Chapter 4

Controlling negative diffusion

In Chapter 2 and Chapter 3, we have studied how to enable positive diffusion under both organic and adversarial dynamics. In this chapter, we switch gear to controlling negative diffusion.

Over the recent decades, there has been an explosive growth in the use of personal digital devices of various kinds. This has, unfortunately, been accompanied by significant increase in worm attacks. While, effective anti-virus software and patches are readily available, the average user is very independent and does not often loading these latest patches due to software cost and other efforts involved. Also if enough other nodes in the network are secured, the likelihood of a specific device getting infected would go down, leading to a natural game theoretic scenario. Aspnes et al [15] introduced an innovative game for modeling the containment of the spread of viruses and worms (security breaches) in a network. In this model, nodes choose to install anti-virus software or not on an individual basis while the viruses or worms start from a node chosen uniformly at random and spread along paths consisting of insecure nodes. They showed the surprising result that a pure Nash Equilibrium always exists when all nodes have identical installation costs and identical infection costs.

In this chapter we present a substantial generalization of the model of [15] that allows for arbitrary security and infection costs, and arbitrary distributions for the starting point of the attack. More significantly, our model GNS(d) incorporates a network locality parameter d which represents a hop-limit on the spread of infection as accounted for in the strategic decisions, due to either the intrinsic nature of the infection or the extent of neighborhood information that is available to a node.

We determine that the network locality parameter plays a key role in the existence of pure Nash equilibria (NE): local ($d = 1$) and global games ($d = \infty$) have pure NE, while for GNS(d) games with $1 < d < \infty$, pure NE may not exist, and in fact, it is NP-complete to determine whether a given instance has a pure NE. For local and global games, we also characterize the price of anarchy in terms of the maximum degree and vertex expansion of the contact network; these suggest natural heuristics to aid a network planner in enforcing efficient equilibria.

We design a general LP-based framework for approximating the NP-complete problem of finding a socially optimal configuration in our game. Our framework yields a $2d$-approximation for general GNS(d) games, and an $O(\log n)$-approximation for the global model where n is the number of network nodes; the latter result improves on the approximation bound of $O(\log^{1.5} n)$ of [15] achieved for a special case of our global model.

We study the characteristics of NE and the quality of our approximations empirically in two distinct classes of graphs: random geometric graphs and power law graphs. We find that in local and global games on these real-world networks, best response dynamics converge in linear or sub-linear time and have costs comparable to the social optimum. Finally, we study the performance of our approximation algorithms, and find that the approximation guarantees with respect to social cost are much better in practice than our theoretical bounds.

4.1 Related Work

Non-cooperative game theory has been used in analyzing a number of problems in traffic and communication networks, e.g., routing [117], topology control and network formation [59, 107] and security [71, 112]. The basic questions of interest have usually been about the existence and the structure of Nash equilibria and the price of anarchy, which is the worst case cost of a Nash equilibrium to the social optimum, as defined formally later. See [111] for a good introduction on the use of game theoretic techniques for networking applications.

Several formulations have been proposed for analyzing network security problems and the spread of epidemics in networks [15, 16, 35, 67, 71, 94, 126]. This thesis directly builds on the formulation of Aspnes et al. [15], who model the risk of infection

for an insecure node v as the probability that the initial infection, which is assumed to originate at a node chosen uniformly at random, starts in the same component as v in the subgraph induced by v and the other insecure nodes. They show the surprising result that pure Nash equilibria always exist in such games. They also establish a high price of anarchy and give an $O(\log^{1.5} n)$ approximation algorithm for computing the social optimum, where n is the number of nodes in the network. Their approximation algorithm uses an $O(\sqrt{\log n})$-approximation for the sparsest cut problem [14], which is based on a semidefinite programming relaxation of the problem. In this thesis, we are able to give a much simpler LP-based approximation algorithm using the vertex multicut problem, which improves the approximation ratio to $O(\log n)$ and also applies to a more general model. Another direction of work is based on SIS models for the worm spread, e.g., the n-intertwined model [112]. In this model, nodes are in two states - susceptible or infected. Each infected node spreads the infection to its neighbors with some probability. Another closely related class of models is that of Interdependent Security games (IDS) [81], which is similar to our model for the special case of $d = 1$. One crucial technical difference between the two models, which leads to two different games, is the assumption about the initial infection: in IDS, it is assumed to originate independently at different nodes, while in our GNS(1) model, we assume an initial location is selected according to a given probability distribution.

Our formulation of generalized network security games is largely motivated by mechanisms to protect communication networks. Some of our model and results, especially the lower bound results, however, also apply equally well to the spread of diseases and the protection of communities through vaccinations. The pure Nash equilibria correspond to stable points in the space of vaccination decisions made by individuals, and our approximation algorithms yield public policies for vaccination that well-approximate the social welfare. There is considerable work in epidemiology, both from a game-theoretic perspective, as well as on the analysis of disease spreads through SIR and SIS models [32, 31, 85, 30, 33]. The game-theoretic models adopted in these studies, however, do not consider the impact of the underlying contact network. Furthermore, there is little work on quantifying the effect of locality (in disease spread or in information availability).

4.2 Model and Definitions

In this section, we present our game-theoretic model for network security.

Contact Graph. Let V denote the set of users/devices (henceforth, referred to as *nodes*), each of which is assumed to be an autonomous player. Let G denote the underlying contact graph over the node set V; an edge $(u, v) \in G$ indicates that nodes u and v are directly connected, so that if node u is infected by a worm it can potentially spread to node v. Let $N(v)$ denote the set of neighbors of v in G. We will frequently work with certain subgraphs of G, for which we introduce the following notation. For any undirected graph H and subset S of vertices of H, we let $H[S]$ denote the subgraph of H induced by the vertices in S.

Strategies. The strategy for each node v is the decision of whether to install an anti-virus software or not; we use a variable $a_v \in [0, 1]$ to denote the probability of securing the device. In this paper, we focus on *pure* strategies, i.e., $a_v \in \{0, 1\}$. Let $\vec{a}$ denote the strategy vector of all nodes. Following [15], the *attack graph*, $G_{\vec{a}}$, is the subgraph of the contact graph induced by the set of insecure nodes according to $\vec{a}$. For notational convenience, let $\vec{a}[v/x]$ be the strategy vector obtained by replacing a_v by x in the vector $\vec{a}$.

Infection model. We assume that the infection is initiated at a node chosen from V according to an arbitrary probability distribution. Let w_v denote the probability that node v is chosen as the initial infection point; for convenience, we introduce the notation $w(S)$ to denote the sum of w_v over all v in S. We parameterize the infection model by d, the maximum number of hops over which the probability of infection spread is taken into account in the decision making. Thus, for a given contact graph G and strategy vector $\vec{a}$, an infection originating at node v infects node u if and only if u is within d hops of v in $G_{\vec{a}}$. Since G is fixed and d is clear from the context, denote by $S_v(\vec{a})$ the set of nodes that are within d hops of v in $G_{\vec{a}[v/0]}$. For a given strategy vector $\vec{a}$, therefore, the probability that node v gets attacked in this model (denoted by $p_v(\vec{a})$) is $w(S_v(\vec{a}))$.

Generalized Network Security Game GNS(d). We now present our model for a generalized network security game GNS(d), parameterized by the hop-limit d in the infection model. The game GNS(d) is specified by a contact graph G, initial infection probability distribution w, and two costs per network node. Let C_v denote the security

cost (installing an anti-virus software) of user v; we assume the software is fool-proof so that secure nodes do not get attacked. Let L_v denote the infection cost of user v (recovering from a worm attack in case an insecure node v gets attacked). Then, the cost to node v is defined as

$$\text{cost}_v(\bar{a}) = a_v C_v + (1 - a_v) L_v \cdot p_v(\bar{a}).$$

A pure Nash equilibrium (henceforth, pure NE) is a strategy vector $\vec{a}$ such that no node v has any incentive to switch his strategy, if all other nodes' strategies are fixed. $\vec{a}$ is a Nash equilibrium if $\text{cost}_v(\vec{a}[v/x]) \geq \text{cost}_v(\vec{a})$ for $x \in \{0,1\}$. Therefore, a pure NE is a natural configuration to aim for in a non-cooperative game. It is easy to verify that the following characterization of a pure NE (shown in [15] for the special case where G is the complete graph) holds.

Lemma 28. *For $v \in V$, let $t_v = C_v/L_v$. A strategy vector $\vec{a} \in \{0,1\}^n$ is a pure NE if the following conditions hold: (i) for all v such that $a_v = 0$, $w(S_v(\vec{a})) \leq t_v$, and (ii) for all v such that $a_v = 1$, $w(S_v(\vec{a}[v/0])) > t_v$.*

Social cost. The total social cost of a strategy profile is the sum of the individual costs, which is $\text{cost}(\bar{a}) = \sum_{v=1}^{n} \text{cost}_v(\bar{a})$. A socially optimum strategy is a vector $\vec{a}$ that minimizes this cost - this is not necessarily (and is not usually) a pure NE. Therefore, the cost of a pure NE relative to the social cost is an important measure; the maximum such ratio (i.e., over all possible pure NE) is also known as the *price of anarchy* [88].

For convenience, Table 4.1 summarizes our notations.

4.3 Nash equilibria

4.3.1 The local infection model: $d = 1$

For the local infection model, we show that a pure NE always exists. Our proof is by a reduction to a result of Borodin et al. [41] on existence of subgraphs with restricted degree sequences; their result is based on a potential function argument.

Theorem 29. *Every* GNS(1) *instance has a pure NE.*

Proof. We first define two functions $a : V \to \mathbb{R}$ and $b : V \to \mathbb{R}$. For each $v \in V$, $a(v) = w(N(v)) - \frac{C_v}{L_v} + w(v)$ and $b(v) = \frac{C_v}{L_v} - w(v)$. We argue next, using a generalization

Table 4.1: A list of notations.

Notations	Explanation
G	Contact graph.
$G[S]$	Subgraph of G induced by the vertices in S.
C_v	Security cost for node v
L_v	Infection cost for node v
$\vec{a}$	Strategy vector of nodes.
$G_{\vec{a}}$	Attack graph, i.e. the subgraph of the contact graph induced by the set of insecure nodes according to $\vec{a}$.
$\vec{a}[v/x]$	Strategy vector obtained by replacing a_v by x in the vector $\vec{a}$.
$S_v(\vec{a})$	Set of nodes that are within d hops of v in $G_{\vec{a}[v/0]}$.
w_v	Probability that node v is chosen as the initial infection point.
$w(S)$	Sum of w_v over all v in S.
$\text{cost}_v(\vec{a})$	Cost to node v given strategy vector $\vec{a}$.
$\text{GNS}(d)$	Generalized network security game parameterized by the disease hop limit d.

of an argument due to [41], that there exists a partition $V = A \cup B$ such that for each $v \in A$, we have $w(A \cap N(v)) \leq a(v)$ and for each $v \in B$, we have $w(B \cap N(v)) \leq b(v)$. Consider the following function that defines a potential for each partition (A, B).

$$\begin{aligned} R(A,B) = \quad & \textstyle\sum_{v \in A} w(v) \left(w(A \cap N(v)) - 2a(v)\right) \\ & + \textstyle\sum_{v \in B} w(v) \left(w(B \cap N(v)) - 2b(v)\right) \end{aligned}$$

Among all the partitions, we take a partition (A^*, B^*) minimizing R and assert that (A^*, B^*) is the partition we need. Suppose that a vertex x belongs to A^*, and $w(A^* \cap N(x)) > a(x)$. Now we move x from A^* to B^* to obtain the partition $(A' = A^* \setminus \{x\}, B' = B^* \cup \{x\})$. Because $a(x) + b(x) \geq w(N(x))$, we have $w(N(x) \cap B^*) \leq b(x)$. It is easy to verify that $R(A^*, B^*) - R(A', B')$ equals $w(x) \left(w(N(x) \cap A^*) - 2a(x)\right) + w(x)w(N(x) \cap A^*) - w(x) \left(w(N(x) \cap B^*) - 2b(x)\right) - w(x)w(N(x) \cap B^*) = 2w(x) \left(w(N(x) \cap A^*) - a(x)\right) - 2w(x) \left(w(N(x) \cap B^*) - b(x)\right) > 0$. This means $R(A^*, B^*) > R(A', B')$, which is a contradiction. A similar inequality follows if there is a vertex $x \in B^*$ with $w(B^* \cap N(x)) > b(x)$. Therefore, such a vertex x doesn't exist implying that (A^*, B^*) is the desired partition.

Given such a partition (A, B), we establish the existence of pure NE. Let $\vec{a}$ be a strategy vector with $a_v = 1$ for all $v \in A$ and $a_v = 0$ for all $v \in B$; i.e., A denotes the set of secure nodes. Then, we argue that $\vec{a}$ is indeed a pure NE. First consider the case where $v \in A$. Then v is secure and pays cost C_v. If v changes strategy, its expected infection cost is $L_v \left(w(N(v) \cap B) + w(v)\right)$. Since $v \in A$, we have $w(N(v) \cap A) \leq a(v) = w(N(v)) - C_v/L_v + w(v)$. Therefore, $C_v \leq L_v \left(w(N(v) \cap B) + w(v)\right)$, i.e. v won't change its strategy. Next consider $v \in B$. Then v is not secure and its expected infection cost is $L_v \left(w(N(v) \cap B) + w(v)\right)$. If v changes strategy, its cost is C_v. Since $v \in B$, we have $w(N(v) \cap B) \leq b(v) = C_v/L_v - w(v)$. Therefore, $L_v \left(w(N(v) \cap B) + w(v)\right) \leq C_v$, i.e. v won't change its strategy. Thus it follows that $\vec{a}$ is a Nash equilibrium.

□

When the security and infection costs are uniform, we show that for the case of $d = 1$, the maximum ratio of the cost of a pure NE to the social optimum is bounded by the maximum degree.

Lemma 30. *When security and infection costs are uniform, and $w_v = 1/n \ \forall v$, the price of anarchy in* GNS*(1) is at most $\Delta + 1$, where Δ is the maximum degree of the contact graph.*

Proof. Let C and L denote the security and infection costs, respectively. Suppose $C > L(\Delta + 1)/n$. Then no node is secured in any pure NE and therefore, the cost of

any pure NE is at most $L(\Delta+1)$. In the optimum strategy, each node has a cost of C if it is secured, or at least L/n otherwise. Therefore the optimal cost is at least L, and the lemma follows in this case.

Next, consider the case $C \leq L(\Delta+1)/n$. In any pure NE, any node has cost at most C, and therefore the cost of a pure NE is at most Cn. If $C \leq L/n$, the optimum cost is also Cn, and therefore, we assume $C \geq L/n$. In an optimum solution, each node has cost at least L/n, and therefore, the optimal cost is at least L. Therefore, the price of anarchy in this case is at most $\Delta+1$. □

4.3.2 The global infection model: $d = \infty$

In this section, we consider the global model ($d = \infty$); thus, any node v is capable of infecting any other node u as long there is a path of insecure nodes between v and u in the contact graph G. In this special case, our model is a generalization of the model of [15] in that we allow different security costs, infection costs, and initial infection probabilities.

Theorem 31. *Every* GNS(∞) *instance has a pure NE.*

Proof. Let $t_v = C_v/L_v$; we refer to t_v as the threshold for v. We relabel the n nodes so that $t_1 \geq t_2 \geq \cdots \geq t_n$, where we break ties arbitrarily. Given a strategy vector $\vec{a}$, we say that a secure node v is *happy* if $w(S_v(\vec{a}[v/0])) > t_v$, and *unhappy* otherwise. Similarly, an insecure node v is *happy* if $w(S_v(\vec{a})) \leq t_v$, and *unhappy* otherwise. Recall that when $d = \infty$, $S_v(\vec{a})$ is the set of nodes that can reach v in $G_{\vec{a}}$.

Consider the following potential function.

$$\hat{\Phi}(\vec{a}) = (\Phi_1(\vec{a}), \Phi_2(\vec{a}), \ldots, \Phi_n(\vec{a}))$$

where $\Phi_v(\vec{a})$ is 0 if v is secure, -1 if v is insecure and happy, and 1 otherwise. We next show this potential always lexicographically decreases. There are two cases:

1. Some node v switches from being an insecure unhappy node to being a secure happy node, changing the strategy vector from $\vec{a}$ to $\vec{b}$. In this case $w(S_v(\vec{a})) > t_v$. Since the set of secure nodes in $\vec{b}$ is a superset of the set of secure nodes in $\vec{a}$, it follows that for any node u, $w(S_u(\vec{b})) \leq w(S_u(\vec{a}))$; it thus follows that no insecure happy node in $\vec{a}$ can become unhappy in $\vec{b}$. Therefore, the vth component of the potential decreases by 1, while none of the other components increases.

2. Some node v switches from being secure to not being secure, changing the strategy vector from $\vec{a}$ to $\vec{b}$. In this case, $w(S_v(\vec{b})) \leq t_v$. We thus have the vth component of

the potential changing from 0 to -1. Consider any node $u \neq v$. If u is secure, then the uth component of the potential is unchanged. Otherwise, consider two cases. If v and u are in different connected components, then $w(S_u(\vec{b})) = w(S_u(\vec{a}))$, implying that the uth component of the potential is unchanged. If v and u are in the same connected component, then $w(S_u(\vec{b})) = w(S_v(\vec{b}))$; thus, if u is happy in $\vec{a}$ but unhappy in $\vec{b}$, then it must be the case that $t_u < t_v$, implying that $u > v$. Thus, the only components of the potential that can increase are the components greater than v, implying that the potential decreases lexicographically.

Since the value of each column in the potential vector is between -1 and 1, and this potential vector lexicographically decreases, we conclude that this process converges to a pure Nash equilibrium (in fact, in at most 3^n steps). □

Even when the security and infection costs are uniform, [15] showed that the price of anarchy is $\Omega(n)$. We give a more precise characterization in terms of the vertex expansion of the contact graph. For any graph H over vertex set V, the vertex expansion $\alpha(H)$ is defined as the largest number c such that for any subset V' of the vertices such that $|V'| \leq |V|/2$, the set of vertices in $V \setminus V'$ that are adjacent to a vertex in V' is at least $c|V'|$.

Lemma 32. *When security and infection costs are uniform, and $w_v = 1/n$ $\forall v$, the price of anarchy in any* GNS(∞) *game is $O(1/\alpha(G))$.*

Proof. First we calculate the lower bound for social optimum. Let $\vec{a}$ be the strategy vector of a social optimum, and $S_1, S_2, \ldots, S_m$ denote the connected components in $G_{\vec{a}}$. Without loss of generality, we can assume $|S_1| \leq |S_2| \leq \cdots \leq |S_m|$. We consider the following 3 cases:

1. $\sum_i |S_i| < n/2$, where n is the total number of nodes in G. In this case more than half of the nodes are secure. Thus, social optimal cost is at least $Cn/2$.

2. $\sum_i |S_i| \geq n/2$ and $|S_m| \geq n/4$. Then social optimal cost is at least $\sum_{v \in S_m} \text{cost}_v(\vec{a}) \geq \frac{n}{4} L \frac{n/4}{n} = Ln/16$.

3. $\sum_i |S_i| \geq n/2$ and $|S_m| < n/4$. Then there must be a j such that $\sum_{i \leq j} |S_i| \geq n/4$. Let $S = \cup_{i \leq j} S_i$. Then the number of neighbors of set S in G is at least $\alpha(G)|S| \geq \alpha(G)n/4$. This implies social optimal cost is at least $C\alpha(G)n/4$.

Therefore, the lower bound for social optimum is $\min\{Cn/2, Ln/16, C\alpha(G)n/4\}$.

Next we calculate the upper bound for NE cost. Let $\vec{a}$ be the strategy vector of a NE. Again, let $S_1, S_2, \ldots, S_m$ denote the connected components in $G_{\vec{a}}$. $|S_1| \leq |S_2| \leq \cdots \leq |S_m|$. We consider the following 2 cases.

1. $L \leq C$. In this case no one is going to be secure in NE, which implies its cost is nL. The ratio between NE and the social optimum is no more than $\max\{2, 16, 4/\alpha(G)\}$.

2. $L > C$. The cost of NE is no more than $\sum_i L|S_i|^2/n + Cn$. Because this is a NE, for those who choose to be insecure, $L|S_i|/2 \leq C$. Therefore, we have $\sum_i L|S_i|^2/n + Cn \leq \sum_i C|S_i| + Cn \leq 2Cn$. The ratio between NE and the social optimum is no more than $\max\{4, 32, 8/\alpha(G)\}$.

Putting these 2 cases together completes the proof of this lemma. □

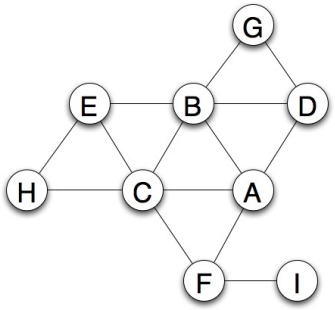
4.3.3 The d-neighborhood infection model: $d > 1$

Having established the existence of a pure NE for every instance of the generalized network security game in both the local and the global models, a natural question is whether pure NE exist for the entire spectrum of d in between these two extremes. In this section, we show that for any $1 < d < \infty$, there exist instances of GNS(d) for which there are no pure NE. Furthermore, it is NP-complete to determine whether a pure NE exists for a given instance. We first present the non-existence result which also provides the basis for the NP-hardness reduction.

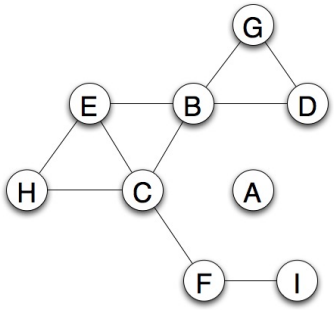
Lemma 33. *For any fixed d, $1 < d < \infty$, there exists an instance of* GNS(d) *in which no pure NE exists.*

Proof. We first consider the case $d = 2$. Consider the instance defined by the contact graph in Figure 4.1a. $w_v = 1/n$ for all node v. We set the infection cost to be identical, say L, for all nodes. For nodes D through I, we set the security cost to be high enough so that in any equilibrium they are all insecure. That leaves nodes A, B, and C, for whom we set the security cost such that $9C_v/L = 7 + \epsilon$ for v in {A,B,C}; thus, in any pure NE $\vec{a}$, node v in {A, B, C} is secure if and only if $|S_v(\vec{a}[v/0])| \geq 7 + \epsilon$. We now consider four cases. If all of A, B, and C are insecure in $\vec{a}$, then we do not have a pure NE since $|S_v(\vec{a}[v/0])| = 9$ for each v in {A, B, C}. If exactly one of A, B, or C – say A – is secure, as shown in Figure 4.1b, then B won't change its strategy since $|S_B(\vec{a})| = 7$, but C will change its strategy since $|S_C(\vec{a})| = 8$ (Notice C can reach I, but B cannot). If exactly two of A, B, C – say A and B – are secure, as shown in Figure 4.1c, then

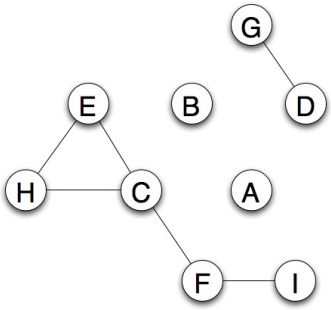
B will change its strategy since $|S_B(\vec{a}[B/0])| = 7$. Finally, if all three are secure, then none of A, B, or C will stick to its current strategy since $|S_v(\vec{a}[v/0])| = 5$ for each v in {A,B,C}. We have thus established that there is no pure NE in the instance of Figure 4.1a. It is easy to extend the above non-existence proof to larger d by replacing selected edges in the instance of Figure 4.1a by multi-hop paths. □



(a) An instance of a contact graph that has no pure NE.



(b) Residual graph when A chooses to secure itself.



(c) Residual graph when A and B choose to secure themselves

Figure 4.1: No pure NE example with nonuniform security costs and infection costs.

In the above non-existence proof, nodes have different security costs and infection costs. We can extend the proof to the case of uniform security costs and infection costs by inserting additional nodes in the proximity of those nodes in the above instance that have lower security costs, as shown in the following lemma.

Lemma 34. *For any fixed d, $1 < d < \infty$, there exists an instance of* GNS(d) *in which*

no pure NE exists.

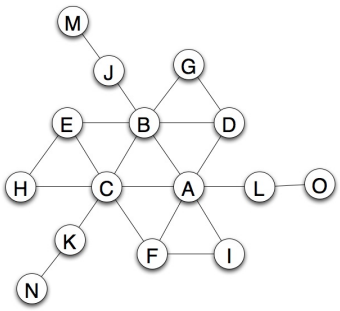
Proof. We first consider the case $d = 2$. Consider the instance defined by the contact graph in Figure 4.2a. $w_v = 1/n$ for all node v. We set the infection cost to be L and security cost to be $C = (10 + \epsilon)L/15$ for all nodes. Thus, in any pure NE $\vec{a}$, node v is secure if and only if $|S_v(\vec{a}[v/0])| \geq 10+\epsilon$. Therefore, nodes D through O are all insecure in any pure NE. We now consider four cases. If all of A, B, and C are insecure in $\vec{a}$, then we do not have a pure NE since $|S_v(\vec{a}[v/0])| = 13$ for each v in {A, B, C}. If exactly one of A, B, or C – say A – is secure, as shown in Figure 4.2b, then B won't change its strategy since $|S_B(\vec{a})| = 10$, but C will change its strategy since $|S_C(\vec{a})| = 11$. If exactly two of A, B, C – say A and B – are secure, as shown in Figure 4.2c, then B will change its strategy since $|S_B(\vec{a}[B/0])| = 10$. Finally, if all three are secure, then none of A, B, or C will stick to its current strategy since $|S_v(\vec{a}[v/0])| = 7$ for each v in {A, B, C}. We have thus established that there is no pure NE in the instance of Figure 4.2a. It is also easy to extend the above non-existence proof to larger d by replacing selected edges in the instance of Figure 4.2a by multi-hop paths. □

We next show that it is, in fact, NP-complete to determine whether a given instance of the generalized network security game with $1 < d < \infty$ has a pure NE. It is easy to argue that the problem is in NP since one can efficiently verify whether a given strategy vector $\vec{a}$ is a pure NE. In the remainder of this section, we focus on the hardness reduction.

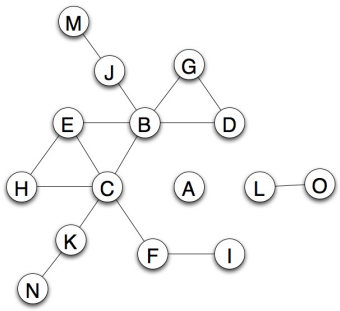
Our starting point is the non-existence instance defined in the Lemma 33. We observe that if the security cost of exactly one of the three nodes in {G, H, I}, say G, is reduced so that G always secures itself, then we do have a pure NE in which C secures itself, while A and B are insecure. Thus, if we can control the decision of G through an external input, then we can use the above instance as a gadget which has the property: it has a pure NE if and only if G is secure. We now show how to use this gadget to obtain an NP-hardness reduction.

Theorem 35. *The problem of determining if a* GNS(d) *instance,* $1 < d < \infty$*, has a pure NE is NP-complete.*

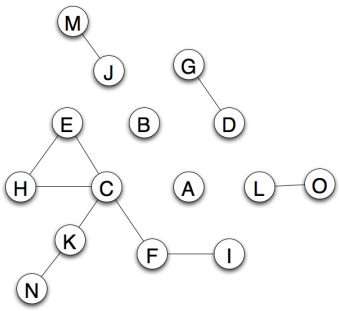
Proof. We reduce 3SAT problem to a GNS(2) instance, and show that a given formula ϕ is satisfiable if and only if the corresponding game has a pure NE. The reduction is shown in Figure 4.3. For each variable X in the formula, we create two nodes in the contact graph, X and $\bar{X}$, which are connected to each other. For each literal l in the



(a) An instance of a contact graph that has no pure NE.



(b) Residual graph when A chooses to secure itself.



(c) Residual graph when A and B choose to secure themselves.

Figure 4.2: No pure NE example with uniform security costs and infection costs.

formula, we create a node, and connect it with corresponding variable. For each clause C, we create a gadget, treat node G as clause node, and connect it to its 3 literal nodes. The costs for gadget nodes are as before. The costs of literal nodes are set such that their "threshold" – the number of insecure nodes that can tolerate without securing themselves – is 1. And the threshold for X is set to be $a + 1$ where a is the number of adjacent literal nodes; the threshold for $\bar{X}$ is set to be $b + 1$ where b is the number of adjacent literal nodes. We add padding nodes between edges $(X, \bar{X})$, (X, I), $(\bar{X}, I)$, and (C, I). We set their security costs to be 0, so they always wish to be secure.

We first show if ϕ is satisfiable, then there is a pure NE in this game. For variable node X, if its assignment is true, then make it secure. For literal node I, if its assignment is false, then make it secure. If a clause is true, then make it secure. All the other nodes are insecure. We now argue that the defined strategy vector is a pure

NE. If a variable node X is secure, then all the literal nodes connected to it are not secure, $\bar{X}$ is not secure, while all the literal nodes connected to $\bar{X}$ are secure. Since the formula is satisfiable, all the clause nodes are secure. It is clear that $\bar{X}$ is happy, since its threshold is $b+1$ and X is secure. Similarly X is happy since if it were to be insecure, it will be in a component with size $a+2$ which is bigger than its threshold. All the literal nodes connected to X are happy, because for each of them, the only two adjacent nodes are secure. And all the literal nodes connected to $\bar{X}$ are happy, because if any of them does not secure itself, it will be in a component with size 2, which is bigger than its threshold. All the clause nodes are happy because the formula is satisfiable, at least one of its literal is true, which means at least one of its literal nodes is insecure, hence this clause node has to secure itself because its threshold is 6. And within each gadget, we can make node C to secure itself (together with the nodes D, E, and F) to make all the nodes in the gadget happy. We thus have a pure NE in the game instance.

Next, we argue if the game has a pure NE, then the formula is satisfiable. Suppose we have a pure NE strategy vector $\vec{a}$. For each variable node X, if X is secure, we assign X to be true for the SAT formula; and false otherwise. We know that in any pure NE, the clause node in each gadget has to be secure. Furthermore, exactly only of X and $\bar{X}$ is secure. If X is secure, then $\bar{X}$ and all the literal nodes connected to X have to be insecure, while all the literal nodes connected to $\bar{X}$ have to be secure. Since all the clause nodes are happy, at least one of its literal nodes is not secure, implying that in each clause at least one of the literals is true. This establishes that the formula is satisfiable.

In sum, the formula is satisfiable if and only if the security game has Nash equilibrium. It is easy to see that the above reduction can be carried out in polynomial time, thus yielding the NP-hardness of the problem. □

4.4 Optimizing social welfare: NP-completeness and approximation algorithms

4.4.1 NP-completeness of computing the social optimum

We show that computing the social optimum is NP-complete in GNS(d) games for all d. The result for $d = \infty$ follows from Aspnes et al. [15], even for the special case where all security costs, infection costs, and initial infection probabilities are uniform. We now establish NP-completeness for all $d > 0$.

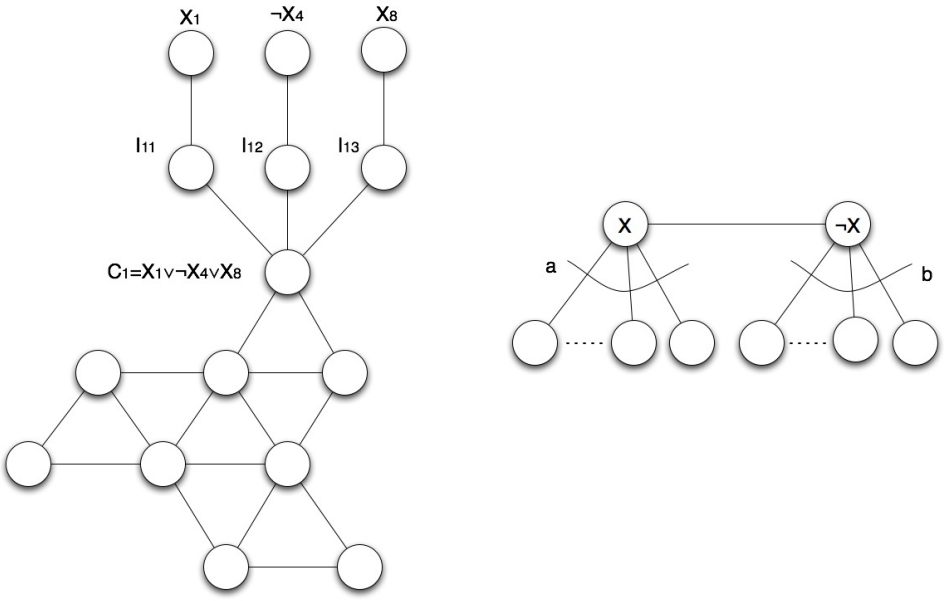


Figure 4.3: Reduction from 3SAT to GNS(d). X_i's refer to variables in the boolean formula. I_{ij} refers to the jth literal in the ith clause. And C_i's refer to the clauses.

Lemma 36. *Computing the social optimum for an instance of* GNS(d) *is NP-complete for all d.*

Proof. We construct a reduction from vertex cover on regular graphs, which is also NP-complete [63]. Consider an instance of vertex cover specified by an r-regular graph $G = (V, E)$. We construct an instance $\mathcal{I}$ of the GNS(d) problem as follows. Let $H = (V', E')$ be a graph obtained by splitting each edge $e = (u, v) \in E$ by $d - 1$ auxiliary nodes $v_{e,1}, \ldots, v_{e,d-1}$, so that $V' = V \cup \cup_{e \in E}\{v_{e,1}, \ldots, v_{e,d-1}\}$, and E' consists of the edges $\cup_{e=(u,v)\in E}\{(u, v_{e,1}), (v, v_{e,d-1}), (v_{e,1}, v_{e,2}), \ldots, (v_{e,d-2}, v_{e,d-1})\}$. For all nodes $v \in V$, let them have the same secure cost C and infection cost L. And we set $C = \frac{L(r(d-1)+1)}{|V'|} + 1$. For each $u \in V' \setminus V$, we have $L_u = 1/|V'|^3$ and $C_u = (C + L)|V'|$. This ensures all nodes in $V' \setminus V$ are insecure, and $\sum_{u \in V' \setminus V} \text{cost}_u(\vec{a}) \leq \epsilon$ for small constant ϵ, for any strategy $\vec{a}$.

Let $B = \{v \in V : a_v = 1\}$ for a pure strategy $\vec{a}$, and let $b = |B|$. It is easy to verify that $\text{cost}(\vec{a}) = \frac{L|V|(r(d-1)+1)}{|V'|} + b + \epsilon + \frac{2L}{|V'|}|\{e = (u, v) : u, v \in V, a_u = a_v = 0\}|$. Therefore, when we set $L > |V| \cdot |V'|$, B is a vertex cover in G of size k, if and only if the social optimum in $\mathcal{I}$ is at most $\frac{L|V|(r(d-1)+1)}{|V'|} + k + \epsilon$. □

For $d = 1$, we also show that while a pure NE always exists, finding the least cost one is NP-complete.

Lemma 37. *Finding the least cost pure NE in a given instance of* GNS(1) *is NP-complete.*

Proof. Our proof is a reduction from Vertex Cover. Let G be an instance of vertex cover. We construct an instance $\mathcal{I}$ of the game in the following manner. We set the contact graph to be $H = (V', E')$ with $V' = V \cup \cup_{i \in V} A(i)$, where the set $A(i) = \{v_{i,1}, \ldots, v_{i,t}\}$, for $t \geq \Delta(G)$, where $\Delta(G)$ is the maximum degree of G. The set E' consists of E along with the edges (i, j), for all $i \in V$ and $j \in A(i)$. The security and infection costs for all nodes in V are identical, C and L, respectively. Set $C = \frac{(t+1)L}{|V'|} + 1$. For nodes in $V' \setminus V$, these corresponding costs are $C' = L'(1 + \epsilon)/|V'|$ and $L' = 1/M$, respectively, where $M \geq |V'|^2 t$. We assume that the initial infection probability distribution is uniform. Therefore, the contribution, $\text{cost}_v(\vec{a})$ of a node $v \in V' \setminus V$ to the total cost $\text{cost}(\vec{a})$ for any strategy vector $\vec{a}$ is at most $\max\{C', 2L'/|V'|\}$, and the total contribution of all such nodes is at most 1. We show that the least cost NE has cost very close to the social optimum.

Let A be a vertex cover for G, with $|A| = a$. Consider the following strategy vector $\vec{a}$: for each $i \in A$, we have $a_i = 1$ and $a_{v_{i,j}} = 0$ for all j, and for $i \notin A$, we have $a_i = 0$

and $a_{v_{i,j}} = 1$ for all j. Following Lemma 28, this vector is a NE because: (i) for each node $i \in A$, there are at least t insecure neighbors (namely, the nodes $v_{i,j}$), (ii) for each $i \notin A$, the number of insecure neighbors is at most $\Delta(G) \leq C|V'|/L$, where $\Delta(G)$ is the maximum degree of G, (iii) if $i \in A$, each node $v_{i,j}$ has no insecure neighbor, and since $C'|V'|/L' = 1 + \epsilon$, such a node won't change its strategy, and (iv) if $i \notin A$, each node $v_{i,j}$ has an insecure neighbor and it will stay being secure. As in the proof of Lemma 36, $\text{cost}(\vec{a}) \leq L + |A| + 1$. Therefore, if G has a vertex cover of size k, the reduced game instance has a pure NE of cost at most $L + k + 1$.

For the converse, let $\vec{a}$ be the strategy vector of a NE, and $A = \{i : a_i = 1\} \cap V$. As in the proof of Lemma 36, $\text{cost}(\vec{a}) = L + |A| + \frac{2L}{|V'|}|\{(u, v) : a_u = a_v = 0,\ u, v \in V\}|$, which implies if A is not a vertex cover for G, $\text{cost}(\vec{a}) > L + |A|$. Therefore, the lemma follows. □

4.4.2 Approximating the social optimum

We describe a general framework to derive approximation algorithms for GNS(d) games for all d. For fixed d, we achieve an approximation ratio of $2d$. For $d = \infty$, we obtain an approximation ratio of $O(\log n)$. Our framework involves the following three steps.

1. Formulate a linear programming relaxation.

2. Let $\mathbf{x}$ be the optimum LP solution. Partially round and filter the variables. Let $\mathbf{x}'$ be the resulting solution.

3. Round the $\mathbf{x}'$ solution appropriately - for constant d, this involves solving a suitable covering problem, while for $d = \infty$ this reduces to a vertex separator problem.

4.4.2.1 An LP Formulation

Let P_{ij}^d denote the set of all simple paths from i to j of length at most d. Let x_v be the indicator variable for node v that is 1 if v is secured. Let y_{ij} be the indicator variable for nodes i and j that is 1 if there is no path $P \in P_{ij}^d$ consisting entirely of insecure nodes. By abuse of notation, for $i = j$, we assume $y_{ii} = 1$ if node i has been secured, i.e., $x_i = 1$. We start with the following integer programming formulation $\mathcal{P}$ of the social optimum.

$$\begin{aligned} \min \quad & \textstyle\sum_v C_v \cdot x_v + \sum_{j \in V} L_j \sum_{i \in V} w_i (1 - y_{ij}) \\ \text{s.t.} \quad & \textstyle\sum_{v \in p} x_v \geq y_{ij} \; p \in P_{ij}^d \\ & x_v \in \{0,1\} \; \forall v \in V \\ & y_{ij} \in \{0,1\} \; \forall i,j \in V \end{aligned} \tag{4.1}$$

The objective function can be interpreted in the following manner: the first part corresponds to the cost of securing nodes, and the second part corresponds to the infection cost, which, for node j is L_j times the sum of the probabilities of all nodes that have a path to j of length at most d consisting entirely of insecure nodes. The first constraint says that in order to separate a pair of nodes i and j, we need to secure at least one node in every path $P \in P_{ij}^d$ between these two. For $i = j$, we define the only path P in P_{ij}^d to consist of the node i.

We relax the IP to a linear program (LP) by changing the last two constraints to $0 \leq x_v \leq 1$ and $0 \leq y_{ij} \leq 1$.

4.4.2.2 Solving the LP and partial rounding and filtering

We now perform the following steps.
(1) Solve the LP: for any fixed d, the number of paths of length at most d, $|P_{ij}^d|$ is at most $n^{O(1)}$, and therefore, the above program can be solved in polynomial time. When d is not a constant, the program cannot be written down efficiently but we can solve it in polynomial time using the ellipsoid method. This requires the construction of a polynomial time separation oracle, which, given a candidate solution $(\vec{x}, \vec{y})$, can decide if it is feasible, or finds a constraint that is infeasible. Such a separation oracle can be designed as follows: define the cost of a path to be the sum of the weights x_v of the nodes on the path. For each pair i, j, compute the shortest path from i to j in the graph restricted to the d-hop neighborhood of node i. If this distance exceeds y_{ij}, the constraints for all the paths $p \in P_{ij}^d$ are satisfied. Else, the constraint corresponding to the shortest such path is violated.

Ellipsoid-based methods are, however, expensive to implement in practice. For the case $d = \infty$, we address this drawback by solving an equivalent polynomial-sized LP in which we introduce a "distance variable" for each pair of nodes and replace the

exponentially-many path constraints given in (4.1) with polynomially-many triangle inequality constraints, and linear number of lower bounds on the distances. It is this more compact LP that we solve in our experiments.
(2) Construct a new vector $\vec{y}'$ in the following manner: for each i, j, $y'_{ij} = 0$ if $y_{ij} \leq 1/2$ and $y'_{ij} = 1$ if $y_{ij} > 1/2$. Next, let $x'_v = \min\{2x_v, 1\}$, for all $v \in V$.

4.4.2.3 Final rounding

We now round the vector $\vec{x}'$ to an integral solution. For $d = 1$, it is easy to see that $\vec{x}'$ is already integral, since each constraint only has two variables. We now consider general d. Consider a pair of nodes i and j such that $y'_{ij} = 1$. By constraint (4.1), along every path p of length at most d between i and j, the sum, over $v \in p$, of x'_v is at least 1. It follows that along every such path p, there exists at least one vertex $v \in p$ with $x'_v \geq 1/d$. Consider now the following filtering procedure: if $x'_v \leq 1/d$, we set $x''_v = 0$; otherwise, we set $x''_v = 1$. It is clear that all the constraints of the LP are satisfied, and the cost of $\vec{x''}$ is at most d times the cost of $\vec{x'}$, yielding a final $2d$ approximation.

We finally consider the $d = \infty$ case. In this case, we are left with a minimum weighted vertex multi-cut problem, where we would like to determine the minimum weight of vertices that can separate all the pairs (i, j) for which $y'_{ij} = 1$. The elegant LP rounding algorithm of [68] yields an integral solution for the vertex multi-cut problem, whose cost is $O(\log n)$ times the cost of fractional solution. We can thus find a set X of vertices to secure such that all pairs of vertices for which $y'_{ij} = 1$ are separated and $\sum_{v \in X} C_v$ is at most $O((\log n) \sum_v C_v x'_v)$.

Putting the above analyses together, we have the following.

Theorem 38. *For any fixed d, the social optimum for an instance of* GNS(d) *can be approximated to within a factor of $2d$ in polynomial time. For $d = \infty$, we obtain an $O(\log n)$-approximation to the social optimum, where n is the number of nodes in the contact graph.*

4.5 Experimental results

We now empirically study the properties of NE and the performance of our algorithms. We use two classes of graphs: (i) random geometric graphs formed by distributing n^2 nodes uniformly at random in an $n \times n$ square and add an edge between a pair

of nodes if there distance is no more than 1, and (ii) power law graphs generated by preferential attachment process [27]. These two graph classes are very different, with the former being a model for wireless networks, while the latter suited for the Internet [62], World Wide Web [27], and email networks [58]. Also, they have very contrasting properties, e.g., the latter class has larger separators, and we expect to see effects of these differences. We set the infection costs to be identical for every node (this can be done without loss of generality for the pure NE analysis) and the security costs are chosen uniformly at random between 0 and the infection cost.

Our main experimental observations are the following.

1. *Convergence time for best response strategies*: We find that best response works pretty well in practice. For $d = \infty$, we find the convergence time to be linear in the number of nodes for both graph classes, while it seems to be sub-linear in the case of $d = 1$. For the d-neighborhood model, with $1 < d < \infty$, best response does not converge to a NE quite often, suggesting that even on average, these games do not have NE.

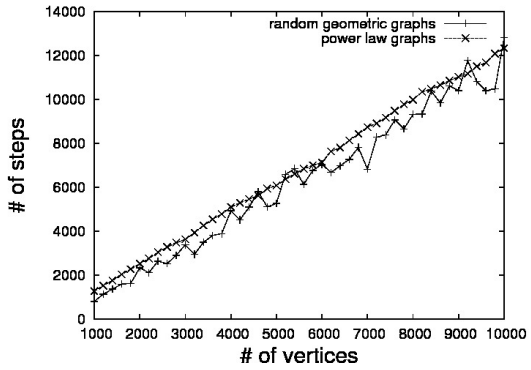
2. *Structural properties of NE and the quality of NE*: We find that high degree nodes tend to be secured in the NE for the local game. Additionally, we find that the cost of NE is very low for $d = 1$ in both the graph classes, but it is somewhat high for $d = \infty$.

3. *Performance of our approximation algorithms for the social optimum*: While we show a worst case bound of $O(\log n)$ for approximating the social optimum (Section 4.4), we find that our algorithms perform much better in practice. For $d = 1$, the approximation bound is very close to 1; while for $d = \infty$, it seems to be a constant.

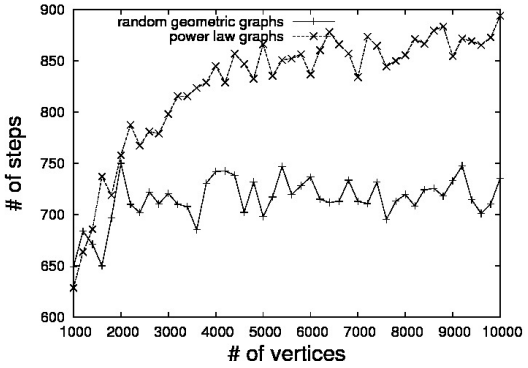
4.5.1 Convergence times for best response strategies

We implement best response in a round robin fashion on both the graph classes and study the convergence time; note that the results of Section 4.3 imply that this converges to a NE. Figure 4.4a shows that the convergence time of the global model for random geometric and power law graphs grows linearly with the number of nodes. Figure 4.4b shows the corresponding plots for the local model and they seem to grow much slower

than in the $d = \infty$ case. Also, for the d-neighborhood model, we find that best response often does not converge to a NE.



(a) Convergence time in the global model ($d = \infty$) for random geometric graphs and power law graphs.



(b) Convergence time in the local model ($d = 1$) for random geometric graphs and power law graphs.

Figure 4.4: Convergence time.

4.5.2 Structural properties of NE

In Figure 4.5, we examine the degrees of secured nodes in the NE computed by best response on power law graph with 5000 vertices, and we find that they tend to be high. In fact, the degree distribution of the secured nodes seems to mirror the overall

degree distribution in the graph. We also study the quality of NE in the local and global models. Figure 4.6a and 4.6b show that the cost of NE is very low for the local model in both graph classes. The ratio to optimal value is at most 1.3. In contrast, Figure 4.7a and 4.7b show that this ratio is larger for the global model, about 7 in both graph classes. We note that this ratio is for the case of non-uniform costs; we expect the ratio to be smaller with uniform costs, especially for power-law graphs owing to their high vertex expansion.

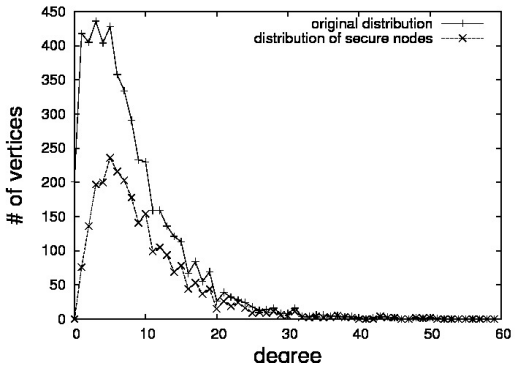
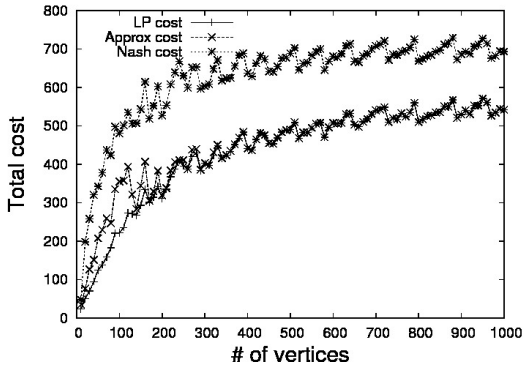


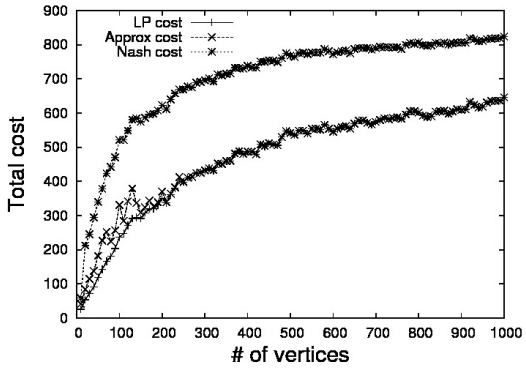
Figure 4.5: Properties of secured nodes in NE in power law graphs.

4.5.3 Empirical performance of approximation algorithms

We now study the empirical performance of the algorithms we design in Section 4.4 for approximating the social optimum. Since computing social optimum is very expensive, we use LP optimal values as lower bound. Figure 4.6a and 4.6b show that our approximation algorithm's cost is almost the same as the LP lower bound for the local model. For the global model, Figure 4.7a and 4.7b show that the approximation algorithm's cost is within a constant of the LP lower bound, in contrast to the worst case $O(\log n)$ bound we prove. Additionally, we observe that our approximation algorithm has a much better guarantee for power law graphs than for random geometric graphs.

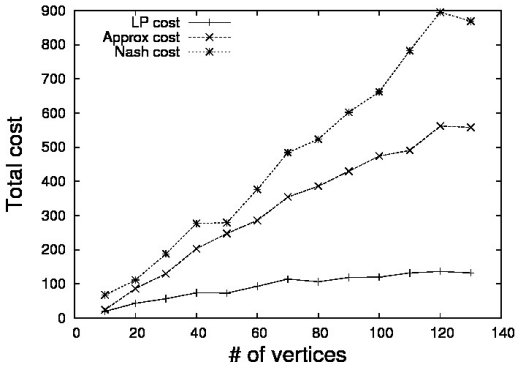


(a) The costs of the LP solution, our approximation algorithm, and the Nash equilibrium computed by best response, for the local model in random geometric graphs.

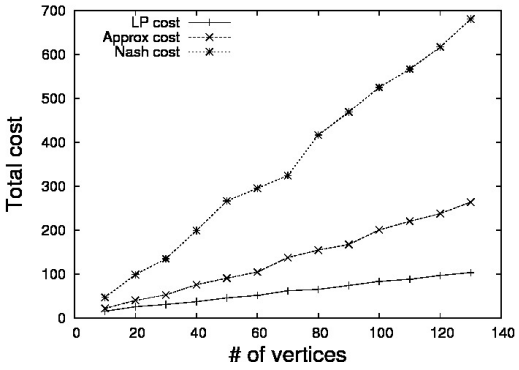


(b) The costs of the LP solution, our approximation algorithm, and the Nash equilibrium computed by best response, for the local model in power law graphs.

Figure 4.6: Costs comparison for the local model.



(a) The costs of the LP solution, our approximation algorithm, and the Nash equilibrium computed by best response, for the global model in random geometric graphs.



(b) The costs of the LP solution, our approximation algorithm, and the Nash equilibrium computed by best response, for the global model in power law graphs.

Figure 4.7: Costs comparison for the global model.

4.6 Conclusion

Non-cooperative games have been recognized as a useful paradigm for studying decentralized network security problems; however, the resources needed for individual decision making are important issues for the implementability of such games. In this paper, we have developed a framework for network security games parametrized by the amount of local information available for individual decision making. We find this parameter plays an important role in the structure of the equilibria, and needs to be taken into account in such analysis.

NE are considered as natural operating configurations in such systems with selfish users. Therefore, ensuring that the system has efficient NE is desirable (equivalently, a low price of anarchy (PoA)) for network planners. Specifically, if the network planner has a limited budget to secure k nodes, an important design problem is to choose a subset of nodes to secure so that the graph restricted to the remaining nodes has low PoA; such a strategy is also referred to as a *Stackelberg* strategy for the network planner [111]. Lemmas 30 and 32, which bound the PoA in terms of the network parameters, suggest natural heuristics to design stackelberg strategies for the network planner. We discuss this briefly below.

In the neighborhood model, Lemma 30 shows that PoA is bounded by $\Delta + 1$. Therefore, given a budget to secure k nodes, the Stackelberg question is to choose a subset of nodes to secure, so that the maximum degree of the residual graph is minimized. An analogous question, dual to this, is the following: for a given target maximum degree Δ', choose the smallest set k of nodes to secure so that the maximum degree in the residual graph is Δ'. Both these versions are NP-complete to solve optimally, but greedy heuristics are likely to perform well. In the global model, Lemma 32 shows that the PoA is bounded by $1/\alpha(\mathcal{G})$. The analogous question of finding an optimal Stackelberg strategy is NP-complete in this case also. We can use the spectral clustering algorithm of [78], which finds an (α, ϵ) clustering of low cost using at most an ϵ fraction of the edges, while ensuring that each cluster has expansion at least α, as a natural heuristic for this problem.