum potential damage that an attacker can achieve who can target up to k leaves,

Table 10.1: Notations

G	Attack Graph $G = (V, E)$
V	Node set or attribute set
E	Set of directed edges
L	Set of leaf nodes or external attributes, $L \subseteq V$
$\mathcal{F}_v$	Decomposition of node $v \in V$; $\mathcal{F}(v) \in \big(0, 1\big]$
$P(v)$	Potential damage of node $v \in V$
$C(v)$	Cost of securing node $v \in V$
$\mathbf{a}$	Defender strategy
$\mathbf{b}$	Attacker strategy
$I^{\mathbf{a},\mathbf{b}}(v)$	Truth value of node v under **a** and **b**
U	Potential damage for the defender

given a defender strategy **a**. Thus, this problem quantifies the maximum damage possible by an attacker with resource constraint.

Given an attack graph instance $\mathcal{A} = (G = (V, E), \mathcal{F}, P, C)$, the MaxAttackerUtil$(\mathcal{A}, k)$ problem is to determine the maximum potential damage possible by an attacker who can attack at most k leaves in L.

10.1.2 Network hardening by the defender

We formalize the hardening problem from the defender's perspective in the face of an attacker who is limited to attacking at most k leaves.

Given an attack graph instance $\mathcal{A} = (G = (V, E), \mathcal{F}, P, C)$, the DefenderHardening (DefHard$(\mathcal{A}, k, B)$) problem is to determine the minimum cost strategy **a** that ensures that any attack on at most k leaves does not cause a total potential damage of more than B.

10.2 The MaxAttackerUtil problem

We first show that the MaxAttackerUtil problem is NP-complete. Then we describe a greedy approximation algorithm for attack graphs in which all node functions are OR.

Lemma 30. MaxAttackerUtil$(\mathcal{A}, k)$ *is NP-Complete.*

Proof. For a attacker strategy **b**, it is easy to verify if at most k leaf nodes are attacked in **b** and if the resulting potential damage is some given value B. Therefore, MaxAttackerUtil is in NP. We now prove that it is NP-hard by reduction from the Minimum Hitting Set problem, which is NP-complete. An instance of the Hitting Set problem consists of an a set system $(S, \mathcal{R})$ and a parameter b, where $\mathcal{R}$ is a collection of subsets of S, and the goal is to determine if there exists a subset $S' \subseteq S$ of size at most b that hits every set $R \in \mathcal{R}$.

We construct an attack graph instance in the following manner. The graph $G = (V, E)$ has three levels, $V = V_1 \cup V_2 \cup V_3$. Set V_1 is the leaf set, and is in direct correspondence with the set S; for notational simplicity, we just set $V_1 = S$, and refer to nodes in V_1 by elements of S. Similarly, V_2 is in direct correspondence with $\mathcal{R}$; we refer to nodes in V_2 by the corresponding sets $R \in \mathcal{R}$. Finally, $V_3 = \{r\}$. There is an edge (u, R) for $u \in V_1$ and $R \in V_2$ if and only if $u \in R$. Each node R in V_2 is associated with the function OR, i.e., F_R =OR. Finally, we have the edges (R, r) for all $R \in V_2$. The node function associated with r is F_r =AND. We have potential damage 0 for all nodes in V, except r. The cost $C(v)$ for all leaves $v \in V_1$ equals 1. Let $\mathcal{A}$ denote the resulting attack graph. We set $k = b$ and $B = 1$. We observe that for $k = b$, there is a hitting set of size b if and only if there is a set of k leaves that can be attacked in $\mathcal{A}$, resulting in a potential damage of $B = 1$. Therefore, the lemma follows.

□

10.2.1 Greedy Solution for OR-only decompositions

The MaxAttackerUtil can be solved easily with a simple greedy algorithm as shown in algorithm 9.

Algorithm 9: Algorithm GreedyMaxAttackerUtil

```
input  : $\mathcal{A}, k$
output: Leaf Node set $L^g$ to attack.
Initialize $L^g \leftarrow \phi$
Let $Q(S)$ be a function defined on $S \subseteq L$ which quantifies the utility that the attacker gets
  by attacking exactly the nodes in $S$.
While $|L^g| < k$:
    • Pick leaf node $l$ such that $l = \text{argmax}_{i:i \in L \setminus L^g} Q(L^g \cup \{i\})$.
    • $L^g = L^g \cup \{l\}$
```

Lemma 31. *Function $Q(S), S \subseteq L$ is submodular for an attack graph where $\mathcal{F}_v = OR, \forall v \in V$.*

Proof. Let $R(v)$ denote the set of nodes reachable from leaf node $v \in L$. Similarly let $R(L')$ denote the set of nodes reachable from leaf node set $L' \subseteq L$, i.e. $R(L') = \cup_{v \in L'} R(v)$.

Note that, this is the same set the attacker is able to compromise by attacking L' when $\mathcal{F}_v = OR, \forall v \in V$. Therefore in this case, $Q(S) = \sum_{v \in R(S)} P(v)$.

Let $L_S, L_B \subseteq L$ and $L_S \subseteq L_B$. For any $i \in L \setminus L_B$ we will show that $Q(L_S \cup \{i\}) - Q(L_S) \geq Q(L_B \cup \{i\}) - Q(L_B)$ and this will prove the submodularity of Q. Note that, $\cup_{l \in L_S} R(l) \subseteq \cup_{l \in L_B} R(l)$, since $L_S \subseteq L_B$. Therefore, $R(i) \setminus \cup_{l \in L_S} R(l) \supseteq R(i) \setminus \cup_{l \in L_B} R(l)$. Hence, the marginal payoff added by i to L_B is not more than the marginal payoff when i is added to L_S, i.e. $Q(L_S \cup \{i\}) - Q(L_S) \geq Q(L_B \cup \{i\}) - Q(L_B)$. Hence the lemma follows. □

Lemma 32. *Algorithm 9 approximates* $\text{MaxAttackerUtil}(\mathcal{A}, k)$ *with an approximation factor of* $(1 - \frac{1}{e})$*, when* $\mathcal{F}_v = OR, \forall v \in V$.

Proof. According to algorithm 9, the solution size $|L^g| = k$ and hence the solution is a feasible solution. We now prove that it has approximation ratio $(1 - \frac{1}{e})$.

It is easy to see that, when $\mathcal{F}_v = OR, \forall v \in V$, then $\text{MaxAttackerUtil}(\mathcal{A}, k)$ corresponds to maximizing $Q(S)$ where $S \subseteq L$ and $|S| \leq k$. Also note that, Algorithm 9 actually gives a greedy solution to maximizing $Q(S)$. According to Nemhauser et al. [82, 28, 52], if f is a non-negative, monotone and submodular function, then the greedy solution is a $(1 - \frac{1}{e})$-approximation for the problem of maximizing $f(S)$ subject to the constraint that $|S| = k$. It is easy to see that, $Q(S)$ is monotone and non-negative by definition. And according to lemma 31, it is submodular when $\mathcal{F}_v = OR$ for every node $v \in V$. Hence, a greedy solution to maximizing $Q(S)$, and therefore determining *MaxAttackerUtil*$(\mathcal{A}, k)$, has approximation factor $(1 - \frac{1}{e})$, when $\mathcal{F}_v = OR, \forall v \in V$. □

10.3 The DefenderHardening problem

In this section, we study the problem $\text{DefHard}(\mathcal{A}, k, B)$ for different k. We first discuss the hardness of this problem, and then discuss approximation algorithms.

Lemma 33. $\text{DefHard}(\mathcal{A}, k, B)$ *is NP-Hard.*

Proof. The NP-hardness of $\text{DefHard}(\mathcal{A}, k, B)$ can be shown by reducing the hitting set problem, which is NP-complete, to it. The goal is to determine if there exists a defender strategy $\mathbf{a}$ of size r or less such that the attacker cannot gain more than B by attacking at most k unprotected nodes.

An instance of the Hitting Set problem consists of an a set system $(S, \mathcal{R})$ and a parameter b, where $\mathcal{R}$ is a collection of subsets of S, and the goal is to determine if there exists a subset $S' \subseteq S$ of size at most b that hits every set $R \in \mathcal{R}$.

We construct an attack graph instance in the following manner. The graph $G = (V, E)$ has two levels, $V = V_1 \cup V_2$. Set V_1 is the leaf set, and is in direct correspondence with the set

S; for notational simplicity, we just set $V_1 = S$, and refer to nodes in V_1 by elements of S. Similarly, V_2 is in direct correspondence with $\mathcal{R}$; we refer to nodes in V_2 by the corresponding sets $R \in \mathcal{R}$. There is an edge (u, R) for $u \in V_1$ and $R \in V_2$ if and only if $u \in R$. Set the node function to AND for each non-leaf node and set the potential damage of each non-leaf node as c, a constant and that of each leaf node 0. Finally, set $k = \max_{R \in \mathcal{R}} |R|$, $r = b$ and $B = c - \epsilon$, where ϵ is a arbitrarily small positive number. This completes the reduction.

It is easy to see that, any solution to DefHard(G, k, B) of size r also gives a corresponding solution of HitSet(S, C) of size r and vice versa. Therefore, DefHard(G, k, B) is NP-Hard.

□

10.3.1 Algorithm for DefHard for constant k

When k is constant, the DefHard$(\mathcal{A}, k, B)$ can be solved with the greedy approach in Algorithm 10.

Algorithm 10: Algorithm GreedyDefHard

```
input  : $\mathcal{A}, k, B$
output: Leaf Node set $L^g$ that needs to be protected.
Initialize $L^g \leftarrow \phi$
Let $\mathcal{X}$ be the set of subsets $X_i \subseteq L$ such that $|X_i| \leq k$ and $P(X_i) \geq B$ and
  $P(X_i - \{x\}) < B, \forall x \in X_i$
While $\mathcal{X}$ is non-empty
    • Pick leaf node $l$ such that $l = \text{argmax}_i(\sum_j \{P(X_j) | i \in X_j, X_j \in \mathcal{X}\}$.
    • Remove all the sets $X_i$ from $\mathcal{X}$ such that $l \in X_i$
    • Include leaf node $l$ into $L^g$.
```

Lemma 34. *Algorithm 10 gives an $O(\log |L|)$ approximation to* DefHard$(\mathcal{A}, k, B)$ *and has running time $O(|L|^{k+1})$.*

Proof. First we prove that algorithm 10 produces a feasible output for the DefHard$(\mathcal{A}, k, b)$ problem, i.e., if the defender secures L^g, the leaf node set output by algorithm 10, the attacker's gain is limited to at most B when it attacks at most k leaf nodes. This is easy to see from the following. Any solution to DefHard$(\mathcal{A}, k, B)$ must hit each subset X_i in $\mathcal{X}$, since otherwise, by definition, the attacker will simply attack X_i and gain more than B. Since the algorithm produces the solution L^g which hits every subset in $\mathcal{X}$, therefore the solution is a feasible solution in which the attacker's gain is no more than B. Since, a greedy solution to hitting set problem has an approximation ratio of $O(\log n)$ where n is the size of the universe of elements and since algorithm 10 is a greedy solution, therefore the solution of algorithm 10 approximates DefHard($\mathcal{A}, k, B$) with factor $O(\log |L|)$.

The algorithm enumerates all the "big subsets" (subsets of potential damage more than B) of k or less leaf nodes which takes $O(|L|^k)$ steps. Therefore, greedily picking the element takes $O(|L|^k \times |L|) = O(|L|^{k+1})$ steps. □

10.3.2 Algorithm for DefHard for large k

We describe a linear programming (LP) based rounding scheme for this problem when k is not a constant. We start with the following integer program $\mathcal{P}_I(G, k, B)$ to capture the DefHard(G, k, B) problem.

$$\begin{aligned} \min \sum_{v \in L} & x(v)P(v) \\ \text{subject to, } \sum_{v \in S} & x(v) \geq 1, \forall S : |S| \leq k, P(S) \geq B \\ & x(v) \in \{0, 1\} \end{aligned}$$

We let $\mathcal{P}_{LP}(\mathcal{A}, k, B)$ denote a relaxation of the above program, in which we have $x(v) \in [0, 1]$ for all $v \in L$.

```
Algorithm 11: Algorithm ApproxDH2
input  : $\mathcal{A}, k, B$
output: Leaf Node set $R$ that needs to be protected.
    • Initialize $R \leftarrow \phi$.
    • Solve program $\mathcal{P}_{LP}(\mathcal{A}, k, B)$ to obtain a fractional solution $\mathbf{x}$.
    • For each $v \in L$, add $v$ to $R$ with probability $x'(v) = \min\{1, (k + 2) \log n x(v)\}$.
```

The program $\mathcal{P}_{LP}(\mathcal{A}, k, B)$ has super-polynomially many constraints, when k is not a constant. So it cannot be solved directly in polynomial time. We describe an approximate solution based on the Ellipsoid method. We can assume from Lemma 36 that it is possible to compute an approximate solution $\mathbf{x}$ such that for all S with $|S| \leq k$ and $P(S) \leq \alpha B$, we have $\sum_{v \in S} x(v) \geq 1$, where $\alpha = \frac{1}{(1-\epsilon)(1-1/e)}$. The rounding step in the above algorithm converts this into a bicriteria approximation.

Lemma 35. *The set R computed by Algorithm* ApproxDH2 *is an $O(k \log n)$ approximation to the* DefHard$(\mathcal{A}, k, \alpha B)$ *problem, with high probability, where $\alpha = \frac{1}{(1-\epsilon)(1-1/e)}$.*

Proof. By Lemma 36, the solution $\mathbf{x}$ is a feasible solution to $\mathcal{P}_{LP}(\mathcal{A}, k, \alpha B)$. Therefore, for each set S with $|S| \leq k$ and $P(S) \geq \alpha B$, we have $\sum_{v \in S} x(v) \geq 1$. We consider one such set S. If there exists $v \in S$ with $x'(v) = 1$, it follows that $R \cap S \neq \phi$ with probability 1. Next, suppose $x'(v) < 1$ for all $v \in S$. Then, $\sum_{v \in S} x'(v) \geq (k+2)\log n$. By a standard Chernoff bound , it follows that $\Pr[|R \cap S| < c_1 k \log n] \leq \frac{1}{n^{k+2}}$. Since there are at most n^k such sets S, it follows that all these constraints are satisfied with probability at least $1 - 1/n$. □

Computing an approximate solution to $\mathcal{P}_{LP}$:

Since the LP $\mathcal{P}_{LP}$ has a super-polynomial number of constraints, we will use the ellipsoid method to solve it approximately. Informally, this method works in the following manner (see [107] for detailed discussion). Let Q be a polytope. We are interested in finding a feasible point[1] $x \in Q$.

An ellipsoid $\mathcal{E} = \mathcal{E}(c, C)$ with center c and positive semidefinite matrix C is defined as $\mathcal{E}(c, C) = \{x : (x-c)^T C^{-1}(x-c) \leq 1\}$. The ellipsoid algorithm constructs a sequence of ellipsoids $\mathcal{E}_i$, $i = 0, 1, \ldots, N$, each containing Q. Let e_i denote the center of ellipsoid $\mathcal{E}_i$. We assume we have an efficient algorithm $\mathcal{O}$ (referred to as a "separation oracle") that, given a point $x \in \mathbb{R}^n$, is able to decide whether $x \in Q$, and if $x \notin Q$, is able to find a constraint $d^T y \leq b$ (referred to as a "violated constraint") that is violated by x, but satisfied by all the points in Q. The ellipsoid algorithm considers the ellipsoids $\mathcal{E}_i$, for $i = 0, 1, \ldots$, and involves the following steps: (a) if $\mathcal{O}$ determines that $e_i \in Q$, we are done, and the algorithm returns e_i; (b) if $\mathcal{O}$ determines that $e_i \notin Q$, and returns a separating constraint $d^T x \leq b$, the algorithm constructs the next ellipsoid $\mathcal{E}_{i+1}$ to be the smallest ellipsoid containing $Q \cap \{x : d^T x \leq d^T e_i\}$. It can be shown that this algorithm involves a polynomial number of calls to the separation oracle $\mathcal{O}$.

Approximate Separation Oracle. Consider a candidate solution $\mathbf{x}$. In order to check its feasibility, we need to check if there exists a set S such that $|S| \leq k$, $P(S) \geq B$ and $\sum_{v \in S} x(v) < 1$. Our approximation oracle involves the following steps:

1. We formulate this as a submodular maximization problem, where the goal is to select a set S such that $P(S)$ is maximized, subject to the conditions $|S| \leq k$ and $\sum_{v \in S} x(v) \leq 1 - \delta$, where δ can be set to $\min_v x(v)$.

[1] The ellipsoid method is basically designed to find a *feasible* solution; however, as discussed in [107], if the goal is to find $x \in Q$ that minimizes a cost function $c^T x$, we can simply add a constraint $c^T x \leq \hat{z}$, where $\hat{z}$ is "close" to the optimum— it is easy to see that though we do not know the optimum, one can get a $(1+\epsilon)$-estimate by a binary search, for any $\epsilon > 0$.

2. Use the algorithm by Kulik et al. [64] to obtain a $(1-\epsilon)(1-1/e)$ approximation to this problem. Let S be the solution returned.

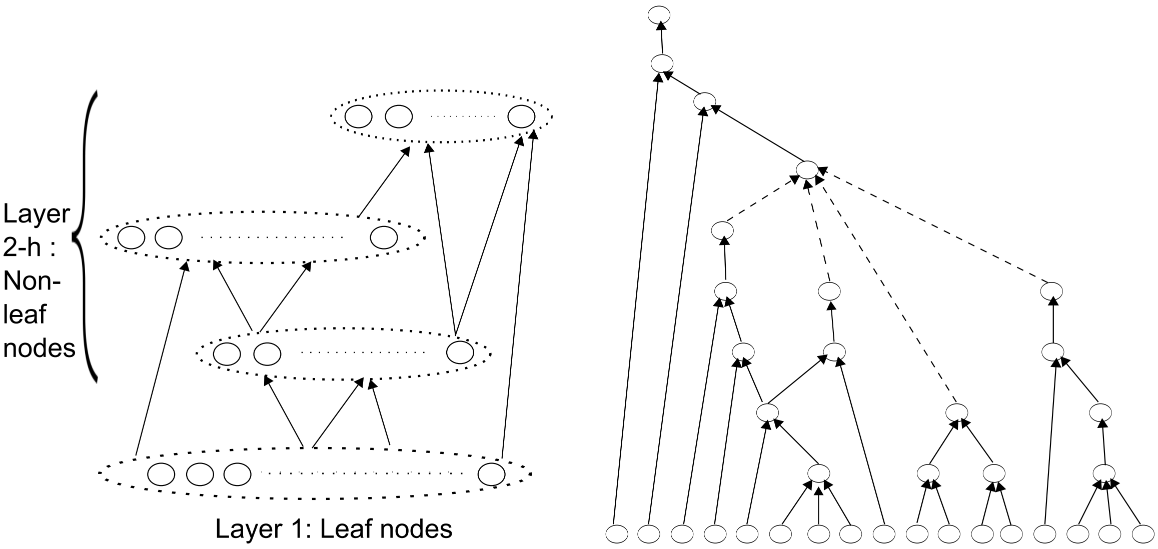
3. If $P(S) \geq B$, we return $\sum_{v \in S} x(v) \geq 1$ as the violated constraint

4. Else: we return $\mathbf{x}$ to be a feasible solution.

Lemma 36. *The ellipsoid method, implemented using the above approximate separation oracle gives a feasible solution* $\mathbf{x}$ *to* $\mathcal{P}_{LP}(\mathcal{A}, k, \alpha B)$*, where* $\alpha = \frac{1}{(1-\epsilon)(1-1/e)}$.

Proof. We first observe that the above separation oracle works "approximately". In the second case, it follows that for every S such that $|S| \leq k$ and $\sum_{v \in S} x(v) \leq 1-\delta$, we have $P(S) \leq \alpha B$. If this was not the case, and if there was a set S' with $|S'| \leq k$, $\sum_{v \in S'} x(v) \leq 1-\delta$ and $P(S') > \alpha B$, the algorithm of [64] would return a set S satisfying the two constraints, and with $P(S) \geq (1-\epsilon)(1-1/e)\alpha B = B$. Therefore, $\mathbf{x}$ is indeed a feasible solution to $\mathcal{P}_{LP}(\mathcal{A}, k, \alpha B)$. In the first case, if the algorithm finds that $P(S) \geq B$, it follows that $\sum_{v \in S} x(v) \leq 1-\delta < 1$ and $|S| \leq k$, so that $\sum_{v \in S} x(v) \geq 1$ is indeed not satisfied. □

10.4 Empirical Results

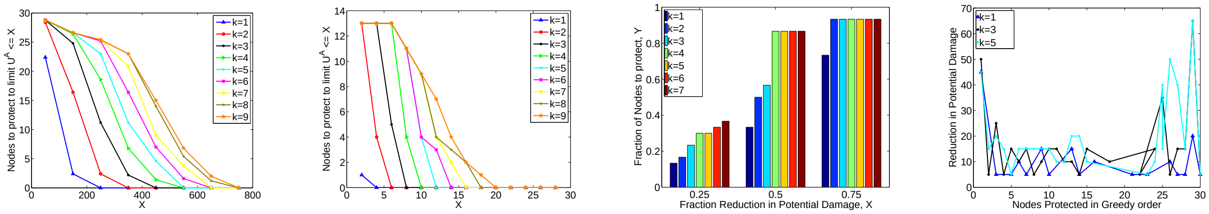


(a) Layered Attack Graph

(b) Real Attack Graph

Figure 10.3: Synthetic and real attack graph [30] in layered structure.

Due to the proprietary nature of network configurations and information about system vulnerabilities in enterprise networks, it is hard to obtain realistic attack graphs. We investigate



(a) OR-only graph: h:4,m:5

(b) Real Attack graph

(c) OR-only graph: h:4,m:5

(d) OR-only graph, h:4, m:5

Figure 10.4: Performance of GreedyDefHard for two attack graphs - an OR-only attack graph with 30 leaf and 50 non-leaf nodes, $h = 4$ and $m = 5$ (10.4a) and a real attack graph (10.4b). The plot in (10.4c) shows what fraction of leaf nodes to protect (Y-axis) to reduce the potential damage by a certain factor (X-axis) from the maximum potential damage. (10.4d) shows that the first few and the last few nodes picked in the greedy algorithm make the largest reduction in potential damage.

the attack graph studied in [30] and a simple layered random graph model for attack graphs, in order to understand the cost of hardening in such systems.

We study the attack graph from [30] (shown in Figure 10.3b), which is based on a small network of four machines in a LAN connected to the Internet through a firewall. The attack graph has 17 leaf and 18 non-leaf nodes. Most of the non-leaf nodes have AND functions while one has OR function. The graph is shown in a layered structure in figure 10.3b.

We also study a simple layered random graph model. The first layer corresponds to the leaf set L, while the remaining nodes are split randomly into h layers, which is a parameter. Each node v in every layer, other than the top-most, picks in-edges from $d_{in}(v)$ random nodes in the previous layer, where $d_{in}(v)$ is picked uniformly at random from $\{1, \dots, m\}$; the maximum indegree m is a parameter. We set $|L| = 30$ and $|V - L| = 50$ for all our experiments. The number of leaves attacked, k is varied from 1 to 9, the number of layers, h, is varied from 2 to 10 and the max indegree m is varied from 3 to 11. The node functions are selected to be OR/AND randomly, and the potential damage of each node is selected uniformly at random from $[0, 25]$.

We consider the defender's cost of hardening, the effect of specific node functions, the indegree and the number of layers. Our main results are summarized below.

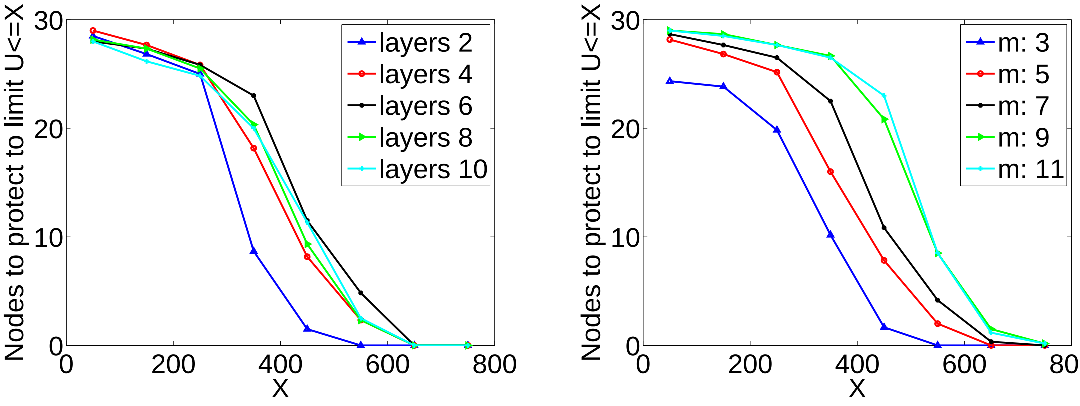
1. *Defender's cost of hardening*: Figure 10.4 shows the cost of hardening for different types of attack graphs on the y-axis, against the resulting potential damage on the x-axis, for different k values (corresponding to the number of attacked leaves). We observe that for both the real attack graph and the layered random graphs, the cost of hardening is quite high– typically more than half the leaves need to be secured in order to bring the potential damage below half of the maximum possible potential damage caused by attacking k leaves. It is interesting to observe that, for the same factor reduction in potential damage, the defender needs to

protect almost the same fraction of nodes irrespective of attacker's capability k. This is shown in figure 10.4c for an OR-only graph, where all the nodes have OR function. The same pattern holds for other graphs with varied layers and in-degrees. Further, there seem to be multiple regimes as the defender's potential damage is varied– initially, the potential damage decreases rapidly, for a small increase in the number of hardened leaves. Then it plateaus out, before finally decreasing rapidly again. This is also illustrated in figure 10.4d

2. Impact of node functions on cost of hardening: Potential damage is incurred less in attack graphs with OR node functions, compared to those with AND functions. This is not surprising because nodes with AND functions are less likely to be compromised when attacker's capability is limited.

3. Impact of the number of layers (h): As demonstrated in figure 10.5a for OR-only graphs, the number of layers have a weaker impact on the solution of DefHard. This is because of the fact that when node function is OR, a node is compromised whenever there is a path from an attacked leaf node to it; therefore layers do not matter much as long the number of nodes remains the same. The same has been observed for AND-only graphs as well.

4. Impact of indegree (m): As Figure 10.5b shows for OR-only graph, for higher indegree of nodes, the defender's cost for hardening increases. The opposite trend holds for AND-only graphs. This is because of the fact that, higher in-degree makes a node more likely to be compromised when the node function is OR; it makes compromise less likely when the node function is AND.



(a) Impact of layers

(b) Impact of in-degrees

Figure 10.5: Impact of layers and in-degrees in OR-only graphs: In each plot, attacker's capability is $k = 5$; in (10.5a) maximum in-degree,m is 5; and in (10.5b) number of layers is 4.

10.5 Conclusion

We study problems of estimating the inherent vulnerability of cyber-systems, and methods to harden them, within the attack graph framework. In contrast to prior work based on heuristics and evolutionary algorithms, we study the computational complexity of these problems, and develop efficient algorithms with provable approximation guarantees. Our results show that as the connectivity increases, attack graphs become increasingly challenging to harden. Our techniques can help system planners study tradeoffs between the hardening cost and the potential damage, thereby identifying the most critical nodes.

Part III

Conclusion

Chapter 11

Conclusion

In this thesis, we have studied problems in securing networks against cascading failures. First, we have studied decentralized and centralized strategies against epidemic outbreaks, particularly when the propagation is modeled with the SIS process. Then, we have studied the cascade of (i) stealthy and persistent attacks and (ii) vulnerability exploitations with attack graph models. We have shown analytic results that provide important insight into (i) the vulnerability of networked systems against cascading failures and, (ii) the outcome of strategic decisions made by self-interested autonomous agents, and how it compares to social optimal outcomes. We have proposed efficient algorithms to compute social optimum measures against cascading failures. We corroborate our analytic findings with extensive experimental studies made on large communication and infrastructure networks. Our work will help in the understanding of security of multi-agent systems and developing efficient security measures and mechanisms.

Chapter 12

Bibliography

[1] Cve: Commmon vulnerabilities and exposures database. https://cve.mitre.org/.

[2] SNAP: Stanford network analysis project.

[3] A. Adiga and A. Vullikanti. How robust is the core of a network? In *Proc. of ECMLP-KDD*, 2013.

[4] F. Allen and A. Babus. Networks in finance. Technical report, Wharton Financial Institution Ctr., 2008.

[5] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the Ninth Conference on Computer and Communications Security (CCS)*, pages 217–224, 2002.

[6] Roy M. Anderson and Robert M. May. *Infectious Diseases of Humans.* Oxford University Press, 1991.

[7] J. Aspnes, K.L. Chang, and A. Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. *J. Comput. Syst. Sci.*, 2006.

[8] J. Aspnes, N. Rustagi, and J. Saia. Worm versus alert: Who wins in a battle for control of a large-scale network? *OPODIS*, 2007.

[9] A.Vázquez, R. Pastor-Satorras, and A. Vespignani. Large-scale topological and dynamical properties of Internet. *Phys. Rev. E*, 2002.

[10] Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, and Manish Karir. A survey of botnet technology and defenses. In *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology*, pages 299–304. IEEE, 2009.

[11] Norman Bailey. *The Mathematical Theory of Infectious Diseases and its Applications.* Griffin, London, 1975.

[12] C.T. Bauch and D.J.D. Earn. Vaccination and the theory of game. *PNAS*, 2004.

[13] Leyla Bilge and Tudor Dumitras. Before we knew it: An empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 833–844, New York, NY, USA, 2012. ACM.

[14] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 2008.

[15] Larry Blume, David Easley, Jon Kleinberg, Robert Kleinberg, and Éva Tardos. Network formation in the presence of contagious risk. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 1–10. ACM, 2011.

[16] Kevin D Bowers, Marten Van Dijk, Robert Griffin, Ari Juels, Alina Oprea, Ronald L Rivest, and Nikos Triandopoulos. Defending against the unknown enemy: Applying flipit to system security. In *Decision and Game Theory for Security*, pages 248–263. Springer, 2012.

[17] Romulus Breban, Raffaele Vardavas, and Sally Blower. Theory versus data: how to calculate r 0? *PloS One*, 2(3):e282, 2007.

[18] Linda Briesemeister, Patric Lincoln, and Philip Porras. Epidemic profiles and defense of scale-free networks. *WORM*, Oct 2003.

[19] Andries E. Brouwer and Willem H. Haemers. *Spectra of graphs.* Springer, 2011.

[20] Hau Chan, Hanghang Tong, and Leman Akoglu. Make it or break it: Manipulating robustness in large networks. In *Proc. of SDM*, pages 325–333, 2014.

[21] P. Chen, Mary David, and David Kempe. Better vaccination strategies for better people. In *In Proc. of ACM conference on Electronic commerce(EC)*. ACM, 2010.

[22] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[23] F. R. K. Chung and L. Lu. The volume of the giant component of a random graph with given expected degrees. *SIAM J. Discrete Math.*, 20(2):395–411, 2006.

[24] Fan Rong K Chung and Linyuan Lu. *Complex graphs and networks.* Number 107. AMS Bookstore, 2006.

[25] EG Coffman Jr, Zihui Ge, Vishal Misra, and Don Towsley. Network resilience: Exploring cascading failures within bgp.

[26] Reuven Cohen, Shlomo Havlin, and Daniel ben Avraham. Efficient immunization strategies for computer networks and populations. *Physical Review Letters*, 91(24):247901, December 2003.

[27] Reuven Cohen, Shlomo Havlin, and Daniel ben Avraham. Efficient immunization strategies for computer networks and populations. *Physical Review Letters*, 91(24), 2003.

[28] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optimize float. *Management Science*, 23, 1977.

[29] David Dagon, Cliff Changchun Zou, and Wenke Lee. Modeling botnet propagation using time zones. In *NDSS*, volume 6, pages 2–13, 2006.

[30] R. Dewri, I. Ray, and N. Poolsappasit. Optimal security hardening on attack tree models of networks. *International Journal on Information Security*, 2012.

[31] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1), 2005.

[32] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, New York, NY, USA, 2001. ACM Press.

[33] S. Eidenbenz, V.S. Anil Kumar, and S. Zust. Equilibria in topology control games for ad hoc networks. *MONET*, 2006.

[34] Matthew Elliott, Benjamin Golub, and Matthew O Jackson. Financial networks and contagion.

[35] S. Eubank, H. Guclu, V. S. Anil Kumar, M. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:180–184, 2004.

[36] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5, 2011.

[37] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. Rumor cascades. In *ICWSM*, 2014.

[38] A. Galvani, T. Reluga, and G. Chapman. Long-standing influenza vaccination policy is in accord with individual self-interest but not with the utilitarian optimum. *Proceedings of the National Academy of Sciences*, 104(13):5692–5697, March 2007.

[39] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55 – 84, 2004.

[40] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proc. of INFOCOM*, 2005.

[41] Michele Garetto, Weibo Gong, and Don Towsley. Modeling malware spreading dynamics. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1869–1879. IEEE, 2003.

[42] 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*, 2001.

[43] M. Granovetter. Threshold models of collective behavior. *Am. Journal of Sociology*, 83(6):1420–1443, 1978.

[44] N.C. Grassly and C. Fraser. Mathematical models of infectious disease transmission. *Nature*, 2008.

[45] J. Grossklags, N. Christin, and J. Chuang. Secure or insure? a game-theoretic analysis of information security games. In *World Wide Web Conference (WWW)*, 2008.

[46] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *In Proc. of WWW*, 2004.

[47] Paul Hines, Karthikeyan Balasubramaniam, and Eduardo Cotilla Sanchez. Cascading failures in power grids. *Potentials, IEEE*, 28(5):24–30, 2009.

[48] V. Jacobson. Congestion avoidance and control. *SIGCOMM Comput. Commun. Rev.*, 18(4):314–329, August 1988.

[49] M. Jain, V. Conitzer, and M. Tambe. Security scheduling for real-world networks. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.

[50] S. Jha, O. Sheyner, and J. Wing. Two formal analysis of attack graphs. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, 2002.

[51] M. Kearns and L. Ortiz. Algorithms for interdependent security games. In *Advances in Neural Information Processing Systems*, 2004.

[52] D. Kempe. Structure and dynamics of information in networks. In *Lecture Notes*, 2011.

[53] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, 2003.

[54] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th KDD*, pages 137–146, 2003.

[55] M. H. R. Khouzani, Eitan Altman, and Saswati Sarkar. Optimal quarantining of wireless malware through reception gain control. *IEEE Trans. Automat. Contr.*, 57(1):49–61, 2012.

[56] M. H. R. Khouzani, Saswati Sarkar, and Eitan Altman. A dynamic game solution to malware attack. In *INFOCOM*, pages 2138–2146, 2011.

[57] M. H. R. Khouzani, Saswati Sarkar, and Eitan Altman. Saddle-point strategies in malware attack. *IEEE Journal on Selected Areas in Communications*, 30(1):31–43, 2012.

[58] Mert Korkali, Jason G Veneman, Brian F Tivnan, and Paul DH Hines. Reducing cascading failure risk by increasing infrastructure network interdependency. *arXiv preprint arXiv:1410.6836*, 2014.

[59] M. Krivelevich and B. Sudakov. The largest eigenvalue of sparse random graphs. *arXiv*, 2001.

[60] Chris J Kuhlman, VS Kumar, Madhav V Marathe, Henning S Mortveit, Samarth Swarup, Gaurav Tuli, SS Ravi, and Daniel J Rosenkrantz. A general-purpose graph dynamical system modeling framework. In *Proceedings of the Winter Simulation Conference*, pages 296–308. Winter Simulation Conference, 2011.

[61] Chris J. Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V. Marathe, and S. S. Ravi. Blocking simple and complex contagion by edge removal. In *Proc. of ICDM*, pages 399–408, 2013.

[62] Chris J Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V Marathe, and SS Ravi. Blocking simple and complex contagion by edge removal. In *2013 IEEE 13th International Conference on Data Mining*, pages 399–408. IEEE, 2013.

[63] Christopher James Kuhlman, VS Anil Kumar, Madhav V Marathe, Samarth Swarup, Gaurav Tuli, SS Ravi, and Daniel J Rosenkrantz. Inhibiting the diffusion of contagions in bi-threshold systems: Analytical and experimental results. In *AAAI Fall Symposium: Complex Adaptive Systems*, 2011.

[64] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proc. ACM Symposium on Discrete Algorithms (SODA)*, 2009.

[65] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. In *In Proc. of WWW*, 2003.

[66] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowedge Discover and Data Mining*, pages 611–617, New York, 2006.

[67] V. S. Anil Kumar, R. Rajaraman, Z. Sun, and R. Sundaram. Existence theorems and approximation algorithms for generalized network security games. In *Proc. of IEEE ICDCS*, 2010.

[68] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[69] Aron Laszka, Gabor Horvath, Mark Felegyhazi, and Levente Buttyn. Flipthem: Modeling targeted attacks with flipit for multiple resources. In *Decision and Game Theory for Security*, pages 175–194. 2014.

[70] M. Lelarge and J. Bolot. Economic incentives to increase security in the internet: The case for insurance. In *IEEE Infocom*, 2009.

[71] Andrea Lelli. The trojan. hydraq incident: Analysis of the aurora 0-day exploit, 2010.

[72] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie S. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pages 420–429, 2007.

[73] Jing Li, Daniel Blakeley, et al. The failure of 0. *Computational and mathematical methods in medicine*, 2011, 2011.

[74] Nilly Madar, Tomer Kalisky, Reuven Cohen, Daniel ben Avraham, and Shlomo Havlin. Immunization and epidemic dynamics in complex networks. *Eur. Phys. J. B*, 38(2):269–276, 2004.

[75] Madhav Marathe and Anil Kumar S Vullikanti. Computational epidemiology. *Communications of the ACM*, 56(7):88–96, 2013.

[76] Juil C Martin, Legand L Burge III, Joseph I Gill, Alicia N Washington, and Marcus Alfred. Modelling the spread of mobile malware. *International Journal of Computer Aided Engineering and Technology*, 2(1):3–14, 2009.

[77] J. Medlock and A. P. Galvani. Optimizing influenza vaccine distribution. *Science*, 325, 2009.

[78] P. Van Mieghem. *Spectral Graph Theory.* Cambridge University Press, 2011.

[79] P. Van Mieghem, D. Stevanovic, F. Fernando Kuipers, Cong Li, Ruud van de Bovenkamp, Daijie Liu, and Huijuan Wang. Decreasing the spectral radius of a graph by link removals. *IEEE Transactions on Networking*, 2011.

[80] Henning Mortveit and Christian Reidys. *An introduction to sequential dynamical systems*. Springer Science & Business Media, 2007.

[81] A. Nahir and A. Orda. Topology design and control: A game-theoretic perspective. In *INFOCOM*, 2009.

[82] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14, 1978.

[83] M. Newman. The structure and function of complex networks. *SIAM Review*, 45, 2003.

[84] M. E. J. Newman. Spread of epidemic disease on networks. *Physical Review E*, 66(1):016128, Jul 2002.

[85] M. E. J. Newman. The spread of epidemic disease on networks. *Physical Review Letters*, 66:016128, 2002. http://arxiv.org/PS_cache/cond-mat/pdf/0205/0205009.pdf.

[86] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[87] S. Noel and S. Jajodia. Managing attack graph complexity through visual hierarchical aggregation. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, VizSEC/DMSEC '04, pages 109–118, 2004.

[88] S. Noel, S. Jajodia, B. O'Berry, and M. Jacobs. Efficient minimum-cost network hardening via exploit dependency graphs. In *Proceedings of the 19th Annual Computer Security Applications Conference*, 2003.

[89] J. Omic, A. Orda, and P. Van Mieghem. Protecting against network infections a game theoretic perspective. In *INFOCOM*, 2009.

[90] T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2), 2009.

[91] Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013.

[92] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic dynamics in finite size scale-free networks. *Physical Review E*, 65:035108, 2002.

[93] C. Phillips and L. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 New Security Paradigms Workshop*, 1998.

[94] N. Poolsappasit, R. Dewri, and I. Ray. Dynamic security risk management using bayesian attack graphs. *Dependable and Secure Computing, IEEE Transactions on*, 9(1):61–74, Jan 2012.

[95] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. *Knowledge and Information Systems*, 2012.

[96] B. Aditya Prakash, Lada A. Adamic, Theodore J. Iwashyna, Hanghang Tong, and Christos Faloutsos. Fractional immunization in networks. In *SDM*, pages 659–667, 2013.

[97] Anirudh Ramachandran, Yogesh Mundada, Mukarram Bin Tariq, and Nick Feamster. Securing enterprise networks using traffic tainting. *Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GTCS-09-15*, 2009.

[98] C. R. Ramakrishnan and R. Sekar. Model-based analysis of configuration vulnerabilities. *Journal of Computer Security*, 10(1-2):189–209, 2002.

[99] I. Ray and N. Poolsappasit. Using attack trees to identify malicious attacks from authorized insiders. In *ESORICS*, 2005.

[100] T. Reluga. Game theory of social distancing in response to an epidemic. *PLOS Computational Biology*, 6(5), 2010. e1000793.

[101] R. W. Ritchey and P. Ammann. Using model checking to analyze network vulnerabilities. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, SP '00, 2000.

[102] Everett M. Rogers. *Diffusion of Innovations, 5th Edition.* Free Press, August 2003.

[103] Joshua J Romero. Blackouts illuminate india's power problems. *Spectrum, IEEE*, 49(10):11–12, 2012.

[104] T. Roughgarden. Stackelberg scheduling strategies. In *Proceedings of the annual ACM symposium on Theory of computing (STOC)*, 2001.

[105] S. Saha, A. Adiga, and Anil Kumar S. Vullikanti. Equilibria in epidemic containment games. Technical report, available at http://ndssl.vbi.vt.edu/specmkt/temp/ecgame_extended.pdf.

[106] B Schneier. Details of the rsa hack. *Schneier on Security, 30th Aug,{Online Resource} Available at: http://www. schneier. com/blog/archives/2011/08/details_of_the. html [Accessed 04/12/12]*, 2011.

[107] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1986.

[108] L. Segall. My hack stole your credit card, December 2015. [money.cnn.com; posted 08-December-2015].

[109] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[110] O. M. Sheyner. *Scenario Graphs and Attack Graphs.* PhD thesis, School of Computer Science, Carnegie Mellon University, 2004.

[111] Dong-Her Shih and Hsiu-Sen Chiang. E-mail viruses: how organizations can protect their e-mails. *Online Information Review*, 28(5):356–366, 2004.

[112] D. Shmoys and D. Williamson. *The Design of Approximation Algorithms.* Cambridge University Press, 2010.

[113] P. Slavik. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 1997.

[114] Tasuku Soma, Naonori Kakimura, Kazuhiro Inaba, and Ken-ichi Kawarabayashi. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In *Proceedings of the 31st International Conference on Machine Learning*, pages 351–359, 2014.

[115] H. Tong, B. Aditya Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *CIKM*, 2012.

[116] Hanghang Tong, B Aditya Prakash, Charalampos Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. On the vulnerability of large graphs. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 1091–1096. IEEE, 2010.

[117] Hanghang Tong, B. Aditya Prakash, Charalampos E. Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. On the vulnerability of large graphs. In *ICDM*, 2010.

[118] Gaurav Tuli, Chris J Kuhlman, Madhav V Marathe, S Ravi, and Daniel J Rosenkrantz. Blocking complex contagions using community structure. In *Proc. Workshop Multiagent Interaction Netw.* Citeseer, 2012.

[119] Marten Van Dijk, Ari Juels, Alina Oprea, and Ronald L Rivest. Flipit: The game of "stealthy takeover". *Journal of Cryptology*, 26(4):655–713, 2013.

[120] L. Wang, S. Noel, and S. Jajodia. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18):3812–3824, November 2006.

[121] Lingyu Wang, Sushil Jajodia, Anoop Singhal, and Steven Noel. k-zero day safety: Measuring the security risk of networks against unknown attacks. In *Computer Security–ESORICS 2010*, pages 573–587. Springer, 2010.

[122] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *Symposium on Reliable Distributed Systems*, pages 25–34, Los Alamitos, CA, 2003. IEEE Computer Society Press.

[123] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *SRDS*, pages 25–34, 2003.

[124] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proc. of STOC*, 2012.