OPTIMIZING DIFFUSION AND PRICING IN NETWORKS OF INDEPENDENT AGENTS

By

Ameya Hate

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Approved by the
Examining Committee:

Elliot Anshelevich, Thesis Adviser

Malik Magdon-Ismail, Member

Sanmay Das, Member

John Mitchell, Member

Rensselaer Polytechnic Institute
Troy, New York

March 2012
(For Graduation May 2012)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Diffusion is a process through which contagious entities (such as ideas, viruses etc.) spread through a population of objects over links between them. An important area for the study of diffusive process is social networks, where objects are individuals and they are linked if they know or can influence each other. Because social networks play an important role in the spread of ideas, influence, and information among individuals of the network, study of diffusive processes in such settings is of interest to many. The entities that have an ability to spread through a network may not always be benign. A malicious meme or a contagious disease can also diffuse over a social network. In such cases it becomes important to study methods and techniques to limit the spread of such diffusive processes.

In this thesis we try and address some of these problems. We first look at the scenario where a malevolent entity is diffusing through a network and there are limited resources available to curb its spread. The goal here is provide rigorous theoretical analysis and come up with provably efficient algorithms. The model used for this analysis is simple but it does capture the essence of hardness of these problems. On the other hand, we also study diffusion in a more general scenario which takes into consideration many of the complexities encountered in real situations. Specifically, we develop heuristics for maximizing diffusion of evacuation and warning messages in real life scenarios taking into consideration the complex social interactions that drive the diffusion.

We also study a somewhat different topic that deals with efficiency of the Internet. One of the principal factors behind phenomenal growth of the Internet has been decentralization of control, which also allows it to be modeled naturally as a system of interacting but independent, self-interested agents. The Internet can be viewed as a collection of ISPs or ASes (Autonomous Systems) that are interested in routing and pricing traffic to maximize their individual revenues. While the study of efficiency of the Internet seems to be unrelated to diffusive processes, many of the analytical tools required to examine both scenarios are the same. Both contain

networks of agents behaving independently of each other. In fact, as we show in our work, efficiency of the Internet depends on optimal allocation of resources which can be achieved by diffusion of prices through the network. As ISPs are generally interested in maximizing their individual profits, this question is naturally studied in the framework of non-cooperative game theory.

CHAPTER 1
Introduction

Graphs have been used extensively to model the environment of many problems. The mathematical structure of graphs used to model pairwise relation between objects lends itself to many real world situations. Graph theory has proved to be critical in the study of a large number of areas ranging from transportation networks to epidemiology. One such area is the study of diffusive processes.

Diffusion is a process through which contagious entities (such as ideas, viruses etc.) spread through a population of objects over links between them. An important area for the study of diffusive process is social networks, where objects are individuals and they are linked if they 'know' or can 'influence' each other. Recently, due to the emergence of computers and the Internet, massive amounts of data regarding the structure of social networks has become available. This has in turn led to a large amount of research in the area.

Because social networks play an important role in the spread of ideas, influence, and information among individuals of the network, study of diffusive processes in such settings is of interest to many. An idea or meme, such as adopting a new product on the market, can originate independently in a group of individuals. This behavior may now spread through the population by means of the existing social network. Understanding such a process is very important to organizations such as advertising companies that make use of viral marketing. They target a subset of individuals to advertise to directly and then rely on word-of-mouth publicity for the adoption of their product to spread to the rest of the social network. This possibility of commercial exploitation has fueled a significant amount of research in the area. The entities that have an ability to spread through a network may not always be benign. A malicious meme or a contagious disease can also diffuse over a social network. In such cases it becomes important to study methods and techniques to limit the spread of such diffusive processes. Research has been carried out to study and identify properties of a social network which ensure infection is contained to a small number of individuals. Yet another way to look at containing this spread is to engage in active vaccination of nodes in the network. A vaccination generally refers to the fact that once a node is immunized, it cannot be infected and hence helps in curbing the spread of the infection. Such an active vaccination strategy can be of two types: preventive and dynamic. Preventive or prophylactic vaccination is given before an infection begins. The idea being that some key nodes in the network should be vaccinated such that even if the infection breaks out, it will not be able spread to a large population. A dynamic vaccination strategy has to be administered when the infection has already broken out in the population. Nodes of the network are vaccinated while the infection is spreading simultaneously. There is usually a restriction on the amount of vaccination that can be given at each step. In both vaccination strategies the goal is to minimize the damage by forming a cut in the underlying graph to isolate the infection from as many nodes as possible.

In real life scenarios, diffusion takes place over networks that have complex attributes. In the case where the diffusing entity is news of an event, the extent to which the news spreads may depend on a multitude of factors like the number of different sources from which the news originated, their trustworthiness, the trust between individuals that exchange information, etc. If the news is a warning of evacuation following a natural, technological, or willful disaster, the individuals may evacuate and leave the network they were part of. This makes the graph on which diffusion occurs dynamic. If the diffusing entity is a disease, there may be several strains of the virus/bacteria spreading through the network and individuals in a population may have different levels of immunity towards them. All this possible heterogeneity makes it difficult for the diffusive processes to be analyzed theoretically. Tractable agent based models which can scale up to real life multi-million node networks can help us uncover the intricacies of diffusion. This is the first step towards developing good heuristics and algorithms for realistic applications where diffusion occurs.

In this thesis we try and address some of these problems. We first look at the scenario where a malevolent entity is diffusing through a network and there are limited resources available to curb its spread. The goal here is provide rigorous theoretical analysis and come up with provably efficient algorithms. The model used for this analysis is simple but it does capture the essence of hardness of these problems.

On the other hand, we also study diffusion in a more general scenario which takes into consideration many of the complexities encountered in real situations. Moreover, this diffusion model can be adapted to a wide range of real life applications. As mentioned earlier, it is difficult to analyze more general models theoretically; so we introduce a new paradigm for developing efficient heuristic algorithms for such problems. Specifically, we develop heuristics for maximizing diffusion of evacuation and warning messages in real life scenarios taking into consideration the complex social interactions that drive the diffusion.

Efficiency of the Internet

We will also study a somewhat different topic that deals with efficiency of the Internet. The process of diffusion discussed above, can take place in physical or virtual networks. In the past few years, there has been an astounding rise of virtual social networks by leveraging the power of the Internet. Their ubiquitous impact on modern life is a testimony to the growth of Internet in the past two decades. One of the principal factors behind this growth has been the decentralization of control, which also allows it to be modeled naturally as a system of interacting but independent, self-interested agents. More specifically, the Internet can be viewed as a collection of ISPs or ASes (Autonomous Systems) that are interested in routing and pricing traffic to maximize their individual revenues [1, 2, 3]. While the study of efficiency of the Internet seems to be unrelated to diffusive processes, many of the analytical tools required to examine both scenarios are the same. Both contain networks of agents behaving independently of each other. In fact, as we shall show in our work, efficiency of the Internet depends on optimal allocation of resources which can be achieved by diffusion of 'prices' through the network. The study of large decentralized networks of self-interested agents, with regard to their efficiency, has sparked an enormous amount of interest, as the insight thus earned can be used to extract maximum utility from existing infrastructure, as well as to make good policy decisions.

While the number and variety of applications that run over the Internet have grown tremendously, its underlying architectural components and protocols have not changed significantly. Reliance on simple traffic routing algorithms like shortest path implies that network resources are not optimally used. There has been much research activity on intra-domain traffic engineering in the recent past [4, 5, 6], which are likely to gradually make their way into practice. In contrast, inter-domain routing/traffic engineering practices have not changed much, and the topic also remains less researched. One of the difficulties in doing inter-domain traffic engineering is that it requires involvement of (agreement by) multiple Internet Service Providers (ISPs) that are interested solely in maximizing their own profits, and their individual objectives are often in conflict with each other. Moreover, inter-domain traffic engineering issues are also closely related to how traffic forwarding services are priced at the inter-domain level. The current inter-domain traffic routing and pricing strategies are typically policy driven, and do not in general optimize the overall use of Internet resources, or maximize ISP profits. Inter-domain routing follows the BGP standard [7] that only attempts to minimize the number of ISP hops on the path of a flow (in addition to following local policy considerations and certain heuristic rules). Traffic exchange contracts between ISPs are still negotiated manually and span months if not years. These contracts when finally resolved are often simplistic in nature, and charges are typically based on the total traffic offered by the customer ISP irrespective of their destination (point-to-anywhere pricing) [8]. It has been argued correctly that point-to-point or destination-based pricing can result in better efficiency of network resource usage [9]. Moreover, recent measurements also show fast changing patterns for inter-domain traffic [10]; inter-domain service pricing and traffic engineering solutions need to be flexible enough to adapt to such dynamically changing traffic patterns, and variations in customer requirements and willingness-to-pay values.

Given the current scenario, it is worth studying the question: how should inter-domain traffic forwarding services be priced? More specifically, this involves finding the “right” price that ISPs should charge for the traffic forwarding services they provide to their upstream neighbors, typically done through SLA agreements (short- or long-term) formed between neighboring ISPs or Autonomous Systems (ASes) in the Internet. As ISPs are generally interested in maximizing their individual profits, this question is naturally studied in the framework of non-cooperative game theory, as we do in this work. The issues that are of interest in this context are whether there exist prices that constitute an *equilibrium*, i.e. prices which the ISPs (acting out of of self-interest) are not likely to deviate from, and whether such prices can be computed efficiently and result in efficient use of the network. Another issue worth investigating is whether stable and efficient prices can be obtained through distributed pricing negotiations and traffic forwarding policies, that can be implemented with modest changes to existing inter-domain routing policies and message exchange protocols. These are some of the questions that we try and address in this thesis.

We now take a brief look at related work in the area of diffusive processes and inter-domain routing. In section 1.2, we give a short description of our contribution to these areas.

1.1 Background

Here we give a short survey on the literature related to the topics of research in this thesis. We first look at work done in diffusion processes in relation to epidemic spread and influence propagation. Then we briefly catalog the literature on controlling spread of epidemics and give detailed background work for the *Firefighter problem*, which we define later in the section. Next, we look at research in social sciences about human response to warnings and evacuation messages. This knowledge is crucial for construction of realistic diffusion models. Finally, we end by giving a background work on selfish routing in connection to inter-domain routing on the Internet.

1.1.1 Diffusion processes

Properties of Large Networks There have been many studies [11] that suggest large networks observe power law distribution and they can be represented using scale-free models. These models can be used to generate networks that closely resemble networks like the World Wide Web (WWW) graph [12] and realistic social contact networks [13]. Studies like [14] show that simple models with modifications as opposed to random graphs can also be used to obtain properties of real world networks. Further, there also has been a great deal of research [15] on how these properties of real world networks affect the spread of infectious processes and on distribution of antidotes [16] to control the spread of such infections.

Epidemic Spread Infectious diseases are similar to infectious ideas and have dynamics similar to evacuation. Many studies use the homogeneous SIR model of Reed and Frost [17]. Both Markov chains (e.g., [18]) and differential equations [19, 20] have been used. Most of these methods make homogeneity assumptions about the underlying network, and thus do not take full advantage of the network topology. More realistic models must relax homogeneity, for example invoking a social structure for contact based spread, or varying disease transmission probability.

Ganesh et al. [21] enquire about properties of network make an epidemic spread weak or potent. They observe that this potency depends on how the cure to infection rate of the diffusive process compares with graph parameters like spectral radius and isoperimetric constant. On the other hand Moore and Newman [22] look at probabilistic model for infection and transmission of disease. They study how the relation between transmission and infection rates and the threshold for site and bond percolation of the network affect the epidemic behavior in their model. Similarly, in [23], Pastor-Satorras and Vespignani study epidemic dynamics in bounded scale free networks with soft and hard connectivity cut-offs. They study how these finite cut-offs affect epidemic thresholds. Leskovec et al. [24] take up the problem of detecting outbreaks of spreading processes in a network. Their objective amounts to placement of sentries in a network so that outbreaks are detected optimally. They observe that many of these objectives display a property of submodularity and exploit that to obtain near optimal solutions.

Aspnes et al. [25] look at a different problem that is actually a game. Here, the infection is a worm and the infected nodes decide which vertices should be infected in the next step while the detector nodes send alerts to catch this spread of worms. These problems of spread of infection and subsequent attempts of controlling the spread have some properties similar to the problem of Length bounded cuts [26]. A L-length-bounded cut in a graph with source and sink, is a cut that destroys all source to sink paths of length at most L. Another problem studied by Sonnerat and Vetta [27], looks at defending graphs such that network connectivity is maintained when attacks on the graph result in certain nodes being destroyed.

Submodularity At this point we take a brief look at the concept of submodularity since it shows in many of the papers discussed here. The formal definition of submodularity is as follows:

Given a ground set U, a function $f : 2^U \rightarrow \mathbb{R}_{\geq 0}$ is a monotone submodular function iff

i. $f(A) \leq f(B)$ whenever $A \subseteq B \subseteq U$; and

ii. $f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \, \forall \, A, B \subseteq U$, equivalently

iii. Given that $A \subseteq B \subseteq U$, for any set C: $f(A \cup C) - f(A) \geq f(B \cup C) - f(B)$.

Essentially submodularity is the notion of concavity applied to set functions. It also captures the notion of *diminishing returns* as can be seen from condition [iii]. It states that the change in the value of the function obtained by adding a set C to a smaller set A will be at least as large as the change obtained by adding the same set C to any superset of A.

Diffusive processes spread in a network by getting transmitted from an infected node to its neighbor. If A is a set of nodes in the network and N_A is the set of its neighbors then consider a set function $f(A) = |N_A|$. The function $f(\cdot)$ can be easily shown to be submodular. This gives us an intuition so as to why structure of many of the problems in this field turn out to be submodular in nature.

Submodular functions is a well studied area of research in itself and some results from this area are helpful in the study of diffusive processes. Some of the particularly useful results are as follows: work done by Nemhauser et al. [28] says that for the problem of maximizing a submodular set function over sets that obey matroid constraints, a greedy heuristics gives a $1/2$ approximation. In [29], Calinescu et al. extend this result and give a $1 - 1/e$ approximation but the algorithm is randomized in nature.

1.1.2 Influence Propagation in Networks

Information diffusion is a long researched area [30], with work in online communities becoming very active topic recently, on account of innovation diffusion, viral marketing and computer virus spread [31, 32, 33, 34, 35, 36, 37, 38]. Most work uses a static network, which is not a good approximation. Kempe et al. [39] look at the problem of maximizing the spread of influence in social networks. The problem can be described informally as trying to find an optimal group individuals that can maximize a spread of an idea in the network. For example if these individuals are convinced to adopt a new idea or product, it would trigger a large cascade of further adoptions. This cascade of adoptions is based on the idea that individual nodes are influenced by the actions taken by their neighbors and each node's tendency to adopt new ideas increases monotonically as more of its neighbors adopt it themselves.

They consider two models for the spread of influence.

- In the *Linear Threshold Model*, the magnitude of influence node w has on node v is reflected by a non-negative weight $b_{v,w}$ such that $\sum_{w \in N(v)} b_{v,w} \leq 1$. Also, every node chooses a threshold uniformly at random in the interval $[0, 1]$. If the at any step, the weighted fraction of v's neighbors that have adopted the new idea crosses its threshold, then v itself adopts the idea.

  It is interesting to note that if the thresholds are made to be constant or part of the input instead of being uniformly random, the model at once becomes intractable and there does not exist any non-trivial approximation unless P=NP. This is also discussed in the paragraph about intractability later in the section.

- They also consider the *Independent cascade model* where whenever a node v first adopts a new idea, it tries once to influence its neighbors (say w) to adopt the idea and succeeds with a certain probability $p_{v,w}$ - which is a parameter of the system.

The authors look at the influence maximization objective on both these models which, for a parameter k, amounts to finding a k-node set to make active initially so as to maximize the spread of this new idea. Such an objective function is interesting when the resources available can be used to target limited number of individuals in a network.

Through the course of the paper the authors show that both these models are essentially equivalent and obtain an $(1 - \frac{1}{e} - \epsilon)$ approximation for the influence maximization objective. This means that they give an algorithm that activates at least $(1 - \frac{1}{e} - \epsilon)$ times the optimal number nodes that can be activated using k initial nodes. Here ϵ is an error that can be made arbitrarily small given a resulting blowup in running time of the algorithm. They achieve this result using a greedy hill climbing algorithm based on submodular functions.

In an extension [40] to this paper, the authors consider a generalization of the independent cascade model. As opposed to the original model where the probability of a node being influenced did not depend on previous tries by its neighbors, here the probability that a node will be subsequently influenced will be greater if more of its neighbors have tried and failed. They achieve similar approximation results for this model which they term as the *Decreasing Cascade Model*.

Bharathi et al. [41] study a competitive version of the influence maximization problem. This game arises when multiple companies market competing products using viral marketing. For a two player version, they give a $(1-1/e)$ approximation for computing the best response to an opponents strategy. As shown in the earlier work, influence maximization for a single player is APX-hard in general. In this paper, the authors show that if the underlying graph is a bi-directed tree, then an FPTAS (Fully Polynomial Time Approximation Scheme) can be found.

A Deterministic Threshold Dreyer and Roberts [42] study a different model of diffusion process. They emphasize on a model they call as the k-threshold process which can be defined as follows:

Consider that there are two states $(0, 1)$ that a node can take. 0 represents the old state that all the nodes were in the beginning and 1 is the new state. In the k-threshold process a vertex enters state 1 from 0 if at least k of its neighbors are in state 1 and never leaves state 1 once entered. The authors discuss this diffusive process mainly from the point of view of infectious diseases hence the irreversibility. The paper studies the problem of identifying sets of vertices (referred to as irreversible k-conversion sets) with the property that if they are all in state 1 at the beginning, then eventually all vertices end up in state 1. This can help develop strategies of vaccination so that the resultant graphs have large irreversible k-conversion sets.

They prove that it is NP-complete to determine if a graph has an irreversible k-conversion set at most size d, at least for $k \geq 3$, by reduction from the Independent Set problem. They then give exact values and bounds on the size of the smallest irreversible k-conversion set for special graphs like multipartite graphs, trees and various kinds of grids. They also show that the trivial lower bound: k, on the size of the k-conversion set is tight as it is achieved by several graphs.

Intractability of Influence propagation Chen [43] has studied the same problem as presented in the previous section [42]. In this study, the author explores the tractability of the k-conversion set problem which he refers to as the *target set selection* problem. He shows inapproximability results for several variants of this problem.

One of the main result of the paper shows that the problem of minimizing a set S, while ensuring that targeting S would influence the whole network into adopting the new idea, is hard approximate within a polylogarithmic factor. The proof is based on a reduction from the Minimum Representative problem. This answers one of the open questions put forth by Dreyer and Roberts [42].

As we can see that the Target set selection (TSS) problem is different form the problem considered by Kempe et al. in [39, 40] in couple of ways. First, Kempe et al. focus on the maximization problem for any give k, find a target set of size k to maximize the (expected) number of active vertices as the end of the process while TSS asks for target set of minimum size that guarantees that all vertices are eventually active. Also, in TSS the thresholds for vertices are deterministic and explicitly given while Kempe et al. [39] consider probabilistic thresholds. Kempe et al. also show that the maximization problem they consider is inapproximable within a ratio of $n^{1-\epsilon}$ for predetermined thresholds. The results from Chen's paper [43] gives us further evidence that without additional assumptions such as probabilistic threshold in [39, 40], the problem is completely intractable.

The results of inapproximability of the TSS problem extends to the case where only a fixed fraction of vertices are to be made active. They also show that for the majority threshold setting, where a vertex becomes active if at least half its neighbors are active, the TSS problem shares the same hardness of approximation ratio in a general setting. The inapproximability results also hold for other cases where thresholds are very small (as small as 2) and when they are unanimous, i.e. every node has the same threshold.

1.1.3 Immunizing Nodes against Epidemic spread

Giakkoupis et al. [44] look at the problem of immunizing nodes against spread of infections in the network. They study analytically the dynamic propagation model where the birth-death process evolves over time. Viruses are continuously propagated in the network, but they may also die. In a discrete time model, a virus dies at node i in a single time step with probability p_i. Here p_i is referred to as the healing probability of node i. A virus is propagated from node i to node j with probability p_{ij}. Under different formulations, the authors provide necessary and sufficient conditions for an epidemic outbreak. Their analysis complements and extends that of Wang et al. [45].

The authors define specific immunization problems for each of the virus-propagation models, and propose immunization algorithms. For the independent-cascade model the proposed algorithm is a greedy method designed to minimize the spread of the virus in the network. For the dynamic-propagation models the virus spread depends on the eigenvalue of an appropriately defined matrix of the network. So their algorithm attempts to reduce this eigenvalue with the least immunization cost.

They study the algorithms experimentally on both real and synthetic networks.

Further, they observe that contrary to popular belief the greedy algorithm that removes the node with the highest degree does not yield the best results. The failure of this highest-degree heuristic is especially pronounced in graphs with high clustering coefficient, a case that one comes across in many social networks.

Related Notions of Graph Cut Protecting nodes of a graph from an infection that is spreading on the graph gives rise to some different ideas of graph cuts. Engelberg et al. [46] introduce a budgeted version of the Multiway cut and the k-cut problem. In the budgeted version of the Multiway cut problem, along with the problem instance a positive integer B is provided which is the budget. The objective is to find a subset of edges whose cost is within the given budget and whose removal maximizes the value of the given objective function. For the Multiway cut problem they consider two objectives, one that maximizes the number of pairs of terminals separated and other that maximizes the number of terminals that are isolated. The authors observe that though these two problems are very much related, they capture different ideas of theory of cuts and hence vary in their level of hardness. Though both problems are proved to be NP hard, they give a constant factor approximation for the terminal isolating objective while for the terminal pair separating objective, they provide a constant factor approximation on trees.

Hayrapetyan et al. [47] introduce a problem of unbalanced graph cuts which is essentially an optimization problem that tries to minimize the number of nodes on one side of the cut while not letting the cut size exceed a pre-specified budget. They prove that this problem is NP complete and give a $(\frac{1}{\lambda}, \frac{1}{1-\lambda})$ bi-criteria approximation that exceeds the budget by at most a factor of $1/\lambda$ while keeping the size of the source side within a factor of $\frac{1}{1-\lambda}$ of the optimal.

A Game Theoretic Look In their work [48], Aspnes et al. look at a game for modeling containment of the spread of viruses in a graph. Each node of the graph chooses to either install antivirus software at some known cost, or risk infection which has its own cost. The virus can start at a random initial point in the graph and infects every neighbor that is unprotected. A node gets infected if a virus can reach it without being stopped by some intermediate node.

The goal of individual nodes is to minimize their individual expected cost. The authors analyze the model theoretically and provide characterization of Nash equilibria for an antivirus software installation game in which each machines owner separately chooses whether to install the software, without regard to the effect on other machines. They also show that finding either the most or least expensive equilibrium is NP-hard, but that some Nash equilibrium can be computed in $O(n^3)$ time.

They also show that any population of nodes will quickly converge to a Nash equilibrium by updating their strategies locally based on the other nodes strategies. On the other hand they observe that allowing selfish users to choose Nash equilibrium strategies is highly undesirable, because the price of anarchy (ratio of cost of the worst possible NE to the cost of the optimal solution) is an unacceptable $\Theta(n)$ in the worst case.

They show that though it is NP-hard to compute the socially optimal solution, this problem can be reduced to a combinatorial problem that they call the sum-of-squares partition problem. Using a greedy algorithm based on sparse cuts, they show that this problem can be approximated to within a factor of $O(\log n)$, giving the same approximation ratio for the inoculation game.

1.1.4 Firefighter Problem

When nodes of a graph need to protected from an infection (or fire) that has already started spreading, the response leads to a *dynamic cut* on the graph. This idea is well encapsulated by the *firefighter problem*. This discrete-time, dynamic problem was introduced by B. Hartnell in 1995 [49]. The problem can be described as follows:

Let (G, r) be a connected rooted graph. At time 0, a fire breaks out at r. At each subsequent time interval, the firefighter defends some vertex which is not on fire, and then the fire spreads to all undefended neighbours of each burning (i.e., on fire) vertex. Once burning or defended, a vertex remains so for all time intervals. The process ends when the fire can no longer spread.

MacGillivray and Wang [50] prove that this problem is NP-complete even when restricted to bipartite graphs and then provide a certain results for trees. In her thesis, Fogarty [51] looks at the Firefighter problem on various types of infinite grids. The objective is to find minimum number of firefighters required so that it would be possible to stop the spread of fire in finite steps.

Develin and Hartke [52] also look at the problem of containing fire on grids. They prove a previously proposed conjecture that $2d-1$ firefighters are necessary to contain a fire outbreak in a d-dimensional square grid. They also prove that for any fixed number f of firefighters, there is a finite outbreak of fire in which f firefighters per time step are insufficient to contain the outbreak.

A related problem is considered by Crosby et al [53]. For a fixed positive integer n, they consider the problem of constructing a graph G that is optimal in the sense that the expected damage resulting from a random outbreak of f fires on G is minimum for graphs of order n. They provide several conditions on the structure of the graph in order for it to be optimally resistant to random outbreaks of fire.

Leizhen and Weifan [54] propose to measure a graphs capability of resisting fire spread. They study the fire defending ability of a graph as a whole by considering the average percentage of vertices the firefighter can save. They define the surviving rate of graph to be the average percentage of vertices that can be saved when a fire randomly breaks out at a vertex of the graph. They also give lower bound on the surviving rate for special graphs like trees and outerplanar graphs.

King and MacGillivray [55] study the optimization version of the firefighter problem where the objective is to determine the maximum number of vertices that can be saved, i.e., that are not burning when the process ends. The process is said to end when the fire can no longer spread. They define the following problem as the 3-Fire problem: Given a rooted cubic graph (G, r) and a positive integer k, then if the fire breaks out at r, is there a strategy for defending nodes such that at most k-vertices burn. The authors subsequently prove that the 3-Fire is NP hard.

They then depart to take a look at a different version of the firefighter problem. Most of the earlier work concentrated on minimizing the number of nodes burnt but there is also a complementary problem which asks if a given set of nodes can be saved. In the new version, along with the rooted graph, a subset of nodes $S \subseteq V(G)$ is also given. The question is that if a fire breaks out at r then does there exist a defending strategy that that can save all nodes of set S. Interestingly, they show that the problem is NP-complete even when restricted to S being the set of leaves of a full rooted tree of maximum degree three. They then describe a polynomial time algorithm to decide whether it is possible to save a given set S of vertices in a graph with maximum degree three, provided the fire breaks out at a vertex of degree at most two.

Firefighting on Trees In [56], Cai et al. take a look at the algorithmic aspects of the firefighter problem on trees. It had been previously proved that a simple greedy algorithm given an 1/2 approximation for maximizing the number of nodes saved in a tree [57]. In [56], the authors give a $(1 - 1/e)$ approximation algorithm for the firefighter problem on trees. Their algorithm uses randomized rounding of an LP relaxation of a $0 - 1$ integer program formulated by MacGillivray and Wang [50]. They also prove that $(1 - 1/e)$ is the best approximation factor one can get using any LP rounding technique with the same LP.

The Resource Minimization Version Chalermsook and Chuzoy [58] study the fire containment version of the firefighter problem. In their model, once a fire starts, at each time step the firefighters can save up to k vertices of the graph, while the fire spreads from burning vertices to all their neighbors that have not been saved so far. Also there is a given subset $T \subseteq V$ of vertices called terminals that need to be protected from fire. The objective is to minimize k - the maximum number of vertices to be saved at any time step, so that the fire does not spread to the vertices of T. They term this as the Resource Minimization Fire Containment (RMFC) problem.

As mentioned previously, King and MacGillivray [55] proved that this problem is NP-complete even on trees. Chalermsook and Chuzoy's main result is an $O(\log^* n)$-approximation algorithm for RMFC on trees. The algorithm rounds a natural LP-relaxation of the problem. The technique is fairly complex and contains a randomized procedure that forms a near-optimal and almost feasible solution such that if a certain small number of vertices are removed from every layer of the tree then the solution would be feasible. Further steps involve making sure that this list of removed nodes is small and to accommodate them without blowing up the final solution too much.

It should be noted that the approximation factor of $O(\log^* n)$ is rather unusual, and the only known natural problem with $\Theta(\log^* n)$ approximability threshold is Asymmetric k-Center problem and this problem appears to be somewhat similar in nature to RMFC on trees. The Asymmetric k-Center problem is defined on an asymmetric metric (V, d) ($d(x, y)$ may not equal $d(y, x)$) where the goal is to output a set of k centers $C \subseteq V$ such that $|C| = k$ that minimizes the function:

$$\phi(C) = min\, max_{v \in V} d(C, v).$$

The similarity between stems from the fact that as in the Asymmetric k-Center problem we are trying to select nodes to immunize in RMFC that curtail the spread of the fire by largest margin. This in turn depends on the length of the shortest paths between the nodes immunized and the nodes of T that we are trying to protect.

Apart from this result, the authors also prove that an even stronger LP-relaxation has an $\Omega(\log^* n)$ integrality gap. Finally they consider RMFC on directed layered graphs and show an $O(\log n)$-approximation LP-rounding algorithm, matching the integrality gap of the LP relaxation.

1.1.5 Warnings and Evacuation

In order to study the diffusion of warnings and evacuation messages through a population we need to understand how various psychological, social, and economic factors drive the process. Humans have a layered warning response process that goes through multiple stages [59]. Hence any attempt at realistically modeling this scenario must consider these factors. For this purpose we turn to work carried out in Social sciences literature.

Much research has focused on employing advanced technology in detection and prediction, or optimizing evacuation routes. Far less attention has been paid to early warning dissemination [60, 61, 62, 63]. We conclude from warning response behavior [64] and the decision making involved [65, 66, 67] that it is as much social as it is about communication. The warning sequence process model [63, 68, 69] posits that there are six phases extending from disseminating the warning to the protective action. Social science theory suggests that peoples response to warnings depends on the social, economic, demographic and physical/technological environment, [59, 60, 62, 70, 71] as well as message content [62, 63, 66, 67, 68, 69]. People may seek additional information and confirmation through observation and direct contact [63, 64, 68, 69]. Thus, special care should be taken to incorporate these effects in any general model for diffusion of warnings. The warning time distribution has two components: the official broadcast component [67, 72, 73] and the information contagion (diffusion) component [68, 69, 71, 72]. There has been considerable research in estimating warning time distributions by modeling the warning network, however the contagion component is little understood [67, 70, 72, 74], and needs to be better understood for protective agencies to fully estimate warning times [61, 62]. Any tool that want to analyze warning time distribution must incorporate the effects of inhomogeneities such as social groups, ethnic groups, differential trust in and access to warning systems [68, 69, 74, 75, 76].

1.1.6 Selfish Routing

Routing essentially is a problem of getting from point A to point B on a network while traversing its edges. It is one of the most basic operation performed on a network. There may be costs associated with traversing edges, for example the edges need payment for traversing or they may have delay associated with them. Consequently, there may also be several types of objectives, such as getting from point A to B along a path with the least delay or the least cost etc., that need to be achieved efficiently. When the act of routing is under the control of a single central entity, achieving a given objective becomes a problem of optimization. In many scenarios though, there does not exist a single central entity. Instead there are many players, each possible of taking one from a set of actions while desiring to minimize their own cost. This naturally gives rise to a game, the most general form of which is referred to as the Selfish Routing game. The study of equilibria in selfish routing games can help us quantify loss of social welfare due to selfish, uncoordinated behavior in networks. The Price of Anarchy quantifies the worst possible loss of social welfare and is defined as the ratio between the optimal social welfare possible and the lowest social welfare obtained by any Nash equilibrium solution. Similarly the Price of Stability quantifies the lowest possible loss of social welfare and is defined as the ratio between the optimal social welfare possible and the highest social welfare obtained by any Nash equilibrium solution. We now take a brief look at some literature in this area.

Although routing games are classified in many ways, for example atomic [77] and non-atomic routing [78], for our purpose, we will classify them into *source routing* and *next-hop* routing. Selfish routing and pricing games have been studied in many contexts (see e.g., [79, 80, 81, 82, 83, 84, 85, 86, 87], and the many references in [88, 89]). Most of the work in this area has been done using source routing. Here, the source node of the traffic determines its entire route from the source to the destination. In essence, the entire route is bought/chosen/decided before the traffic is sent along the path. In one such model, [90, 91] consider a game where there are two sets of players. One set of players own edges of the network (edges have finite capacities) and sell capacity to players of the second set. The second set of players obtain utility for routing a unit amount of flow from their source to destination. Hence the second set of players buy capacity on edges along the route if it is profitable to do so. They characterize the price of anarchy and price of stability with respect to two different social welfare objectives since there are two sets of players. One objective is to maximize the profit for the first set of players, i.e. the sellers of capacity. The second is with respect to maximizing the utility of the second set of players who are akin to consumers.

Inter-Domain Routing on the Internet Over the last two decades, there has been a significant amount of research in traffic engineering issues. However, most of this work is on optimizing routes and traffic flows within a *single* domain (see [4, 5, 6] for some of the prominent solutions approaches), towards attaining high throughput or utilization in the network and/or minimize congestion in the most overloaded links. For inter-domain routing/traffic engineering, much literature has been devoted to instability issues of BGP, and the choice of stable routes for path-vector routing based protocols like BGP (see [92, 93, 94, 95] for examples). Research in inter-domain traffic engineering has gained momentum over the last decade, although the bulk of this work has been devoted to inter-domain traffic engineering solutions in the context of BGP, through intelligent use of some of its parameters/flexibilities. [96, 97] discuss methods for doing inter-domain traffic engineering by careful control of BGP route advertisements, while [98] discuss how the BGP AS-Path attribute can be manipulated for that purpose. General guidelines for traffic engineering within the context of BGP are discussed in [99]. [100] discusses an extension to BGP to enable multi-path inter-domain routing. A cooperative optimization-based approach to inter-domain traffic engineering based on dual decomposition and Nash bargaining is described in [101]. A symbiotic optimization based inter-domain traffic engineering approach is discussed in [102]. The above line of work on inter-domain traffic engineering does not take into account strategic pricing by different ISPs/domains or study the inter-domain traffic engineering question from a game-theoretic perspective. A related work [79] studies flow equilibrium efficiency with optimal routing at the intra-domain level and selfish routing across domains; the pricing competition question across domains (ISPs) is however not considered in that work.

There is a growing body of literature on strategic pricing of bandwidth/ flow/ services over the Internet, although most of the existing work either uses restrictive network topology models, and/or focuses on source routing and pricing game models [89]. As mentioned before, in these models, the source node of the traffic determines its entire route from the source to the destination. The efficiency of selfish flow routing by users is considered in [87]. The flow and price control leader-follower game between a set of users (follower) and an ISP (leader) – which controls a link, or a set of tandem links – is studied in [103, 104]. Flow and pricing competition in single-source single-sink parallel-serial networks, where the network consists of a set of parallel links (paths) controlled by profit-maximizing ISP(s), and selfish users control the flows on these paths in response to the path prices, is studied in [84, 105, 106].

If studied for general network topologies and multiple sources, this problem is the same as the one in [90]. [107, 108] studies the question of pricing/tolling of network edges (links) towards realizing efficient Nash flow equilibria, but does not address the issue of price competition between providers. For a single profit-maximizing ISP, equilibrium efficiency questions are studied in [109].

Next-hop Routing The destination-based next-hop model is in sharp contrast with the source routing and pricing models considered in this prior literature, and provides a better representation of the inter-domain routing protocols and pricing practices in the current Internet. Source routing requires knowledge of the entire path at the source node, and this practical limitation has restricted the use of source routing in the Internet, while next-hop routing involves decision making by agents that is much more local and distributed. Even though BGP determines this next hop based on information on the entire AS-level path, the benefits of making it strictly next-hop have been argued recently [110]. Furthermore, traffic flow and service pricing negotiations in the Internet occur at the inter-domain level, between an ISP and its neighboring ISPs, and have a customer/provider/peering relationship [111]. Next-hop routing and pricing model also closely captures the Path Vector Contract Switching framework proposed for inter-domain routing in the future Internet [9], where neighboring ISPs establish contracts (on the amount of flow and its pricing) at their ingress/egress nodes towards forwarding traffic for a specific destination.

Next-hop routing and pricing models have been considered recently in [3] and [112]. Papadimitriou and Valiant [3], define a next hop routing model where players are edges of a network, and argue for the importance of next-hop routing models. The strategy of players in their game is the following: players charge neighbors for processing and forwarding flow. Their model has edges with (linear) latencies, and deals only with a single source with a fixed demand. [83] considers next-hop inter-domain routing as in BGP, and focuses on the issue of incentive-compatibility while [86] looks at the incentive-compatibility of best-response dynamics in BGP routing.

1.2 Our Contribution

Having taken a brief look at the literature, we now turn our attention towards our contributions to these areas. In this section we provide a short description of the problems we have addressed as well as a summary of the results achieved.

1.2.1 The Firefighter Problem

As described in the Section 1.1.4, there has been a significant amount of work on the Firefighter problem. In our work, we focus specifically on inhibiting the spread of an epidemic or an idea by using vaccination. Building on the work of [47] and others [48, 46], we assume that we know the network's topology, and provide worst-case guarantees over all possible networks. Earlier works consider *prophylactic vaccinations,* where the goal is to vaccinate parts of the graph so that once the epidemic begins, the destruction caused by it will be limited. The problem we consider, however, considers the case where the infection has already begun, and we must attempt to minimize its effect.

The model and the Firefighter problem. We model our network of agents as a graph $G = (V, E)$ where vertices correspond to agents, and an edge $e = (u, v)$ represents contact between the two agents. The above is arguably a simplistic model, however, as we see below, even this model leads to many interesting questions and is the main focus of our work.

In this problem the infection/fire starts at a given node node s (or a set of nodes) at time $\tau = 0$. At every subsequent time step, the infection/fire deterministically spreads to all nodes that have an already infected neighbor. To stop the infection, we are allowed to vaccinate/defend at most B nodes per time-step, where B is a budget representing how much we are able to affect the network in a single time-step. A vaccinated node can no longer contract the infection, and therefore cannot pass it on to others. Once infected or vaccinated the vertex remains so for the rest of the time. The process comes to an end when the infection can no longer spread.

We consider two separate objectives in our work. The first objective, which we

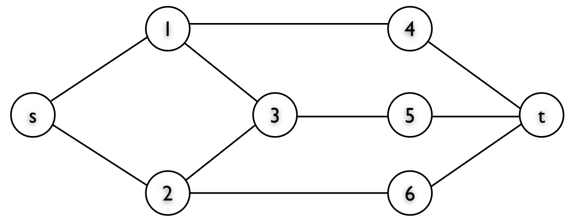


Figure 1.1: This example shows that sometimes vaccinating nodes far away from the infection is the only way to save all the required nodes.

call MaxSave, is to maximize the number of non-infected nodes in the end, when we are given a fixed budget B. The second objective, which we call MinBudget, is to minimize the budget B needed per time instant in order to save a given set of nodes, $T \subseteq V$.

It is not hard to see that in the end, the set of vaccinated nodes form a vertex cut between the set of infected nodes and set of non-infected nodes. However, unlike previous works, such as [47], which examine the *static* problem of vaccinating a 'cut' before the infection has started spreading, we need to find the "best" *cut over time* (where best depends on the considered objective). This temporal nature of our optimization problem makes it significantly different and more challenging than the non-temporal versions.

We also consider a variant of this model, where the vaccination is also a process that spreads through the network. In this "spreading vaccination" model, if a node v is adjacent to a vaccinated node, then v itself becomes vaccinated during the next time-step (unless it is already infected). In the case of ideas propagating through a social network, this represents the fact that an antidote to a harmful idea is often another idea, which can be just as infectious. In disease propagation, this represents the fact that vaccines can be infectious as well, since they are often an attenuated version of the actual disease. We consider the above two objectives in this model as well.

To gain some intuition about this problem, consider the example shown in Figure 1.1 using the non-spreading model of vaccination.

Consider the MinBudget objective for this example. The infection begins at node s, and the goal is to find the smallest number B of nodes that need to be vaccinated at every time step so that we can save the node t, which we assume cannot itself be vaccinated. If we were only allowed to cut nodes during the first time-step, this would be equivalent to the minimum s-t node-cut problem. The temporal nature of the problem, however, complicates matters: intuitively, the tradeoff is between vaccinating a small set of nodes close to the infection source early, or spreading out (over time) the vaccination of a larger set of nodes which are farther away from the source.

For instance, in the above example, a minimum s-t node-cut is $\{1, 2\}$, which requires $B = 2$. However, there *is* a solution to the above problem with $B = 1$, but the final set of vaccinated nodes does not form a minimum s-t node-cut. One such solution is to vaccinate vertices 4, 6, and 5 at time steps 1, 2, and 3 respectively, leading to the final set of vaccinated nodes being $\{4, 5, 6\}$ which is not a minimum cut. In fact, it is not hard to come up with examples where the optimal value of B is much smaller than the size of a minimum node s-t cut *and* the final set of vaccinated nodes is much larger than the size of a minimum node s-t cut (e.g., take a graph where s has k neighbours, each of which is connected to t via k long internally node-disjoint paths). Thus, this "cuts over time" problem is quite different from the classical min-cut problem, and in fact is known to be NP-hard (even when the graph is a tree!) [113].

Our Results. In Section 2.2 of Chapter 2, we consider the model of spreading vaccinations. In general, our results show that this model is more tractable than the model with non-spreading vaccinations. For MaxSave we show that this problem reduces to maximizing a submodular function with a matroid constraint. Therefore a simple greedy algorithm provides a 2-approximation, and a recent result of [29] lets us prove a $e/(e-1)$ factor approximation. Using this, for MinBudget we give a $\ln n$ approximation algorithm, and show that both of our approximation ratios are tight, by showing a set-cover hardness.

The non-spreading model, on the other hand, does not yield itself to good approximation algorithms. In fact, we show in Section 2.3 that it is NP-hard to approximate MaxSave in general graphs to a factor of $n^{(1-\epsilon)}$, for any $\epsilon > 0$. On the other hand, for MinBudget, we give a simple $O(\sqrt{n})$ factor approximation algorithm. For the special class of directed layered networks, we get an improved approximation factor of $H(\ell) = 1 + 1/2 + \cdots + 1/\ell$, where ℓ is the number of layers in the network. In fact, our result shows that the integrality gap of a natural *linear-programming* (LP) relaxation of the problem is $O\big(H(\ell)\big)$. Complementing this, we give an example of a directed ℓ-layered graph showing that the integrality gap is of our LP is also $\Omega(\log n)$. Both our algorithms are combinatorial, and the LP is used only in the analysis. We remark here that the hardness of MinBudget in the spreading model does not carry over to the non-spreading model and as of yet we cannot rule out a constant factor approximation. Table 3.1 summarizes our results.

Table 1.1: The summary of our approximation results. *appx* stands for approximation, *hard* stands for hard to approximate under complexity assumptions.

	Spreading	**Non-Spreading**
MaxSave:	$(1-1/e)$ appx	$n^{(1-\epsilon)}$-hard for $\epsilon > 0$
MinBudget:	$\ln n$ appx $(1-o(1)) \ln n$-hard	General: $2\sqrt{n}$-appx Directed ℓ-Layered Graph: $H(\ell)$-appx

1.2.2 Diffusion in Complex Networks

We develop efficient novel algorithm for targeted diffusion for a general diffusion model that is applicable to a wide variety of diffusions. There exists significant research on targeted diffusion. Much of this research studies models of diffusion which satisfy properties such as “monotonicity” and “sub-modularity” which guarantee that classes of greedy heuristics for the targeting will work well. A general model of diffusion which allows nodes to leave the network when they act upon the diffusion is neither monotonic nor sub-modular, and so these greedy heuristics may not work very well, and other types of heuristics are needed. These heuristics have been developed based on the general model itself. Further, these heuristics are not efficient. We introduce a new paradigm for developing heuristics for the problems of optimal targeted diffusion. Our approach works as follows. First we develop an appropriate simplification of the general model (with certain tunable knobs) for which we can develop efficient, near-optimal targeting algorithms. In contrast to the prior work, we develop the targeting algorithms for a different, more efficient model, not the original general model. The key aspect of our approach is that the simpler model has tunable parameters so that the targeting can be optimized with respect to these parameters to ultimately obtain a good seed set for the true general model. In our results, we illustrate the merit of this new approach in the context of evacuation warnings.

In Section 3.1 we describe a general model of diffusion that can be applied to many real scenarios. Since we specifically look at the spread of evacuation warnings through a community, the diffusing entity here is news or information about the impending evacuation. Nodes in this model have two threshold values for the amount of information they posses. Also, nodes can be in one of four states depending on the amount of information they posses as compared to their threshold values. The action taken by nodes depends on their state. This general model has several parameters that control he manner in which information diffuses through the network as well as its how it is assimilated at the nodes. It also allows for information message passing to be probabilistic and for nodes to leave the network. Thus, the general model provides sufficient control parameters to make it suitable for a wide range of areas.

The objective of this work is to find a set of nodes to be seeded such that spread of evacuation news is maximized. We show in Section 3.2 that this general model of diffusion does not posses desirable and intuitive properties such a submodularity or even monotonicity, which are helpful for theoretical analysis. We then show that this general model can be simplified in way that these properties hold while still retaining some of its original features. In Section 3.3 we describe our algorithm which, given an instance of the general model, extracts the related instance of the simplified model. It then applies greedy heuristics to obtain a near optimal seed set for the simplified instance and then map the seed set back to the general instance.

In Section 3.4 we discuss design of experimental simulations used to compare our projected greedy heuristic with other simple and local heuristics like selecting seed nodes randomly and selecting nodes with high degree. Finally, in Section 3.5, we describe the results of our simulations which show that our heuristic consistently outperforms other simple heuristics. The novelty in our technique is that instead of running heuristics directly on the complex diffusion model, we run near-optimal heuristics on an appropriately simplified model, and then extend the result to the general model. By doing this we get an efficient running time and a near-optimal (in simulation) seed set. The challenge we address is how to determine appropriate simplifications that are tractable but yield similar behavior to the general model.

1.2.3 Strategic Pricing in Next-hop Inter-domain Routing

We introduce and analyze (both theoretically and in simulation) a model of dynamic strategic pricing between ISPs, which is also closely integrated to the question of inter-domain traffic engineering. We model the Internet as an interconnected system of ISPs. We represent an ISP as a *cloud*, that is composed of a set of nodes (routers) and capacitated links connected into an arbitrary topology; a subset of these nodes represents the ISP's ingress and egress nodes. At an ingress node, an ISP offers to its neighboring (upstream) ISP a price for forwarding traffic to a destination, and accepts the traffic to be forwarded. The ISP also derives utility for forwarding the traffic generated within its own network, and must decide how to forward (split) its traffic (that is offered to it by upstream neighbors, or generated inside its network) among its downstream neighbors (egress points). ISPs are assumed non-cooperative in nature, and pricing and traffic forwarding decisions are made such that they maximize the individual ISP's profit.

For this strategic pricing game, we obtain theoretical results for the case when the network is *well-provisioned*. By *well-provisioned*, we mean that bottlenecks only occur on inter-ISP links, and thus the inside of every cloud has enough capacity to support whatever traffic may be required (see Section 4.1 for a precise definition). A well-provisioned ISP can be modeled as a single vertex in a graph since the ISP is able to route traffic internally without any restrictions. Following is a brief description of the model for well-provisioned networks.

Model Summary We are given a directed acyclic graph $G = (V, E)$ containing a special sink node t, and edge capacities c_e. As commonly done when analyzing competition in networks [82, 3, 91], we assume that all edges of this DAG, except the ones that are incident on the sink node, have a special non-monopolistic property. For our model, this property essentially ensures that there exists enough capacity so that no node could charge an infinitely high price for forwarding traffic, and yet have other nodes pay this price because they have no alternative. The players of our game are all vertices of G except the sink node. An edge $e = (u, v) \in E$ with capacity c_e denotes that player u has the capacity to send a flow of size c_e to player v. Additionally, every player v has an associated *source utility* λ_v. This means that if a player v sends f_v amount of its own flow (flow originating form vertex v) to the sink t, then the player will obtain a utility of $\lambda_v f_v$. Thus, the player demands are *elastic*, since each player can choose an amount of traffic to send in order to maximize its utility. We consider an extension of this model in Section 4.4 where source utilities are allowed to be non-linear.

Players choose prices on their incoming edges. For every edge $e = (u, v)$ player v chooses a price p_e such that if u sends a flow of size f_e on edge e then u pays an amount $p_e f_e$ to v. Players route flow on outgoing edges such that their cost is minimized, but are obligated to forward all flow that they receive. Finally, the utility of a player is the total amount of money it receives from upstream players and the utility obtained by sending its own flow minus the amount of money paid to the downstream nodes.

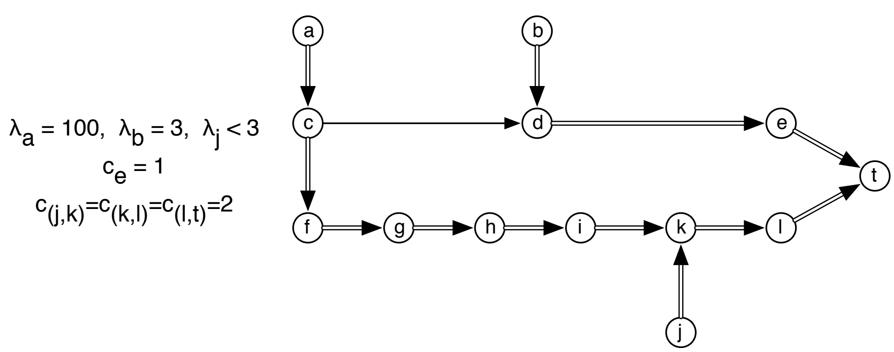


Figure 1.2: An example illustrating our model, and the "long-range" effects that node utilities can have on edge prices.

Example. To illustrate some key consequences of our model, consider the example in Figure 1.2. The utility values of all nodes except the ones mentioned are 0. Edge (j,k), (k,l), (l,t) have capacity 2 and all other edges have capacity 1. Not all the edges in the graph are pictured: for every edge (u,v) shown in the figure that does not satisfy the non-monopolistic property, there also exist (non-pictured) edges (u,w), (w,t) of capacities 1 such that w has a high source utility (say 1000). In any optimal solution edge (u,w) will not have any flow on it whereas (w,t) will be saturated. Now it is not difficult to see that any optimal solution will consist of the flow indicated in Figure 1.2, where the double walled edges are saturated with flow, except edge (j,k) has a flow of size 1 whereas its capacity is 2.

A more careful analysis shows that any Nash equilibrium strategy that yields this optimal flow will have the following edge prices: $p_{(j,k)} = p_{(k,l)} = \lambda_j$, $p_{(b,d)} = 3$. Also $p_{(c,d)} \geq p_{(c,f)} \geq p_{(f,g)} \geq p_{(g,h)} \geq p_{(h,i)} \geq p_{(i,k)} \geq \lambda_j$. This means that the price of edge (c,d) depends on the source utility value of node j, and thus there are "long-range" interactions between source utilities and edge prices.

Results Our main goal involves understanding the properties of stable solutions in this pricing game: specifically we focus on pure Nash equilibrium. We show that in the most general form of our pricing game, Nash equilibrium may not exist but at the same time we obtain several results for *well-provisioned* networks. First, we show that breaking ties appropriately, when identical prices are offered by multiple neighboring ISPs, is crucial: if done correctly this leads to the existence of good Nash equilibria, but if done incorrectly no equilibrium may exist. In Section 4.3.2 we give an efficient algorithm that constructs a Nash equilibrium strategy that is as good as the optimal solution. In other words, we show that the price of stability is 1, and thus it is always possible to implement traffic pricing that maximizes social welfare. This holds for an arbitrary number of sources with elastic demands of heterogenous value. We further give a partial characterization of Nash equilibria with nice structure (which we call *uniform* Nash equilibria), and show how to compute (and the existence of) a Nash equilibrium as good as the optimum centralized solution for the case when the network is *well-provisioned*. We also show in Section

4.3.3 that in case of a single source, under some reasonable behavioral assumptions for the players, the price of anarchy is 1, and in fact player prices at equilibrium are unique.

Until this point, the source utilities λ_v and the allowed price functions p_e were considered to be linear. In Section 4.4, we instead consider the more general case where the source utility can be an arbitrary concave function $\Lambda_v(f_v)$, and the prices can be arbitrary convex functions $\Pi_e(f_e)$. We show that the above results still hold if arbitrary convex prices are allowed, and thus allowing non-linear prices does not impact the quality of equilibrium solutions. On the other hand, if source utilities can be non-linear functions, then we show that pure Nash equilibrium may no longer exist, even for discrete pricing models.

While most of our theoretical results are for well-provisioned networks, our simulations show that our model still behaves extremely well even without this assumption. We observe that simple price dynamics converge to stable solutions in an overwhelming majority of our simulations. Moreover, the solutions that price dynamics converge to are not only stable, but have excellent social welfare (overall utility), often within a few percent of the social optimum.

Our theoretical results on the pricing question for inter-domain forwarding are quite significant, as they show that (at least for well provisioned ISPs) there exists a pricing strategy for inter-domain forwarding that is both *stable* and *efficient*. Adoption of such a pricing strategy would imply that ISPs acting in self-interest will not unilaterally deviate from such pricing. Additionally, such a pricing strategy maximizes aggregate end-to-end throughput (utility), or in other words, makes optimal use of global network resources. Thus use of such a pricing strategy leads to a desirable operating point for the Internet. Our dynamic pricing update policies, that have been observed to quickly converge and result in near-optimal flows (for most topologies), could be used to obtain these prices dynamically in a fully distributed manner. The destination-specific, next-hop nature of our pricing and forwarding framework and policies imply that these can be implemented with minor changes to the path-vector routing framework that constitute the current Internet inter-domain routing practices. Used in this manner, the proposed pricing and flow routing policies could be used for pricing based dynamic inter-domain traffic engineering in the Internet. Our model and results make very loose assumptions on the *intra*-domain routing policy used, and allows a wide variety of intra-domain, packet routing (traffic engineering) policies to be used, which may even differ across domains (ASes or ISPs).

CHAPTER 2
The Firefighter Problem

2.1 Formal description of our model

We are given a directed[1] graph $G = (V, E)$ and a source node s. We assume $|V| = n$ and $|E| = m$. We let the distance between two nodes u and v be the path from u to v of minimum length.

All nodes in the graph can have one of three states: they can be *infected*, *vaccinated*, or *vulnerable*, that is neither vaccinated nor infected. At time $\tau = 0$, all nodes are vulnerable, except node s, which is infected. At each $\tau > 0$, any vulnerable vertex v which is connected to an infected node u, such that $(u, v) \in E$, gets infected at time $\tau + 1$, unless it is vaccinated at time step τ. Infected and vaccinated nodes stay infected and vaccinated respectively. We call a node *saved* if it is either vaccinated or if all paths from any infected node to it contains at least one vaccinated node.

Definition 1 *A vaccination strategy is a set $\Psi \subseteq V \times J$ where V is the set of vertices of graph G and $J = \{1, 2, \ldots |V|\}$. The vertex v is vaccinated at time $\tau \in J$ by the vaccination strategy Ψ if $(v, \tau) \in \Psi$. A vaccination strategy Ψ is valid with respect to budget B, if the following two conditions are satisfied:*

i. if $(v, \tau) \in \Psi$ then v is not infected at time τ,

ii. let $\Psi_\tau = \{(v, \tau) \in \Psi\}$; then $|\Psi_\tau| \leq B \quad for \quad \tau = 1 \ldots |V|$.

The first condition implies we can only vaccinate vulnerable nodes, and the second condition requires us to obey a budget constraint according to which we can only vaccinate a maximum of B nodes at every step.

[1]We use a directed graph since it is more general– an undirected graph is just a directed graph with two arcs per edge.

Portions of this chapter previously appeared as: E. Anshelevich, D. Chakrabarty, A. Hate and C. Swamy, "Approximation Algorithms for the Firefighter Problem: Cuts over Time and Submodularity." *Algorithms and Computation* 5878, 974-983 (2009).

We consider two models of vaccination. In the *Non-spreading Vaccination Model*, vaccinating a vertex simply means that it can no longer be infected. Vaccination of a node doesn't affect the vaccination of other neighboring nodes.

In the *Spreading Vaccination Model*, however, the vaccination spreads to all its neighboring nodes which are still vulnerable, thereby vaccinating them. That is, at time step $\tau > 0$, if a node v is vaccinated and there is a vulnerable node u such that $(v, u) \in E$, then at time $\tau + 1$, the node v also gets vaccinated. Thus, the vaccination also spreads like the infection. Note that there could arise a situation where a vulnerable node is adjacent to both an infected node and a vaccinated node. We will assume that the vaccine prevails over the infection, and in the subsequent time step, the vulnerable node is vaccinated, rather than being infected. This is actually a weak assumption as assuming otherwise doesn't change the quality of our results. In the spreading model, we will say that a node is vaccinated *directly* when it is vaccinated by the vaccination strategy, and it is vaccinated *indirectly* when it is vaccinated by the spread of the vaccine through the network.

The process stops when there are no vulnerable nodes adjacent to an infected node, so the infection cannot spread any further. This must occur before time n, for n being the number of nodes.

Objectives. The main two objectives we consider when developing a vaccination strategy are as follows.

MaxSave(G, B, s, T)
Instance: A rooted graph $(G(V, E), s)$, integer $B \geq 1$ and $T \subseteq V$
Objective: Find a valid vaccination strategy Ψ such that if s is the only infected node at time 0, then at the end of the above process the number of non-infected nodes that belong to T is maximized.

This problem is also referred to as the Firefighter problem in the literature when $T = V$.

MinBudget(G, s, T)

INSTANCE: A rooted graph $(G(V,E),s)$, and $T \subseteq V$

OBJECTIVE: Find a valid vaccination strategy Ψ with minimum possible budget B, such that if s is the only infected node at time 0, then at the end of the above process all nodes in T are saved.

In other words, in MAXSAVE we are interested in saving as many nodes of T as possible given a fixed budget, and in MINBUDGET we are interested in finding the minimum necessary budget to save all nodes in T.

2.2 Spreading Vaccination Model

We achieve tight results in the spreading vaccination model for both the MAXSAVE and MINBUDGET problems. We give a $(1-1/e)$ approximation for the MAXSAVE problem. For the MINBUDGET problem, we use the above approximation to get a $\ln n$-factor approximation. In fact, more generally this reduction (see Theorem 5) implies that a $(1-1/e^{\alpha})$-approximation for MAXSAVE (where $\alpha \geq 0$) yields a $\ln n/\alpha$ approximation guarantee for the MINBUDGET problem. We also show a set cover hardness (i.e., $(1-o(1))\ln n$ hardness) for the MINBUDGET problem. Thus, this hardness result also implies that approximating the MAXSAVE problem to any constant factor $c > (1-1/e)$ is hard. To do so, we use a characterization of the saved vertices given a vaccination strategy. We describe this first.

2.2.1 Characterizing Saved Vertices in the Spreading Model

We start with a useful observation. Let $d(u,v)$ be the shortest distance between the nodes u and v in graph G, and let $N(v,i)$ be the set of all the nodes that are a distance of at most i from v.

Lemma 1 *At time τ, all nodes in the neighborhood $N(s,\tau)$ will either be vaccinated or infected.*

Proof. Let $P(u,v)$ be a shortest path between nodes u and v.

We prove this lemma by induction. The base case for time $\tau = 0$ is true since all nodes in neighborhood $N(s,0)$, which contains only s, are infected. Let us assume that the statement of the lemma is true for all time-steps $\tau = \{1 \dots k\}$ for some $k \geq 0$. Our assumption implies that all nodes in the neighborhood $N(s,k)$ are either infected or vaccinated at time $\tau = k$. Now consider the set of nodes $\Delta = \{v : v \in N(s,k+1) \backslash N(s,k)\}$. Each of these nodes have a neighbor in $N(s,k)$ which are either infected or vaccinated according to the hypothesis. According to the definition of the model, all nodes that are infected\vaccinated, infect\vaccinate their neighboring nodes in the next time-step. Hence all nodes in Δ will be either infected or vaccinated at time-step $k+1$. This proves the inductive hypothesis and hence statement of the lemma. ■

A consequence of the above lemma is that if (u,τ) is in a optimal vaccination strategy, we will have $\tau < d(s,u)$. Henceforth, we will enforce any vaccination strategy to satisfy this property.

We now define a set $\Gamma(v) \subseteq V \times J$ for every node $v \in V$ which will be used to characterize if v is saved by a vaccination strategy Ψ or not. Let

$$\Gamma(v) := \{(u,\tau) | u \in V \text{ and } 0 < \tau \leq d(s,v) - d(u,v)\}.$$

Recall, the tuple (u,τ) represents the direct vaccination of the node u at time τ. The following theorem states that a vertex v is saved by a vaccination strategy Ψ in the spreading model, if and only if the strategy vaccinates some u directly at time τ, such that $(u,\tau) \in \Gamma(v)$. This allows us to get a handle on the structure of Ψ.

Theorem 1 *A node $v \in V$ is saved by the vaccination strategy Ψ if, and only if, $\Psi \cap \Gamma(v) \neq \emptyset$.*

Proof. ***(if:)*** Let $(u,\tau_u) \in \Psi \cap \Gamma(v)$. Let $P(u,v)$ be a path from u to v of length $d(u,v)$. We prove a stronger statement: all the vertices in $P(u,v)$ are saved. Suppose a vertex $w \in P(u,v)$ is infected. Let it be the nearest such vertex to u. Then it must be the case that $d(s,w) < \tau_u + d(u,w)$, for otherwise the vaccination from u would've spread to w. This implies, $d(s,v) \leq d(s,w) + d(w,v) < \tau_u + d(u,w) + d(w,v) = \tau_u + d(u,v)$; the last equality follows since $w \in P(u,v)$. However, this contradicts $(u,\tau_u) \in \Gamma(v)$.

(only if:) Suppose a vertex v is saved by Ψ. v must be vaccinated either directly or indirectly. Either v is vaccinated directly at time τ_v; in this case by

Lemma 1 we must have $0 < \tau_v \leq d(s,v)$ and thus $(v, \tau_v) \in \Gamma(v)$. Or, v is vaccinated indirectly. Let the first vaccination to reach v be from vertex u. Lemma 1 implies that u had to vaccinated at time $0 < \tau_u \leq d(s,u)$. Furthermore, $\tau_u + d(u,v) \leq d(s,v)$, otherwise v would have been infected by s before vaccination reached v from u. Therefore, (u, τ_u) belongs to the set $\Gamma(v)$. ■

2.2.2 The MaxSave Problem

Using Theorem 1, we now show that the MaxSave problem in the spreading vaccination model can be cast as the problem of maximizing a monotone submodular function over a partition matroid constraint.

Recall, given a ground set U, a function $f : 2^U \rightarrow \mathbb{R}_{\geq 0}$ is a monotone submodular function iff

- $f(A) \leq f(B)$ whenever $A \subseteq B \subseteq U$; and
- $f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \, \forall \, A, B \subseteq U$.

A set system $(U, \mathcal{F})$ is a partition matroid if there is a partition of the ground set $U = U_1 \cup \cdots \cup U_k$, and there exists integers $\ell_1, \ldots, \ell_k$ such that $X \subset U$ is in $\mathcal{F}$ if and only if $|X \cap U_i| \leq \ell_i$, for all i. The problem of maximizing a monotone submodular function f over a partition matroid $(U, \mathcal{F})$ is to find a subset $X \in \mathcal{F}$ which maximizes $f(X)$.

The following is known about the problem.

Theorem 2 *There is a deterministic algorithm which achieves a 2-approximation for the problem of maximizing a monotone submodular function over any matroid. ([114]).*
There is a randomized algorithm which achieves a $e/(e-1)$-approximation for the problem of maximizing a monotone submodular function over any matroid. ([29]).

Theorem 3 *In the spreading vaccination model,* MaxSave *is a special case of maximizing a monotone submodular function over a partition matroid constraint.*

Proof. For the sake of expositional simplicity, we consider the MaxSave Problem where $T = V$; the same analysis holds for any $T \subseteq V$.

We define $\mathcal{E}$ as the set of all possible direct vaccination tuples (v, τ) where $v \in V$ and $\tau = 1 \ldots n$. Note that from Lemma 1, we can assume $\tau \leq d(s, v)$ for all $(v, \tau) \in \mathcal{E}$. $\mathcal{E}$ can be partitioned as follows:

$$\mathcal{E} = \bigcup_{\tau=1}^{n} \mathcal{E}_\tau \quad \text{where} \quad \mathcal{E}_\tau = \{(v, \tau) | (v, \tau) \in \mathcal{E}\}.$$

Proposition 1 *A vaccination strategy Ψ is valid if and only if $|\Psi \cap \mathcal{E}_\tau| \leq B$, for all τ. That is, the collection of vaccination strategies forms a partition matroid over $\mathcal{E}$.*

Proof. A valid vaccination strategy should directly vaccinate at most B nodes at each time-step. Thus, if Ψ is valid, then it satisfies the condition of the proposition. On the other hand, since we assume $\tau \leq d(s, v)$ for all $(v, \tau) \in \mathcal{E}$, each tuple (v, τ) is a valid direct vaccination option, and the partition condition satisfies the budget constraint. ■

We now show that the optimal Ψ, the vaccination strategy which maximizes the number of saved nodes, also maximizes a certain submodular function over $\mathcal{E}$. This function will be a *coverage* function. That is, for each element $(u, \tau) \in \mathcal{E}$, we define a set $S_{(u,\tau)}$, and the function $f : 2^{\mathcal{E}} \to \mathbb{R}_{\geq 0}$ is given as:

$$f(\Psi) := \left| \bigcup_{(u,\tau) \in \Psi} S_{(v,\tau)} \right|.$$

Such a function is known to be submodular (see [114], for instance).

We now define the sets $S_{(u,\tau)}$ and show that the number of nodes saved by a vaccination strategy Ψ, is precisely $f(\Psi)$. This completes the proof. These sets will be derived using Theorem 1. Recall, we proved that a node v is saved iff $\Gamma(v) \cap \Psi$ is nonempty, where $\Gamma(v) = \{(u, \tau) | u \in V \,\text{and}\, 0 < \tau \leq d(s, v) - d(u, v)\}$ Define

$$S_{(u,\tau)} := \{v \in V : (u, \tau) \in \Gamma(v)\}.$$

Thus a node v is saved if and only if $v \in S_{(u,\tau)}$ for some (u, τ) in Ψ. Therefore, the number of nodes saved by Ψ is precisely $f(\Psi)$. ■

Thus using Theorem 2 and 3, we get constant factor approximations for MaxSave in the spreading vaccination model.

It should be noted here that the same $(1 - 1/e)$ approximation can also be obtained by applying a randomized rounding technique similar to [56] to a modified version of the MaxSave problem. We believe, however, that modeling the problem using partition matroids and submodular functions has great advantages. The algorithms produced in this way are combinatorial and far more efficient than randomized rounding, since they do not require solving a linear program. It also allows the use of a much simpler greedy algorithm as mentioned above, which has the advantage of being deterministic and efficient. Finally, the connection between submodular functions and MaxSave provides insight into the intrinsic structure of the problem, and allows for future results in the field of submodular functions to become applicable to MaxSave.

2.2.3 The MinBudget Problem

We first show that in the spreading model, the MinBudget problem on directed graphs is as hard as set cover. This implies a logarithmic inapproximability. Subsequently, we show a $\ln n$ factor approximation.

Theorem 4 *In the spreading vaccination model, the* MinBudget *problem is as hard as set cover.*

Proof. Consider an instance of Set Cover: a collection $\mathcal{C}$ of finite subsets of U. Let $|\mathcal{C}| = k$ and $|U| = n$. An instance of the MinBudget problem can be constructed as follows:

- Construct node s which will be the root node.
- For each subset $S \in \mathcal{C}$, construct a node v_S with a directed edge (s, v_S).
- For each element $e \in U$, construct k nodes $v_{e_1} \dots v_{e_k}$ such that there is a directed edge (v_S, v_{e_i}) for $i = 1 \dots k$ if $e \in S$.

This graph forms the input for the MinBudget problem with $T = \{v_{e_1} \dots v_{e_k} \, \forall e \in U\}$. Each element of U is represented by k nodes, each of them connected by an edge coming from the nodes that represent the sets to which the elements belong. The following proposition proves the theorem.

Proposition 2 *There is a set cover of size B if and only if there is a vaccination strategy immunizing at most B nodes per time step.*

Proof. ***(only if:)*** Suppose there is a set cover $\mathcal{C}' \subseteq \mathcal{C}$ of size B. Consider the vaccination strategy of immunizing all v_S for $S \in \mathcal{C}'$ in the first time step. Since $\mathcal{C}'$ is a set cover, all vertices v_{e_i} get indirectly vaccinated on the second time step.

(if:) We may assume $B < k$, for otherwise the proposition is trivial. We may assume that in any optimal vaccination strategy, no vertex v_{e_i} is directly vaccinated in the first time step. This is because it is better to vaccinate v_S for any S containing e, as that would lead to spread of vaccination to other v_{e_j}. Let $\mathcal{C}' := \{S : v_S$ is immunized on the first time step $\}$. Note that $|\mathcal{C}'| \leq B$. We claim that this is a set cover. If not, there is an element e not in any set in $\mathcal{C}'$. But then, on the second time step, the infection reaches all the k copies of this element. Since $B < k$, one cannot directly vaccinate all these copies, and hence we reach a contradiction. ∎

Corollary 1 *There is no $(1-o(1)) \ln n$ approximation for the* MinBudget *problem in the spreading model, unless $NP \subseteq DTIME(n^{\mathtt{polylog}(n)})$.*

The above follows from the result of Feige [115]. ∎

Theorem 5 *There is a polynomial time $\ln n$-factor approximation algorithm for* MinBudget *in the spreading model.*

Proof. Such an algorithm can be found by using the $O(1)$ approximation algorithms for the MaxSave problem, described in the previous section. We guess the optimum B of the MinBudget problem by doing a binary search. Since there is a vaccination strategy with budget B which saves all of T, we can save at least

$(1 - 1/e)$ fraction of the vertices in T by running the algorithm described in Theorem 2. Let T_1 be the set of nodes in T not saved. Repeating the argument, there is a strategy with budget B, which saves $(1 - 1/e)$ fraction of the nodes of T_1. This implies, by running the two strategies together, there is a budget $2B$ vaccination strategy, which saves all but a $1/e^2$ fraction of nodes of T. Further, we can compute a vaccination strategy with budget $(\ln n) \times B$, which saves all the nodes of T. ■

2.2.4 Other Results and Extensions

1. We consider another objective which is to minimize the *total* number of nodes directly vaccinated in order to save a subset of nodes. This objective could be important when there is not only a limitation on the number of direct vaccinations at each time-step, but also on the total number of vaccinations that can performed. We can obtain a $(\log n, \log n)$ bi-criteria approximation that would directly vaccinate at most $O(\log n)$ times the optimal number of nodes over all time-steps while vaccinating at most $O(\log n \cdot B)$ nodes at each time-step by performing randomized rounding.

This problem can be represented as an Integer Program (IP). For representing the tuple (v, τ) (immunizing vertex v at time τ), we define a variable $x_v^\tau \, \forall \, v \in V, \tau = 1 \ldots |V|$. The value of $x_v^\tau = 1$ if node v is directly immunized at time τ and 0 otherwise. Also T is the set of nodes that has to be saved. The IP is as follows,

$$\text{Minimize} \quad \sum_{\tau=1}^{n} \sum_{v \in V} x_v^\tau$$

$$\text{Subject to:} \quad x_v^\tau \in \{0, 1\} \qquad \forall \, v \in V, \tau = 1 \ldots |V|,$$

$$\sum_{v \in V} x_v^\tau \leq B \qquad \tau = 1 \ldots |V|, \tag{2.1}$$

$$\sum_{(u,\tau) \in \Gamma(v)} x_u^t \geq 1 \qquad \forall v \in T. \tag{2.2}$$

Condition (2.1) refers to the fact that only B nodes can be immunized at each time-step and condition (2.2) refers to the fact that the immunization strategy has to select at least one tuple from $\Gamma(v)$ for all $v \in T$. By relaxing the condition on x_v^τ such that $0 \leq x_v^\tau \leq 1$, we get a linear program (LP) whose solution can be found in polynomial time.

Once we obtain the solution to the LP, say x_v^{t*}, we apply the following randomized rounding technique to the get the immunization strategy Ψ:

- Add element (v, τ) to Ψ with probability $x_v^{\tau *}$.
- Repeat the above step $d = O(\log n)$ times. (Here $n = |V|$)

It becomes necessary to repeat the selection process so that all the nodes in set T are saved with high probability.

Analysis: The analysis of this algorithm is similar to the analysis of the randomized rounding algorithm for Set Cover. The main difference is that while in Set Cover the goal is to guarantee that the total number of sets chosen is small w.h.p., in our case we must guarantee that with high probability, there is no time-step where the number of nodes immunized is large. For this we require a slightly stronger analysis.

We define random variables X_τ as the number of nodes directly vaccinated at time τ. We can also define a random variable X that gives the total number of nodes vaccinated directly. Consider the expected value of X:

$$
\begin{aligned}
E[X] &= \sum_{\tau=1}^{n} E[X_\tau] \quad \ldots \text{(Linearity of expectation)} \\
&\leq \sum_{\tau=1}^{n} d \log n \cdot \sum_{v \in V} x_v^{\tau *} \\
&\leq d \log n \cdot \sum_{\tau=1}^{n} \sum_{v \in V} x_v^{\tau *} \\
&\leq d \log n \cdot \mathrm{OPT}.
\end{aligned}
$$

By using Markov's inequality, we know that X cannot be very large as compared to $d \log n \cdot \mathrm{OPT}$ with high probability. Rest of the analysis remains is similar to that of Set Cover.

This way we obtain a vaccination strategy Ψ that saves all nodes in T with high probability, vaccinates at most $O(\log n \cdot B)$ nodes at each time-step, *and* vaccinates at most $O(\log n)$ times the optimal number of nodes in total over all time-steps.

2. We also consider a requirement that only nodes that have infected neighbors can be vaccinated directly. This is important because vaccinating a node that is not directly a neighbor of an infected node means that the strategy is guaranteeing that a certain node will be infected when it could have been saved. When nodes represent people and infection a fatal disease, saving people who are next in the line of fire, even though it is not optimal, can become a necessity.

In such cases the techniques discussed in previous sections can be applied and we get similar results.

By Lemma 1 we can infer that at time τ all vulnerable nodes that have infected neighbors lie at distance of τ from s. Since we know that a node v will be directly vaccinated at time $\tau = d(s, v)$ or not at all. For the MaxSave and MinBudget problems, this additional constraint on choosing a valid vaccination strategy preserves the partition matroid property of the set of strategies. Hence the theorems used to prove the approximation guarantees still hold.

3. Consider the weighted version of the MaxSave problem where all nodes have weights $w_v \in \mathbb{Z}^+$ associated with them that signify their importance. Now the objective would be to save a set of nodes with maximum total weight. The results for MaxSave apply to this version as well.

2.3 Non-Spreading Vaccination Model

The non-spreading model is considerably more difficult to reason about than the spreading model. For instance, the structural lemma, Lemma 1 (or any simple modification of it), is no longer true in this model. Thus, it is difficult to characterize the set of nodes which are saved by a vaccination strategy. We show that the MaxSave problem is hard to approximate in this model to any nontrivial factor. Nonetheless, we achieve a $O(\sqrt{n})$ for the MinBudget problem, which we improve to $O(\log n)$ for directed layered networks.

2.3.1 The MaxSave Problem

Unlike the spreading vaccination model where we got constant factor approximation algorithms for the MaxSave problem, we show in our next theorem that it is NP hard to get an approximation factor of $n^{1-\epsilon}$, for any constant $\epsilon > 0$. In fact, the theorem holds even when the budget parameter B is set to 1.

Theorem 6 *For any $\epsilon > 0$, it is NP hard to obtain an $n^{1-\epsilon}$ factor approximation to the* MaxSave $(G, 1, s, T)$ *problem.*

Proof. We introduce an auxiliary problem, Save-t, whose input is a graph G, and two specified nodes s and t. The problem is to decide if t can be saved from infection spreading from s, by vaccinating 1 node (other than t) at each time step. The NP-completeness of this problem follows from known NP-completeness of the MinBudget problem[113] (let the neighbors of t be T for the MinBudget problem). The hardness of MaxSave is obtained by replacing t with a polynomially large set.

Given an instance $(G = (V, E), s, t)$ of Save-t with $|V| = n$, we construct an instance $(G' = (V', E'), 1, s, T')$ of MaxSave as follows. The vertex set V' contains a copy of each vertex in $V \setminus t$ and an additional set T of n^β vertices, with the value of β to be fixed later. We abuse notation and call the set of copies of $V \setminus t$, as V as well. The edge set E' consists of all original edges $\{(u, v) : (u, v) \in E\}$ of G not incident on t, and the set of edges $\{(u, v) : u \text{ is incident to } t \text{ in } G, \text{ and } v \in T\}$. Thus, G' is exactly G, except we take n^β copies of t, which we call T. The target set T' in the MaxSave problem is the set V', that is, we want to save all the vertices of G'. We let $N = |V'| = n^\beta + n - 1$.

We then give a gap introducing reduction from the Save-t problem to the MaxSave problem such that if there exists any n^α approximation for the MaxSave problem then we can solve the Save-t problem in polynomial time. Consider an instance of the Save-t problem $\{G(V, E), s, t\}$. Let $n = |V|$. We construct an instance of the MaxSave problem $\{\bar{G}(\bar{V}, \bar{E}), \bar{s}, 1, \bar{V}\}$ as follows:

i. Let $\beta \in \mathbb{Z}_+$ such that $\beta > \max\{\frac{\ln 2n^2}{(1-\alpha)\ln n}, 3\}$.

ii. Let $N(t)$ be the set of nodes that are neighbors of t. We construct $\bar{G}(\bar{V}, \bar{E})$ as follows:

a. $\bar{V} = V \cup T$ such that $|T| = n^{\beta} - n$,

b. $\bar{E} = E \cup \{(u, v) : u \in N(t) \,\text{and}\, v \in T\}$.

iii. $\bar{s} = s$.

So we essentially construct an instance of MaxSave by creating multiple copies of the sink node of the Save-t problem. Let OPT be the best possible solution for the Save-t instance and $\overline{OPT}$ be the optimal solution to the corresponding MaxSave instance. We state the following lemmas.

Lemma 2 *If OPT can save t in the instance of* Save-*t, then $\overline{OPT}$ can save at least $n^{\beta} - n$ nodes in the corresponding instance of* MaxSave.

Proof. If OPT can save t then $\overline{OPT}$ will be able to save all nodes that belong to T in $\bar{G}$. There are $n^{\beta} - n$ such nodes. ■

Lemma 3 *If OPT cannot save t in the instance of* Save-*t, then $\overline{OPT}$ will be able to save at most $2n$ nodes in the corresponding instance of* MaxSave.

Proof. If OPT cannot save t in G then it is clear that $\overline{OPT}$ will not be able to cut off the nodes of T in $\bar{G}$ from the infection since both can vaccinate only one node at each time-step. The infection will reach the nodes of T in at most n time-steps. The only way $\overline{OPT}$ can save nodes of T is by vaccinating them. In n time-steps, the optimal algorithm can save at most n nodes in T. Even if the optimal algorithm manages to save all other $\bar{V} \backslash T$ nodes ($|\bar{V} \backslash T| = n$), it will be able to save at most $n + n = 2n$ nodes. ■

Now consider the condition for choosing the value β:

$$
\begin{aligned}
&\beta > \frac{\ln(2n^2)}{(1-\alpha)\ln n} \\
&\Rightarrow \beta \cdot (1-\alpha)\ln n > \ln(2n^2) \\
&\Rightarrow \beta \ln n - \alpha\beta \ln n > \ln 2 + 2\ln n \\
&\Rightarrow (\beta - 2)\ln n > \ln 2 + \alpha\beta \ln n \\
&\Rightarrow \ln(n^{\beta-2}) > \ln(2n^{\alpha\beta})
\end{aligned}
$$

$$\Rightarrow n^{\beta-2} > 2n^{\alpha\beta}. \tag{2.3}$$

Also we can safely assume that $n \geq 2$. This leads to the following:

$$
\begin{aligned}
&\Rightarrow n - 1 \geq 1 \\
&\Rightarrow n^{\beta-2}(n-1) > 1 \quad \ldots (\because n \geq 2,\, \beta > 3) \\
&\Rightarrow n^{\beta-1} - 1 > n^{\beta-2}.
\end{aligned} \tag{2.4}
$$

Using inequalities (2.3) and (2.4), it follows that:

$$
\begin{aligned}
&n^{\beta-1} - 1 > 2n^{\alpha\beta} \\
&\Rightarrow n^{\beta} - n > n^{\alpha\beta} 2n.
\end{aligned} \tag{2.5}
$$

Suppose to the contrary that there exists an algorithm that gives a n^{α} approximation for the MaxSave problem. This means that the algorithm should be able to give a $n^{\alpha\beta}$ approximation for the above constructed instance of MaxSave since $|\bar{V}| = n^{\beta}$. Let the value obtained by running this algorithm on the above instance of MaxSave be ϕ. We consider two cases for the value of ϕ.

1. $\phi \leq 2n$; Since ϕ is an $n^{\alpha\beta}$ approximation, we get the following inequalities,

$$\overline{OPT} \leq n^{\alpha\beta}\phi$$
$$\overline{OPT} \leq n^{\alpha\beta}2n$$
$$\overline{OPT} < n^{\beta} - n. \qquad \ldots\text{(from inequality (2.5))}$$

   According to Lemma 2 if OPT could save t then $\overline{OPT}$ can never be lower than $n^{\beta} - n$. Hence we can infer that OPT in the original SAVE-t instance does not save t.

2. $2n < \phi$; Since ϕ is an approximation, the optimal solution is at least as big as ϕ. Hence, $\overline{OPT} > 2n$. According to Lemma 3, this precludes the possibility that OPT in the original SAVE-t instance does not save t. Hence, we can infer that t can be saved in the original SAVE-t instance.

Therefore we see that if there exists an n^{α} approximation for MAXSAVE, then we can solve the SAVE-t decision problem which is NP-complete in polynomial time. Hence the MAXSAVE problem is n^{α} inapproximable unless P=NP. ■

2.3.2 The MINBUDGET Problem

Recall that we need to save *all* the nodes in a set T with the minimum number of vaccinations required per time step. To simplify notation, we consider the following equivalent problem: we add a new node t with edges from all nodes in T to t, and consider the problem of saving t with minimum budget *under the additional constraint that t itself cannot be vaccinated*. We call s the source and t the sink. Let $\mathcal{P}$ denote the collection of all s-t paths.

In the remainder of the section, we give two approximation algorithms. First is a $2\sqrt{n}$ factor algorithm for the problem; contrast this with the hardness of approximation for the MAXSAVE version. In fact, we show that the minimum s-t cut achieves this factor. We also give an improved $O(\log n)$ algorithm for the case of directed layered graphs. The analysis of our algorithm utilizes a natural LP relaxation for the problem, and as a corollary we obtain that the integrality gap of this

LP is $O(\log n)$. We also give an example that proves a matching lower bound of $\Omega(\log n)$ on the integrality gap of this LP relaxation.

We start with the $O(\sqrt{n})$ factor algorithm.

Theorem 7 *There is a $2\sqrt{n}$ factor approximation algorithm for the* MinBudget *problem.*

Proof. Let B^* be the budget required by the optimum solution. We will show that the minimum s-t cut is of size at most $(B^*+1)\sqrt{n}$, which will prove the factor since $B^* \geq 1$. Note that vaccinating any s-t cut on the first time step is a valid solution.

Consider the set of nodes Y vaccinated in the first $\sqrt{n}$ time steps. Note that $|Y| \leq B^*\sqrt{n}$. Since this is a valid vaccination strategy, any path from s of t of length at most $\sqrt{n}$ contains at least one vertex of Y. Consider the graph G' induced by the vertex set $V' = V \setminus Y$. If s and t are disconnected in G', then Y is an s-t cut. Otherwise, the shortest path from s to t in G' is at least $\sqrt{n}$. This implies that the minimum s-t cut, call it X, in G' is of size at most $\sqrt{n}$. This follows from Menger's theorem which states that if the minimum s-t cut is k, there are k internally vertex disjoint paths from s to t; since each such path is of length at least $\sqrt{n}$, we must have $k \leq \sqrt{n}$. Thus there is a cut $(X \cup Y)$ of size $(B^*+1)\sqrt{n}$ separating s and t. ■

Now we give an approximation algorithm for directed layered graphs. In an *s-t directed ℓ-layered network*, the vertex set consists of $V = (L_0 := \{s\}) \cup L_1 \cup \cdots \cup L_\ell \cup \{t\}$, and all arcs except those entering t are from a vertex in some layer L_i to a vertex in L_{i+1}; arcs entering t may originate from any vertex (other than t). We give an $H(\ell)$ factor, where $H(t) = 1 + \frac{1}{2} + \cdots + \frac{1}{t}$. The algorithm is based on the LP relaxation and its dual described in Figure 2.1.

The primal LP has a variable x_v^τ which indicates whether vertex v is vaccinated at time τ or not. $\ell \leq n$ is the length of the longest path from s to t; it is easy to see that we will not vaccinate any vertex after time ℓ. The first constraint bounds the number of vaccinations at every time instance. The second constraint says that for every path $(s, v_1, \cdots, v_k, t)$ to the sink t, one of the nodes, say v_i, must be vaccinated *by* time i. This is a necessary and sufficient condition for this path not to transmit

$$\text{Minimize} \quad B \tag{Primal}$$

$$\sum_{v \in V} x_v^\tau \leq B \quad \forall \tau = 1, \ldots, \ell \tag{2.6}$$

$$\sum_{i=1}^{k} \sum_{\tau=1}^{i} x_{v_i}^\tau \geq 1 \quad \forall (s, v_1, \cdots, v_k, t) \in \mathcal{P} \tag{2.7}$$

$$x_v^\tau \geq 0, \quad \forall v \in V, \ \forall \tau = 1, \ldots, \ell \tag{2.8}$$

$$\text{Maximize} \quad \sum_{P \in \mathcal{P}} f_P \tag{Dual}$$

$$\sum_{\tau=1}^{\ell} z_\tau \leq 1 \tag{2.9}$$

$$\sum_{P \in \mathcal{P}: v \in P^{(\tau)}} f_P \leq z_\tau \quad \forall v \in V, \tau = 1, \ldots, \ell \tag{2.10}$$

$$z, f \geq 0 \tag{2.11}$$

Figure 2.1: The LP relaxation for MinBudget **in the non-spreading model and its dual.**

the infection to t. In the dual, we have a flow for every s-t path P. We also have a variable z_τ which add up to 1. The second constraint in the dual is a bit subtle: it says, for every τ, the total flow through a vertex v via paths such that v lies at a distance τ or more from s on the path, is at most z_τ. In the LP, $P^{(\tau)}$ denotes the portion of the path from the τth vertex to t. That is if $P = (s, v_1, \ldots, v_k, t)$, then $P^{(\tau)} = (v_\tau, v_{\tau+1}, \ldots, t)$.

Although the primal LP above has exponentially many constraints, it can be solved in polynomial time since one can obtain the separation oracle in polynomial time. However, we will give a combinatorial algorithm and use the LP (in fact, the dual) only for analysis. Strictly speaking the LP (Primal) may have an integrality gap of $n = |V|$. However note that if OPT denotes the optimal value of (Primal), then in fact $\lceil OPT \rceil$ is a lower bound on the minimum budget, and by comparing the budget of our solution against this lower bound, we prove the following theorem.

Theorem 8 *If the network is a layered directed graph with ℓ layers, then there is a* *poly-time $H(\ell)$ approximation algorithm to the* MinBudget *problem.*

Proof. The algorithm first computes a 'fractional' vaccination strategy, that is, the strategy would be feasible except it could possibly vaccinate a non integral number of vertices. The second step converts this strategy into a feasible one with essentially the same guarantees. For this last step we need the following fact which follows from standard results about minimum-cost network flows.

Fact 9 (see, e.g., Application 6.3 in [116]) *Given a matrix M' with possibly fractional entries, one can obtain another* integral *matrix M such that for every row i and column j, we have (i) $M_{ij} \in \{\lfloor M'_{ij} \rfloor, \lceil M'_{ij} \rceil\}$; (ii) the row-sum of row i in M is the floor or ceiling of the row-sum of row i in M'; and (iii) the column-sum of column j in M is the floor or ceiling of the column-sum of column j in M'.*

Algorithm DirLayNet

1. Set the capacity of each vertex $v \in L_i$ at $\frac{1}{iH(\ell)}$.
2. Find the minimum cut in this capacitated network, let it be $(N_1 \cup \cdots \cup N_\ell)$ with $N_i \subseteq L_i$.
3. Construct the following possibly non-integral matrix M'. Let $M'_{ij} := |N_j|/j$, for all $1 \leq i \leq j \leq \ell$. Note that for any column j, the column sum of M' is exactly $|N_j|$.
4. Apply Fact 9 to construct the corresponding integral matrix M from M'. The vaccination strategy is as follows: on day i, vaccinate M_{ij} nodes from layer j, for all $i \leq j \leq \ell$.

Proposition 3 shows that the above algorithm returns a feasible vaccination strategy. Define $ALG := |N_1| + \frac{|N_2|}{2} + \cdots + \frac{|N_\ell|}{\ell}$. The number of vertices vaccinated on any day i is

$$\sum_{j \geq i} M_{ij} \leq \lceil \sum_{j \geq i} M'_{ij} \rceil \leq \lceil \sum_{j \geq 1} M'_{ij} \rceil = \lceil ALG \rceil.$$

Thus, the cost of algorithm DirLayNet (i.e., the maximum number of vertices vaccinated in any time step) is at most $\lceil ALG \rceil$. Proposition 4 proves that there is a feasible dual solution to (Dual) of value $\frac{1}{H(\ell)} \cdot ALG$, and hence $OPT \geq \frac{1}{H(\ell)} \cdot ALG$. This implies that the cost of algorithm DirLayNet is at most $\lceil H(\ell) \cdot OPT \rceil \leq H(\ell) \lceil OPT \rceil$. ■

Proposition 3 *Algorithm* DirLayNet *returns a feasible vaccination strategy.*

Proof. We claim that for any $1 \leq j \leq \ell$, all the nodes of N_j are vaccinated by time step j. This ensures that the vaccination strategy is feasible, since any path $(s, v_1, v_2, \ldots, v_k, t)$ from s to t (where $k \leq \ell$), must have $v_j \in N_j$ for some j ($\bigcup_j N_j$'s form a cut).

By day j, we vaccinate $\sum_{i \leq j} M_{ij}$ nodes of N_j. But, $\sum_{i \leq j} M'_{ij} = |N_j|$, and therefore integral; this implies $\sum_{i \leq j} M_{ij}$ is also precisely that. This completes the proof. ■

Proposition 4 *There is a feasible dual solution to* (Dual) *of value $ALG/H(\ell)$.*

Proof. Look at the LHS of the second constraint in the dual. We claim that for layered networks, if vertex v lies in L_i, for any feasible dual solution f we have

$$\sum_{P : v \in P^{(\tau)}} f_P = \begin{cases} \sum_{P : v \in P} f_P & \text{if} \quad t \leq i, \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

This is because *every path that contains $v \neq t$, contains it at exactly the ith position.* Now we exhibit the desired dual solution. To do so, note that corresponding to the minimum vertex cut $(N_1 \cup \ldots \cup N_\ell)$, we have a feasible flow f of the same value. Furthermore, because of the capacity constraints, we get for $v \in L_i$,

$$\sum_{P : v \in P} f_P \leq \frac{1}{iH(\ell)} \,. \tag{2.13}$$

Construct the dual solution with this f and let $z_i = \frac{1}{iH(\ell)}$ for $i = 1, \ldots, \ell$. Note that the first constraint of the dual is satisfied. The second constraint follows from (2.12) and (2.13), and the fact that z_i's are decreasing. Note that the value of the dual is equal to the capacity of the minimum cut which is

$$\frac{1}{H(\ell)}|N_1| + \frac{1}{2H(\ell)}|N_2| + \cdots + \frac{1}{\ell H(\ell)}|N_\ell| = \frac{ALG}{H(\ell)}.$$

■

We remark that the special case where the underlying graph G is a tree rooted at s has received a lot of attention [56, 57] and is computationally difficult [113]. To give some intuition for why this is the case, we present a difficult scheduling problem that is equivalent to MinBudget on trees. Consider a problem where there is a set of jobs, each with a deadline for completion. If a job is not completed by its deadline, then it splits into multiple jobs, each with a new deadline. We have a capacity of completing only a fixed number of jobs at every time-step. In this case what strategy should be applied so that all remaining jobs are completed by processing the fewest jobs at each step?

Notice that for trees the spreading model and non-spreading model are equivalent due to the following reason. For the spreading model on general graphs we defined a function $\Gamma(v)$ as a set of all tuples (u, τ) such that if u is vaccinated directly at time τ then the node v will be saved. For a tree, it is easy to observe that a node v will be saved if any of its ancestors is vaccinated directly before the infection reaches v. Therefore, the optimal strategy will be the same on a given tree irrespective of the vaccination model being spreading or non-spreading. This implies that all the positive results from Section 2.2 also hold for trees. Also, observe that the MinBudget problem on a tree G with height h yields an instance of MinBudget on an s-t directed graph with h layers. Hence, we immediately obtain the following corollary of Theorem 8.

Corollary 2 *There is an $O(\log h)$-approximation for* MinBudget *on trees, where the set T is the set of leaves and h is the height of the tree.*

2.3.3 An Integrality-gap Example

We conclude this section by showing an integrality gap of $\Omega(\log n)$ for the above LP relaxation. The example is an ℓ layered graph containing $O(\ell^3)$ vertices.

We show a gap of $H(\ell) = \Omega(\log n)$.

Consider an s-t ℓ-layered graph with layers $(L_0 = \{s\}, L_1, \ldots, L_\ell)$ with an arc from s to all vertices of L_1 and an arc from any vertex in L_ℓ to t. Furthermore, there will be an arc from any vertex in L_i to any vertex in L_{i+1} for $i = 1, \ldots, \ell - 1$. The size of L_i is $i\ell$. Thus, the total number of vertices is $O(\ell^3)$. Note that the integral optimum will be ℓ; one can show if one vaccinates less than $i\ell$ nodes by time i, then the infection spreads to layer $i + 1$. To see a fractional solution, consider the solution where for each i, we set $x_v^i = \frac{1}{iH(\ell)}$ for all $v \in L_i$, and $x_v^i = 0$ for all other v. Any path from s to t then has the LHS of constraint 2 precisely equal to $\frac{1}{H(\ell)}(1 + 1/2 + \cdots + 1/\ell) = 1$. The value of the solution is $i\ell\frac{1}{iH(\ell)} = \frac{\ell}{H(\ell)}$.

CHAPTER 3
Diffusion in Complex Networks

3.1 The Diffusion Model

We describe the diffusion model used to study propagation of evacuation news through a population that was first introduced and motivated by C. Hui in her doctoral thesis [117]. It takes into consideration the fact that agents may act on information and leave the network; which means diffusion occurs on network that is dynamic. The model is based on concepts found in the widely studied SIR model of epidemiology and standard threshold and cascade models.

We have an initial graph $G = (V, E)$ where V are individuals in the population while edge $(u, v) \in E$ represents a social connection between individuals u and v. There are K sources with information value $\{I_1, I_2, \ldots, I_K\}$. Also, a source $i \in \{1, \ldots, K\}$ can seed $S_i \geq 0$ nodes.

Associated with edge $(u, v) \in E$ is a trust value $0 \leq \alpha(u, v) \leq 1$. It represents the amount of trust node v has on information provided by u. The graph may be directed with different trust values on edges (u, v) and (v, u) representing asymmetrical trust. Each node $v \in V$ also has a similar trust value for each source $i \in \{1, 2, \ldots, K\} : \alpha(v, i)$.

Associated with each node u are two thresholds, $t_l(u)$ and $t_h(u)$ such that $0 \leq t_l(u) \leq t_h(u)$. Also, each node u possesses an *information value set* $S(u) = \{(s_1, v_1), \ldots, (s_K, v_K)\}$ for the K information sources. It is made up of tuples (s_i, v_i) where v_i is the information from source s_i. Based on its information value set, a node calculates its *information value* as follows:

$$I(u) = \lambda_d \sum_{i=1}^{K} v_i + (1 - \lambda_d) \cdot \max_{i=1,..,K} v_i,$$

where $0 \leq \lambda_d \leq 1$ is a model parameter.

Based on the information value, a node u can lie in any of the following states:

1. **Disbelieved**: If $I(u) < t_l(u)$ then the node does not believe it has any meaningful information. In this state nodes do not take any action except for incorporating information value sets shared with them by their neighbors.

2. **Undecided**: If $t_l(u) \leq I(u) < t_h(u)$ then the node believes it has some information but is uncertain about evacuating. In this state nodes query their neighboring nodes and acquire their information sets to be incorporated with their own.

3. **Believed**: If $I(u) \geq t_h(u)$ then the node has enough information to accept the news of evacuation. In this state the node actively propagates its information value set with its neighbors for X time-steps before evacuating. Here $X > 0$ is a model parameter.

4. **Evacuated**: X time-steps after a node enters the Believed state, it evacuates resulting in a disconnection from the network. Hence the graph changes and a new graph G' is obtained where $G' = (V', E')$, $V' = V - u$ and $E' = E - (u, v)\, \forall v \in V'$, $(u, v) \in E$.

3.1.1 The Diffusion Process

Before information from sources is received, for all nodes $v_i = 0\, \forall i \in \{1, \ldots, K\}$, i.e., the information value in every source-value pair is 0 and all nodes are in disbelieved state. The diffusion process begins when some nodes are seeded with information from sources according to the seeding strategy. Seeding entails transfer of information from sources to the information value set of the selected nodes. This transfer of information gets attenuated by the trust between the node and the source. So if source s seeds node u then the information transfer to node u is of value $\alpha(u, i) \cdot I_s$. Each node then computes its information value and ascertains its state. At every consecutive step nodes take action depending on their state and assimilate information thus acquired.

We now describe the process of propagating and updating information sets. When node u propagates its information set to neighbor v, the value of information gets similarly attenuated by a factor of $\alpha(u, v)$. Thus if (s_i, v_i) is the source-value propagated to node v, then the source value pair received by node v will be $(s_i, \alpha(u,v) \cdot v_i)$. In this way after propagation a node might end up with, including its own, multiple source-value pairs of the same source. Lets say that node u has m source-value pairs of source s_i: $v_i^1, v_i^2, \ldots, v_i^m$. Then node v will construct a new source-value pair (s_i, v_i^*) as follows:

$$v_i^* = \lambda_s \cdot \sum_{j=1}^{m} v_i^j + (1 - \lambda_s) \cdot \max_{j=1,..,i} v_i^j.$$

Again λ_s is a model parameter such that $0 \leq \lambda_s \leq 1$. Thus the new information value set, and hence the information value, is calculated by assimilating all source value pairs obtained after propagation.

The diffusion process continues until either all nodes evacuate or there is no change in information value sets of the nodes. Note that the 3 model parameters, $(\lambda_s, \lambda_d, X)$, and threshold values of nodes are specified externally to the diffusion process.

3.1.2 Objective

The goal of this study is to find a seeding strategy that maximizes the number of nodes that end up in their Believed state given a set of sources. For this purpose we define a *coverage function* on the nodes $\Gamma : (2^V)^K \to \mathbb{R}_{\geq 0}$. The input of this function is K sets of nodes, each of size at most S_i, that are seeded by the source nodes. It gives the number of nodes converted to Believed state as output. Therefore, the objective is to find K sets of nodes $\{\psi_1, \psi_2, \ldots, \psi_K\}$ where each $\psi_i \in V$ and $|\psi_i| \leq S_i$, such that $\Gamma(\psi_1, \psi_2, \ldots, \psi_K)$ is maximized.

3.2 Analysis

In order to find optimal sets of nodes to seed, it would be beneficial if we could obtain some theoretical insight about the behavior of the diffusion process. Unfortunately the complexity of the model described in the previous section makes it intractable for theoretical analysis. Instead, we consider a simplified model of diffusion in our theoretical analysis, use the insight from this analysis to develop a seeding algorithm for the general model, and then test this algorithm in many instances of the general diffusion process.

To demonstrate the complexity of the general diffusion model described above, note that it would be desirable for Γ to have certain simple and intuitive properties, such as monotonicity and submodularity. But as we show below, these properties fail to hold in the general diffusion model. In the following examples, $K = 1$, i.e., there is only one source and its information value $I_1 = 1$.

Monotonicity: Monotonicity of Γ simply means that, the more nodes are seeded, the more nodes are converted in the end. Unfortunately, this property fails to hold. Consider the example shown in Figure 3.1.

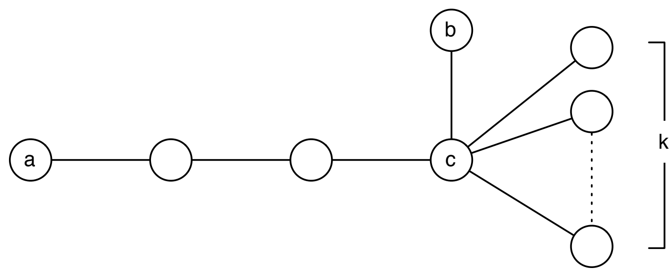


Figure 3.1: Example which shows that diffusion model in non-monotonic.

Here $\lambda_d = \lambda_s = 0$ and $X = 1$. Trust values on all edges are 1.0 except for edge (b, c), which has trust value 0.1. For all nodes $t_l = t_h = 0.5$ except for node c for which $t_l(c) = t_h(c) = 0.1$. Now consider the coverage function for this example. Monotonic property implies that $\Gamma(S) \leq \Gamma(T) \, \forall S \subseteq T$.

Now consider the sets $S_1 = \{a\}$ and $S_2 = \{a, b\}$. It can be easily shown that $\Gamma(S_1) = k + 4$ as information of value 1.0 will successfully propagate to all nodes except b thereby converting them to the Believed state. Now consider what happens when the set S_2 is seeded. Seeding node b will cause a propagation of information value 0.1 to node c which is enough to push c into the Believed state. Once in the Believed state, node c will propagate its information to its neighboring nodes but their thresholds are too high for them to change their state. Since $X = 1$, node c will leave the network before the higher value information from path to node a

reaches it. This results in the k neighbors being cut off from information and they remain in Disbelieved state. Hence $\Gamma(S_2) = 4$.

As the example shows, if nodes evacuate and leave the network, it may result in non-monotonic behavior of the coverage function.

Submodularity One of the equivalent definitions of a submodular function is as follows. Given that $A \subseteq B \subseteq U$, a set function $f : 2^U \to \mathbb{R}_{\geq 0}$ is submodular if for any set C: $f(A \cup C) - f(A) \geq f(B \cup C) - f(B)$. Thus submodularity is essentially the discreet version of concavity.

Now consider the example shown in Figure 3.2.

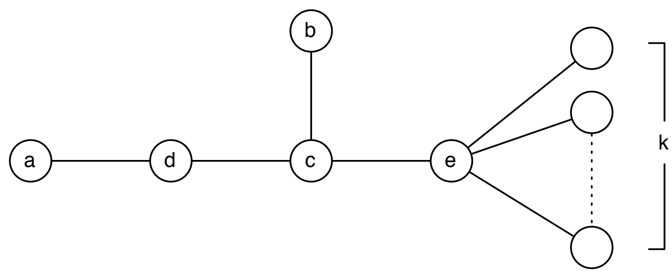


Figure 3.2: Example which shows that diffusion model is not submodular.

In this example we have $\lambda_s = \lambda_d = 0$ and $X = 5$. Trust values $\alpha(a, d) = 0.9$ and $\alpha(b, c) = 0.1$ while all other edges have trust value 1.0. All nodes except c and d have $t_l = t_h = 0.5$, whereas $t_l(d) = t_h(d) = 0.91$ and $t_l(c) = 0.1$, $t_h(c) = 0.9$.

Now consider the seed sets $S_1 = \emptyset$, $S_2 = \{a\}$, $S_3 = \{b\}$ and $S_4 = \{a, b\}$. If the coverage function is submodular then the following inequality should hold:

$$\Gamma(S_3) - \Gamma(S_1) \geq \Gamma(S_4) - \Gamma(S_2). \tag{3.1}$$

It is easy to see that $\Gamma(S_1) = 0$. Also seeding node a (set S_2) will result in propagation of information of value 0.9 to node d. Since $t_l(d) = 0.91$, it will remain in its Disbelieved state. Therefore $\Gamma(S_2) = 1$. Similarly if node b (set S_3) is seeded then information of value 0.1 will propagate to node c causing it to enter the Undecided state. Node c will now query its neighbors but none of them have any new information and so c will remain in the Undecided state and diffusion process will end. So

$\Gamma(S_3) = 1$.

Now, if set S_4 is seeded, node d will have information of value 0.91 and c will be in Undecided state as explained in the above two cases. However when c queries d, it receives enough information (0.9) to push it into the Believed state. Node c in turn propagates this information to its k neighbors ultimately converting all of them to the Believed state. Therefore, $\Gamma(S_4) = k + 4$. This means Inequality 3.1 does not hold and hence the coverage function is not submodular.

The fact that node c could query for information that d did not transmit resulted in the cascade effect which converted all the nodes to Believed state. Hence, existence of the query state in this case caused the coverage function not to be submodular.

3.2.1 A Simplified Model

Taking a cue from the examples above, we define a simplified model that is a strict specialization of the general diffusion model defined earlier. Note that while doing experimental simulations we will use the general diffusion model. We use the simplified model only to carry out theoretical analysis. Insights obtained for this simplified model help us develop heuristics that we then apply to the general case.

In this model $\lambda_d = \lambda_s = 0$. This implies that while calculating information value, the maximum value from source-value pairs is chosen i.e., $I(u) = \max_{j=1,..,K} v_j$. Similarly, only the maximum value is chosen while updating the source-value pair for the same node. In the simplified model, nodes have a single threshold $t = t_l = t_h$. This eliminates the existence of the Undecided state which, as we saw leads to undesirable properties in the coverage function. Also, X is sufficiently large (say $\omega(|E|)$) so it can be safely assumed that nodes do not leave the network while diffusion is occurring. In this model, all source information values are same and denoted by I. Also, the trust values between sources and nodes are identical. Therefore, $I_1 = I_2 = \ldots, = I_K = I$ and for all $u, v \in V$ and s in sources, $\alpha(v, s) = \alpha(u, s)$.

If there are multiple sources $A = \{I_K, \ldots, I_K\}$, then consider an alternate source s' with seed set of size $\sum_{i=1}^{K} S_i$ and information value I. Its trust value is same as the original sources. Given any seeding strategy $\psi = \{\psi_1, \ldots, \psi_K\}$ for the original set of sources, a new seeding strategy can be constructed for the the alternate source as $\psi' = \bigcup_{i=1}^{K} \psi_i$.

Proposition 5 *$u \in V$ will be converted to Believed state by source s' and seeding ψ' if and only if u is also converted by seeding ψ with sources A.*

Proof. Consider a node v that is converted to Believed state in source s' and seeding ψ'. Since at every step of propagation of information, only the maximum value is chosen, the information value at v can be traced back to a node $u \in \bigcup_{i=1}^{K} \psi_i$. Let the path from u to v along which the information travelled be $(u, w_1, w_2, \ldots, w_k, v)$. Since there is no Undecided or query state in the simplified model, node w_k would have had to transmit information to node v, i.e., it had be in the Believed state. And since w_k transmitted its highest information value to v, an information value of $I(v)/\alpha(w_k, v)$ was enough to convert w_k to Believed state. Applying this argument serially in the reverse order to all nodes in path $(u, w_1, w_2, \ldots, w_k, v)$, we can say that every node would have been converted to Believed state by information received from its predecessor in the path.

Hence, v is converted to the Believed state as u is part of the seeding strategy, irrespective of the other seeds. But u is also part of the seeding strategy ψ with sources A. Moreover, the information value and trust is also the same for sources S and s'.

This proves the statement of proposition. ∎

Therefore, we only consider 1 source for the following analysis of the simplified model. We show that the coverage function Γ is submodular and monotone for the simplified model with the help of Lemma 4.

Lemma 4 *Consider a function $\bar{\Gamma} : 2^V \rightarrow 2^V$ that maps the seed nodes to nodes that are consequently converted to Believed state. For any set $S \subseteq V$,*

$$\bar{\Gamma}(S) = \bigcup_{x \in S} \bar{\Gamma}(\{x\}).$$

Proof. We will prove the statement by induction. For the base case of $S = \emptyset$ the statement holds trivially. Now let $\bar{\Gamma}(S) = \bigcup_{u \in S} \bar{\Gamma}(\{u\})$ for some $S \subseteq V$ and let $v \in V$ be any node such that $v \notin S$.

Consider a node $x \in \bar{\Gamma}(S \cup v)$. The information value at node x after the diffusion process ends is $x(I)$. Since at every step of propagation of information, only the maximum value is chosen, the information value $x(I)$ can be traced back a to a node $u \in S \cup v$. Let the path from u to x along which the information travelled be $(u, w_1, w_2, \ldots, w_k, x)$. Since there is no Undecided or query state in the simplified model, node w_k would have had to transmit information to node x, i.e., it had be in the Believed state. And since w_k transmitted its highest information value to x, an information value of $x(I)/\alpha(w_k, x)$ was enough to convert w_k to Believed state. Applying this argument serially in the reverse order to all nodes in path $(u, w_1, w_2, \ldots, w_k, x)$, we can say that every node would have been converted to Believed state by information received from its predecessor in the path. This means $x \in \bar{\Gamma}(u)$, that is to say, either $x \in \bar{\Gamma}(S)$ or $x \in \bar{\Gamma}(v)$.

Hence $\bar{\Gamma}(S \cup v) = \bar{\Gamma}(S) \cup \bar{\Gamma}(v)$ and so the lemma is proved by induction. ■

Lemma 4 implies that the coverage function is of the following form

$$\Gamma(S) = \left| \bigcup_{x \in S} \bar{\Gamma}(\{x\}) \right|.$$

First of all, this shows that the function if monotone. Also functions of this form are known to be submodular (see [114] for example).

We can additionally use the following theorem to show that a simple greedy algorithm gives good approximation to the problem of maximizing the value of the coverage function.

Theorem 10 *There is a deterministic algorithm which achieves a $(e-1)/e$-approximation for the problem of maximizing a monotone submodular function. ([114]).*

If n nodes are to be selected for seeding then the greedy algorithm is as follows:

Algorithm 1: Greedy algorithm on instance of simplified model

- Initiate converted nodes $C = \emptyset$ and selected nodes $S = \emptyset$;

while $|S| < n$ *and* $|C| < |V|$ **do**

  Calculate $\bar{\Gamma}(v)$ for each $v \in V \backslash C$;

  Choose v that maximizes $|\bar{\Gamma}(v) \backslash (\bar{\Gamma}(v) \cap C)|$;

  Set $S = S \cup v$ and $C = C \cup \bar{\Gamma}(v)$;

end

- Output set S.

Implementation of this algorithm is discussed in the section 3.3.

3.3 The Projected Greedy Heuristic

We now describe Projected Greedy heuristic which takes an instance $\mathbb{G}$ of the general model and generates a seed set $\{\psi_1, \psi_2, \ldots, \psi_K\}$. It utilizes the fact that we can give provably close to optimal seed set for instance of the simple model. Given an instance $\mathbb{G}$ of the general model it creates an instance of the simplified model $\mathbb{S}$. It then runs the Greedy algorithm described in Algorithm 1 to obtain a seed set for $\mathbb{S}$ which it then maps to a seed set for $\mathbb{G}$. We will distinguish between the coverage functions for as follows: Γ_G is the coverage function for instance of the general model whereas Γ_S is the coverage function for instance of the simple model.

Creating instance of simple model Given an instance $\mathbb{G}$ of the general model the Projected Greedy heuristic creates an instance of the simplified model $\mathbb{S}$, in the following manner.

- For the K sources in $\mathbb{G}$ construct one source s of $\mathbb{S}$ with information value

$$I = \frac{1}{(K \cdot \sum_{i=1}^{K} S_i)} \sum_{i=1}^{K} I_i \cdot S_i.$$

- For any $v \in V$ the trust value for source s is given by

$$\alpha(v, s) = \frac{1}{(K \cdot \sum_{i=1}^{K} S_i)} \sum_{i=1}^{K} \left[\frac{1}{|V|} \sum_{u \in V} \alpha(u, i) \right].$$

- The number of nodes that s can seed is $\sum_{i=1}^{K} S_i$.
- Set $\lambda_s = \lambda_d = 0$.
- Select a constant $0 \leq \tau \leq 1$ and set thresholds $t(v) = \tau$ for all $v \in V$. The selection of τ is discussed later.
- Finally, edge trust values in $\mathbb{S}$ are the same as in $\mathbb{G}$.

Generating seeds Once instance $\mathbb{S}$ is created, the Projected Greedy heuristic uses the algorithm described in Algorithm 1 to produce seed set ψ. This seed set is then arbitrarily partitioned into K sets $\{\psi_1, \psi_2, \ldots, \psi_K\}$ such that $|\psi_i| \leq S_i$, i.e., they are distributed among the K sources of $\mathbb{G}$ forming a valid seed set for $\mathbb{G}$. $\Gamma_G(\psi_1, \ldots, \psi_K)$ is evaluated for instance $\mathbb{G}$.

The Projected Greedy heuristic generates seeds for set of single thresholds $\Omega = \{\tau_1, \tau_2, \ldots, \tau_c\}$ where each $0 \leq \tau_i \leq 1$ and c in the following manner. Let $\bar{t_l}(u)$, $\bar{t_h}(u)$ respectively be the lowest lower threshold and the highest upper thresholds of nodes in instance $\mathbb{G}$. Consider a case where all edges have the same trust. Let this trust value be t_{avg}. The Projected Greedy heuristic then sets $\tau_1 = \bar{t_h}(u)$. For $i \in \{2, \ldots, c\}$, $\tau_i = \tau_{i-1} \cdot t_{avg}$. The number of thresholds selected c is such that $\tau_c \geq \bar{t_l}(u)$ and $\tau_c \times t_{avg} < \bar{t_l}(u)$.

In the case where there are two types of edges: ones with high trust value τ_{high} and low trust values τ_{low}, the set Ω is constructed as follows:

- $\bar{t_h}(u) \in \Omega$,
- For every $\tau \in \Omega$, $\tau_{high} \cdot \tau \in \Omega$ if $\tau_{high} \cdot \tau \geq \bar{t_l}(u)$,
- For every $\tau \in \Omega$, $\tau_{low} \cdot \tau \in \Omega$ if $\tau_{low} \cdot \tau \geq \bar{t_l}(u)$.

As the number of distinct trust values on edges increases, set Ω can be similarly calculated. So the value c essentially depends on the rate of decay of information value from source s as it diffuses through the network. The faster the decay of information, the smaller is the value of $c = |\Omega|$.

The heuristic then evaluates $\Gamma_G(\psi_1, \ldots, \psi_K)$ for each of the seed sets thus generated and outputs the seed set $\{\psi_1, \psi_2, \ldots, \psi_K\}$ that maximizes Γ_G for $\mathbb{G}$. We discuss the time complexity of Projected Greedy heuristic in section 3.3.1.

Although the seed set generated by greedy algorithm gives a close to optimal solution for instance $\mathbb{S}$, it does not provide any performance guarantees for the original instance of the general model $\mathbb{G}$. It can be used only as a heuristic. But as we describe in later sections, experimental simulations show that the Greedy heuristic performs significantly better that widely used simple seeding strategies.

3.3.1 Time Complexity and Comparison with Actual Greedy

As described in the previous section, the basic methodology of Projected Greedy heuristic includes mapping instance of general model to instance of simple model, calculating seed set using greedy algorithm and then projecting it to instance of general model. A small number (c) of seed sets are calculated and the best among them is chosen.

One may intuitively question the efficacy of *mapping* the instance from general to the simple model. After all, there seemingly exists a better seeding strategy where instead of calculating it in the simplified version, compute the seed set greedily on the instance of the general model itself. In order to select seed set of size say P, start with an empty set ψ and iteratively add P nodes that give the largest increase in the value of $\Gamma_G(\psi)$. Lets call this strategy *Actual Greedy.*

Consider the time-complexity for this algorithm. For selecting the best node to be added to ψ, each node has to be tried successively and $\Gamma_G(\psi)$ has to be evaluated. Since calculating Γ_G by simulation over sparse graphs takes $O(n)$ time, selecting the best node at every step takes $O(n^2)$ time. If the number of nodes to be selected is large, i.e., $P = O(n)$, then the whole Actual Greedy algorithm will take $O(n^3)$ time.

As mentioned earlier, the diffusion process on the general model is complex: it is stochastic with multiple sources having different information values etc. As a result, there are no nice properties that be exploited for reducing its running time. At each step Γ_G has to be explicitly simulated. Contrast this with the simple model where the diffusion process is deterministic and Γ_S is monotone submodular.

In our implementation we create a special data structure δ such that $\delta(v)$ is the set of all nodes u such that v is converted to Believed state when u is seeded. While calculating δ, we also create array N where $N(u)$ is the number of nodes that u converts to the Believed state. For sparse graphs, building δ and N for all $v \in V$ takes $O(n^2 \ln n)$ time. After selecting the node that gives the best improvement in Γ_S, we can to update the array N using δ. This means we do not need to recalculate N at every step and the update takes only $O(n)$ time. The process of updating array N depends on properties of Γ_S such monotonicity. Thus selecting a seed set of size $O(n)$ in the simple model takes $O(n) * O(n) + O(n^2 \ln n) = O(n^2 \ln n)$ time. Since the Projected Greedy repeats this selection process c times and also evaluates Γ_G for the projected seed set, the total running time is $c \cdot (O(n^2 \ln n) + O(n)) = O(n^2 \ln n)$. To give an idea about how large the value of c is, imagine an extreme case with homogenous trust values where $\bar{t_l}(u) = 0.01$, $\bar{t_h}(u) = 0.99$ and $t_{avg} = 0.9$. In this unlikely scenario $c = 44$. Even if the rate of information decay is extremely low and the average trust is as high as $t_{avg} = 0.99$, the value of c blows up to 458. This is still orders of magnitude better than n which can be in millions.

In practice though, our method of pre-calculating the data structure δ can lead to space complexity of $O(n^2)$ in the worst case. In order to avoid this, we use a hybrid strategy for computing seed set for the Projected Greedy heuristic. In this strategy δ is not stored initially and array N is recalculated each time the best node is to be selected. Even this process benefits from the submodular property of Γ_S and performs much better as compared to the brute force approach required for Actual Greedy. We use a technique mentioned in [24] according to which array N is stored as a priority queue and each selection requires recalculation of $N(u)$ for only a small number of nodes u. Only after a certain threshold is breached do we switch to populating δ for all nodes. Since nodes that convert the highest number of other nodes to Believed state have already been selected at this stage, it helps in reducing the size of $\delta(u)$ for each of the remaining nodes. Not only does this help reduce memory usage but it also does not affect the running time adversely.

Therefore using the Projected Greedy heuristic gives an improvement of a factor of n over Actual Greedy. In large graphs, this is a significant improvement and can sometimes be the difference between feasibility and intractability.

3.4 Experiment Design

In order to compare the Greedy heuristic with other seeding strategies, we simulate the spread of evacuation warnings in a social network. The simulation of diffusion is carried out on different types of networks with several parameter values.

3.4.1 Networks

We have used three different network structures, each with $100,000$ nodes with average graph degree being approximately 4.00. Frequently, in real world scenarios individuals form groups based on race, ethnicity, nationality, etc. Not only are individuals within the same social group more likely to be acquainted with each other, they are also inclined to place more trust with people in the same group as theirs. Since these connections and trust disparities play an important role in the flow of information, we model social groups by dividing the population into 2 groups in following networks.

Scale-free Graphs: We use the Albert-Barabási model for generating random scale-free networks using the preferential attachment model [118, 119]. Once the graph is created, $50,000$ nodes are randomly assigned to one group and the rest to other.

Scale-free graphs do not take into consideration the fact that nodes within group are more likely to communicate with each other.

Random Group Model: Nodes are randomly assigned such that each of the 2 groups contains $50,000$ nodes. The probability that 2 nodes are connected (edge probability) depends on whether they belong to the same group or not. If two nodes belong to the same group then the edge probability is p_s while if they belong to different groups, the edge probability is p_d. Here $p_s = 2 * p_d$ and the absolute value is adjusted so that average degree of graph is 4.00. Thus in this network, individuals within a group are more likely to have and edge between them.

San Diego Network: This is a random geometric graph that is constructed from actual demographic data in the San Diego area [117]. Since there is a large population of Hispanics in the area, the two groups considered are Hispanics and Non-hispanics. The population of nodes belonging to each group is based on the racial demographics of the region. Also, the edge probability between two nodes depends not only on the group they belong to, but also the physical distance between them. For example the edge probability between two nodes belonging to the same group is larger if they live close to each other than if they live far apart. Again the edge probabilities are scaled such that average graph degree is 4.00.

3.4.2 Node Characteristics

The spread of evacuation warnings intend to convert the nodes to their Believed state. Since an individual eventually evacuates after reaching the Believed state, the consequences of being convinced of the evacuation message are costly. Individuals may want to make sure that they have significant amount of information before they decide to evacuate. This can be modeled by keeping the t_h value of nodes high. On the other hand, for some individuals who have say sufficient insurance on their house and belongings, evacuation may be a low risk endeavor as compared not leaving. For such cases t_h value can be kept low. In some cases individuals may be more risk averse and proactive, that is to say, they may be willing to put more effort in collecting information. This can be modeled by keeping the t_l value low.

In our experiments, we simulate 3 different threshold value pairs for all nodes: $[t_l, t_h] = [0.2, 0.3], [0.15, 0.55]$, and $[0.4, 0.5]$. Also, nodes leave the network at $X = 5$ time-steps after they are converted to their Believed state.

3.4.3 Trust Scenarios

Since nodes are split into 2 groups, there are 2 kinds of edges in the graph. The first type (denoted by A) of edge is incident on nodes from the same group. The second type (denoted by B) of edge is incident on nodes from different groups. Based on the trust values on these edges we have two trust scenarios. In each scenario, the average trust on all edges of the graph is denoted by t_{avg}.

Homogenous trust: In this scenario, all edges have the same trust value. This models situations when no social groups exist and trust between every pair of individuals is the same. The trust value on every edge is thus the same as t_{avg}.

Group Variable trust: In this scenario, the trust value on type A edges is $t_{avg}+\varepsilon$ where $\varepsilon > 0$. Trust value on type B edges is scaled down such that the average trust in the graph is still t_{avg}. In effect, this reduces the trust on type B edges. So this models the circumstance where individuals place more trust with others within their social group as compared to those from different groups.

We use $t_{avg} = 0.7$ and $\varepsilon = 0.05$ for the purpose of simulation.

3.4.4 Seeding Strategies

We have 5 trustworthy sources each with information value $I = 0.95$ and trust value $\alpha(v, i) = 0.9$ for all $v \in V$. We look at two scenarios, one in which 5% of total nodes are seeded and the other in which 10% are seeded. In both cases, each source seeds equal number of nodes. Therefore, if total number of nodes seeded is P, then $S_i = P/5$ for $i \in \{1, 2, 3, 4, 5\}$.

We test and compare the following algorithms for generating the seed set of size say P.

Random: This algorithm randomly selects a subset of nodes of size P to be seeded.

High Degree: This algorithm selects the P highest degree nodes on the graph. Here, degree for node v is defined as the sum of the trust values on all outgoing edges of v:

$$degree(v) := \sum_{(v,u)\in E} \alpha(v, u).$$

Projected Greedy heuristic: Seeds are generated according to the Projected Greedy heuristic described in Section 3.3.

3.4.5 Parameters

For all our experiments we use $\lambda_s = 0$. This means nodes always chose the maximum value while combining information from the same source. We perform simulations over several values of $\lambda_d = 0.0, 0.05, 0.1, 0.2$.

Also, in real life scenarios communication between nodes is not likely to succeed every time. Hence, when a node that is not source queries or passes information set to other nodes, it succeeds with probability $p_t = 0.75$.

3.5 Results and Discussion

We run each simulation for 50 steps and repeat it 100 times. It is observed that 50 steps are enough for the diffusion process to conclude. Since the networks are generated randomly, we repeat the whole simulation on at least 10 instances of the graph. The resulting variance as we show, is very small. We use average value of nodes evacuated in each case.

3.5.1 Seed Size

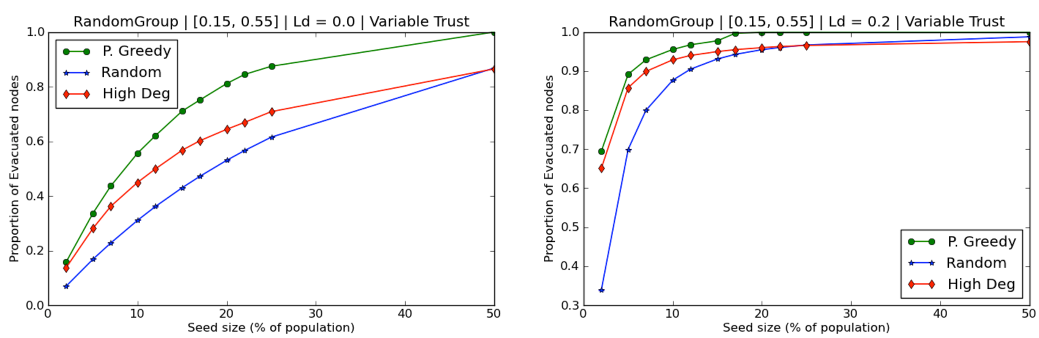


Figure 3.3: Random Group network, Group variable trust, $[0.15, 0.55]$, $\lambda_d = 0.0$ vs $\lambda_d = 0.2$ over various seed sizes. Note that for plot on the right, the y-axis range is $[0.0, 1.0]$ while for the right plot it is $[0.3, 1.0]$.

We start with discussing performance of the three seeding strategies over a range of seed sizes. Since it will be cumbersome to show all results, we show plots for two sets of parameter values in Figures 3.3, 3.4 and discuss the findings in this section. It should be noted that these results are representative and largely hold for parameter values that are not shown here.

Figure 3.3 shows plots for the Random Group model with node threshold $[0.15, 0.55]$ and Group Variable trust. The graphs contain results of simulation for initial seed sizes being the following percentage of total nodes:

$$\{2, 5, 7, 10, 12, 15, 17, 20, 22, 25, 50\}.$$

The plots compare performance of seeding strategies for $\lambda_d = 0.0$ and $\lambda_d = 0.2$. Note that the plot contains error bars for variance over all observations but they are too small to be visible, i.e., variance for our simulation results is very low.

The first observation we make is that the Γ_G function behaves like a monotonically increasing function of seed size. Also, its shape is concave which means in an average case, it behaves like a submodular function for all three types of networks. The Projected Greedy heuristic performs consistently better than both Random and High Degree seeding strategies. When $\lambda_d = 0.0$, Projected Greedy and High Degree evacuate about the same number of nodes for low values of seed size (2%). But as the seed size increases, so does the gap between their performance. Most likely, this happens due to the fact that High Degree ends up seeding nodes that are distance wise close to, and hence already converted by its other selections. It may also be the reason why Random strategy seems to eventually overtake High Degree at seed sizes close to 50%. This crossover point for Random vs High Degree occurs much earlier for $\lambda_d = 0.2$. It, most likely, is the result of the fact that having an even spread of seeded nodes is more beneficial when intermediate nodes can sum information coming from two or more seeded nodes. This advantage is diminished when there are not enough seeds to fuel the process of information gain due to summing. Hence, High Degree and Projected Greedy heuristic perform much better than Random when seed size is small, but Random catches up fast as seed sizes increase.

Figure 3.4 shows results for the same set of parameters but for San Diego network. As we can see, the results are very similar to the Random Group model. The increase in performance of Random strategy as compared to High Degree at 50% seeds is slightly more in this case. This is probably due to the structure of San

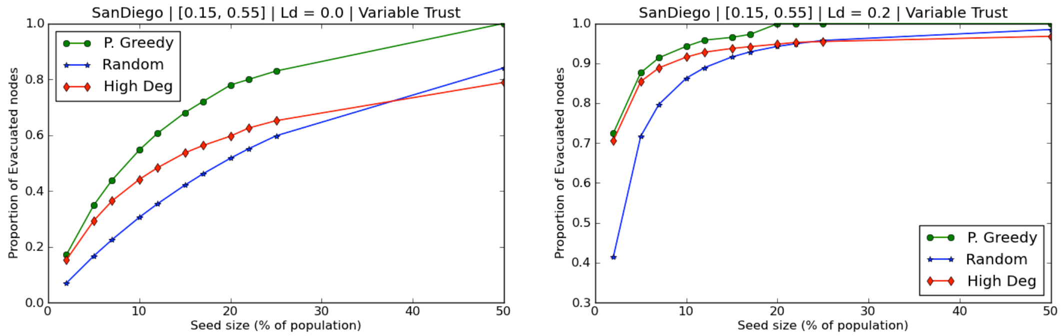


Figure 3.4: San Diego network, Group variable trust, $[0.15, 0.55]$, $\lambda_d = 0.0$ vs $\lambda_d = 0.2$ over various seed sizes. Note that for plot on the right, the y-axis range is $[0.0, 1.0]$ while for the right plot it is $[0.3, 1.0]$.

Diego network which has a number of small dense clusters that weakly connected to each other. This makes selecting high degree nodes from he same cluster a bad seeding strategy eventually worsening the performance of High Degree as compared to Random.

From both figures we observe that the Projected Greedy has significant performance advantage for $\lambda_d = 0.0$. For this case, information does not aggregate as it traverses through the network; in fact information attenuates as trust values are less than 1.0. As a result seeding strategies like Projected Greedy that take into account global structure of network have a striking advantage. Though as λ_d value increases, the amount of information in the network may actually increase making it easier to breach higher threshold of nodes. Hence comparative performance of naive strategies like Random and High Degree improves with increasing λ_d. It should also be noted that in the case where $\lambda_d = 0.2$, the Projected Greedy strategy manages to convert almost every node to believed state with seed set size al low as 20%.

3.5.2 Cross section of Results

In this section we discuss results of simulations over a wide range of parameters. Table 3.1 contains results of simulation for all three type of networks and both trust scenarios. It contains results for node thresholds $[0.2, 0.3], [0.15, 0.55]$ and $\lambda_d =$ 0.0, 0.2 since they closely represent of the behavior seen for the omitted parameters.

Table 3.1: Cross section of results. Relative performance drop in % as compared to the best seeding strategy. 0.00 indicates best performance for the scenario.

			Homogenous Trust			Variable Trust		
			R	HD	PG	R	HD	PG
SF	[0.2, 0.3]	$\lambda_d : 0.0$	25.76	1.33	**0.00**	4.32	0.08	**0.00**
		$\lambda_d : 0.2$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	[0.15, 0.55]	$\lambda_d : 0.0$	70.76	3.23	**0.00**	71.00	3.20	**0.00**
		$\lambda_d : 0.2$	25.75	1.33	**0.00**	14.39	0.63	**0.00**
RG	[0.2, 0.3]	$\lambda_d : 0.0$	31.85	11.16	**0.00**	16.21	3.87	**0.00**
		$\lambda_d : 0.2$	2.40	2.03	**0.00**	2.77	2.30	**0.00**
	[0.15, 0.55]	$\lambda_d : 0.0$	46.46	10.94	**0.00**	50.06	16.48	**0.00**
		$\lambda_d : 0.2$	32.20	10.64	**0.00**	21.68	3.87	**0.00**
SD	[0.2, 0.3]	$\lambda_d : 0.0$	31.85	11.02	**0.00**	14.09	2.91	**0.00**
		$\lambda_d : 0.2$	2.68	2.33	**0.00**	3.32	2.76	**0.00**
	[0.15, 0.55]	$\lambda_d : 0.0$	47.28	11.39	**0.00**	52.14	15.60	**0.00**
		$\lambda_d : 0.2$	32.19	10.52	**0.00**	18.16	2.41	**0.00**

The results are presented in a regret table format which helps us compare the relative performance of the seeding strategies. An entry in the table is calculated as follows:

$$val = \frac{\Gamma_{best} - \Gamma}{\Gamma_{best}} \times 100.$$

where Γ_{best} is the number of nodes evacuated by the best performing seeding strategy and Γ is the number of nodes evacuated by seeding strategy given in the column. An entry of 0.0 thus indicates that the seeding strategy has performed the best. All shown results are calculated for seed size being 5% of the total nodes.

In general when t_h is high, the proportion of nodes converted to Believed state is smaller. Same is the case with λ_d values; when $\lambda_d = 0.0$, a smaller proportion of nodes are converted to Believed state as compared to the case when $\lambda_d = 0.2$. The Projected Greedy strategy either performs better than or matches the performance of Random and High Degree seeding strategies in all cases. The performance advantage of Projected Greedy is more pronounced for the cases when $\lambda_d = 0.0$. This observation is in accordance to results shown in Figures 3.3, 3.4. This phenomenon most likely occurs due to the fact that information aggregation takes place as λ_d increases which makes it easier to convert nodes. When t_h is low, breaching this threshold is even easier. So in the cases where nodes have thresholds $[0.2, 0.3]$ and $\lambda = 0.2$, the differences in performances of the three seeding strategies is negligible, though Projected Greedy still performs better. Conversely, when $t_h = 0.55$, relative performance of Projected Greedy is much better. The relative performance loss of High Degree strategy may not seem very high $(10 - 15\%)$ but note that this gap increases with the percentage of nodes seeded.

Table 3.2: More nodes are converted on Scale-free networks. Proportion of nodes converted to Believed state.

		Homogenous Trust			Variable Trust		
		R	HD	PG	R	HD	PG
$[0.15, 0.55]$ $\lambda_d : 0.0$	SF	20.82	68.89	71.19	20.65	68.92	71.20
	RG	22.26	37.03	41.58	16.85	28.17	33.73
	SD	22.19	37.30	42.09	16.72	29.48	34.93

In Scale-free networks, Projected Greedy and High Degree perform almost identically. This is not surprising since Scale-free graphs have power-law degree distribution and diameter of these networks is small as well. So by selecting high degree nodes, information reaches a large number of nodes. The Random strategy though is thoroughly outperformed. This is corroborated by the fact that the number of nodes converted in Scale-free network, for the same number of seeds, is higher as compared to the other networks. The absolute values of fraction nodes converted are shown in Table 3.2 for one set of parameter values.

Also, not shown in the table is the effect of increasing λ_d values. As its value increases beyond $\lambda_d = 0.2$, the effect of summing information quickly takes over and there is little difference between the performance of the 3 algorithms as almost all nodes are converted to Believed state.

3.5.3 Threshold Selection

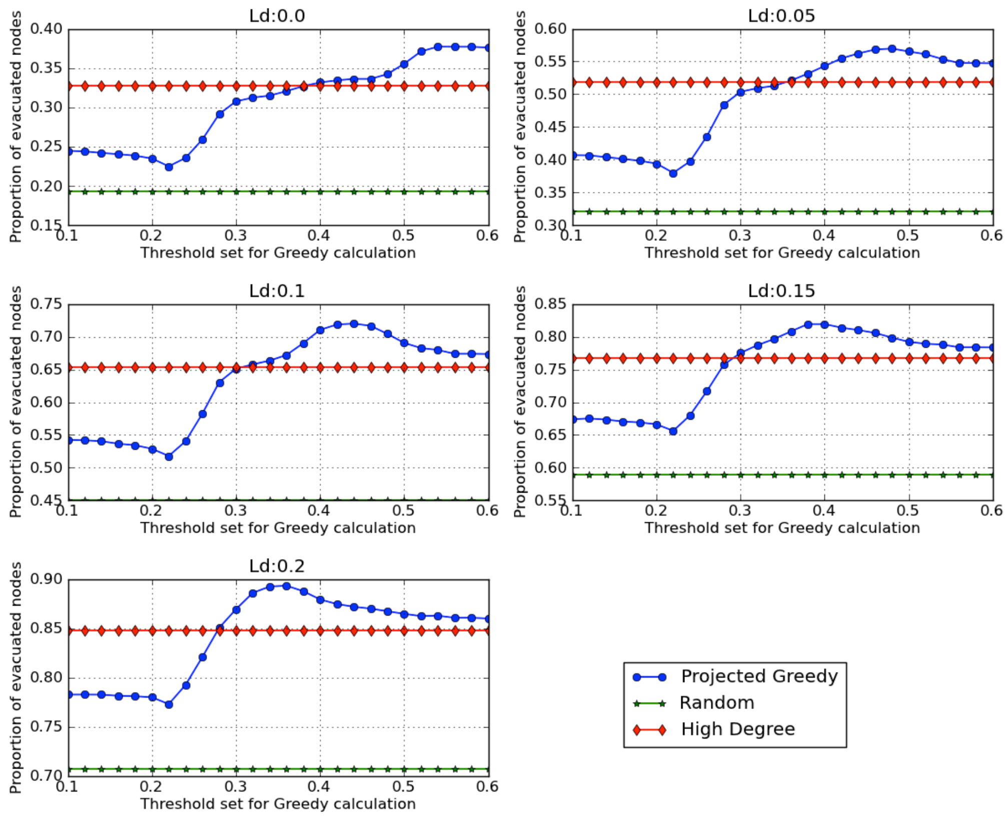


Figure 3.5: Plots showing change in optimal threshold for Projected Greedy as λ_d increases. The x-axis shows threshold selected for calculating Projected Greedy seed set. y-axis shows the proportion of nodes evacuated by that seed set. The plots are labeled with the λ_d values used for simulation.

As described in Section 3.3, in order to select the best seed set we carry out simulations over a range of values for threshold t in the simplified model. Specifically we repeat the simulation c times. It will be interesting to see how the optimal threshold, say t_{opt}, is affected by parameters of the general model like information fusion parameters (λ_s, λ_d), the lower and upper node thresholds (t_l, t_u) and probability of transmission p_t. Our simulation results show that there is an interesting relationship between t_{opt} and t_u under different values of λ_d. In order to observe this relationship, we perform simulations on a generalized version of the San Diego network. In this

Table 3.3: $\mathbf{t_{opt}}$ decreases as $\boldsymbol{\lambda_d}$ increases.

$\boldsymbol{\lambda_d}$	$\mathbf{t_{opt}}$
0.0	0.54
0.05	0.48
0.1	0.44
0.15	0.38
0.2	0.36

generalized version, edge trust values and node thresholds are selected uniformly at random from a range instead of being fixed to specific values. Thus the generalized network tries to incorporate variations observed in real networks. The network used for simulation has the following parameter values. For edges between nodes belonging to the same group, trust values selected uniformly at random from the range $[0.7, 0.8]$. For edges between nodes belonging to different groups, trust values are selected uniformly at random from the range $[t_{low}-0.5, t_{low}+0.5]$. Here t_{low} is selected such that the expected value of edge trusts in the network is 0.7. Similarly, for every $u \in V$, $t_l(u)$ is selected uniformly at random from range $[0.1, 0.2]$ and $t_u(u)$ from range $[0.5, 0.6]$. For the probability of transmission $p_t = 0.75$ and seed set size 5% of total nodes, Figure 3.5 shows the proportion of nodes evacuated with threshold values for Projected greedy in range $[0.1, 0.6]$ and $\lambda_d = [0.0, 0.05, 0.1, 0.15, 0.2]$.

It is intuitive to expect that t_{opt} will be close to $E[t_u]$ since nodes get converted to Believed state only after information value crosses t_u. This is exactly what we see in the case where $\lambda_d = 0.0$. The optimal threshold t_{opt} is very close to $E[t_u] = 0.55$. But as λ_d increases we see a gradual decrease in the value of t_{opt}. As λ_d increases, information gets aggregated and the fused information value becomes progressively more successful at breaching t_u. In other words,the general model starts behaving like a simple model with a smaller t_u value where, for the same amount of initial information, it is comparatively easier to convert nodes. Table 3.3 shows how t_{opt} value reduces with increasing λ_d. The rate at which t_{opt} decreases may depend upon factors such as the type of network (i.e. network structure), t_l etc.

CHAPTER 4
Strategic Pricing in Next-hop Inter-domain Routing

4.1 Network Model and Pricing Game Formulation

Our routing is destination-based, and therefore we consider routing of flows to a specific destination node (edge network), denoted by t, where the flow can originate at multiple source nodes (edge networks). The network used to forward flows towards the destination under consideration is modeled as a directed acyclic graph $G = (V, E)$ containing a special sink node t, and edge capacities c_e. The multi-provider structure of the Internet is captured by partitioning the nodes of this graph into sets $S_1, S_2, \ldots, S_k$, such that the set of nodes S_i, as well as all the links between nodes of S_i, are owned by ISP[2] i. Destination node t is assumed to belong to an ISP of itself. All ISPs, except for the one to which the destination belongs, act as "players" in the pricing game, choosing the forwarding paths and prices in order to serve their own interests. Throughout the chapter, therefore, the term "player" refers to one of these non-destination ISPs. Thus, in our model, the graph G is actually a "network-of-networks", where each subnetwork S_i is a "cloud" (or "domain") owned and operated by a separate player (ISP), and edges connecting S_i with S_j are inter-domain links where pricing negotiations between these players (ISPs) occur. It is also through these inter-domain links that ISPs direct flows toward each other.

To arrive at a meaningful model of price competition, we assume that each subnetwork S_i has enough capacity to forward any amount of traffic that enters it.

Portions of this chapter previously appeared as: E. Anshelevich, A. Hate, and K. Kar. 2011. "Strategic pricing in next-hop routing with elastic demands". In *Proceedings of the 4th international conference on Algorithmic game theory* (SAGT'11), Giuseppe Persiano (Ed.). Springer-Verlag, Berlin, Heidelberg, 278-289.

[2]We use the terms "domain", "AS" and "ISP" interchangeably. While the terms domain and ISP are often used loosely, these terms are not strictly equivalent. For example, an ISP can control multiple ASes, while there can be customer ASes that are not a part of any ISP [120]. In our framework/model, these terms represent a generic network "cloud" that is under a single administrative control, and the basic unit over which routing/traffic engineering and forwarding pricing decisions are taken.

More precisely, we assume that, if all the links entering S_i are saturated, then there is enough capacity on links inside S_i and on links leaving S_i to be able to forward all of this incoming traffic. This condition would arise naturally if there is a large penalty for not forwarding flow that is sent to an ISP i, giving incentive to build more internal infrastructure or reduce incoming capacity. Moreover, as is common in these types of pricing models [90, 3, 112], we make a *non-monopolistic* assumption that, even if we remove a subnetwork S_j from the graph, there would still be enough capacity on links leaving S_i to be able to forward all of its incoming traffic. As discussed below, this prevents ISP j from charging exorbitant prices to ISP i for forwarding i's traffic, which i would have to pay because without S_j, there would not be enough capacity available. Although the non-monopolistic assumption is necessary for our theoretical results to hold, as we show in Section 4.5, price dynamics still behave extremely well even without this assumption.

Other than forwarding each other's traffic, ISPs also generate and forward traffic of their own, which originate from the end-users that the ISP directly serve. In our model, every subnetwork S_i contains a special node w_i where the traffic of ISP i originates. This node represents the ISPs own customer network, that serves the end-users that get direct service from the ISP. Additionally, every ISP i has an associated *source utility* λ_i. This means that if a ISP i sends f_i amount of its own flow (flow originating form vertex w_i) to the sink t, then the ISP will obtain a utility of $\lambda_i \cdot f_i$.

4.1.1 ISP Traffic Pricing and Forwarding Strategy

The core behavior of our model is that ISP i can charge a price on every edge entering S_i. For example, for an edge $e = (u, v)$ with $u \in S_j$ and $v \in S_i$, player i can set some price p_e. In this case, if ISP j sends flow f_e on this edge, then it has to pay $f_e \cdot p_e$ to ISP i. (Our theoretical results also hold for the case where the price per unit flow forwarded changes with the amount of flow being sent on the edge and we will consider this in Section 4.4.) Since the destination ISP is not a player, it does not set prices for edges that are incident on it, and we assume those prices are fixed at 0. Figure 4.1 overviews the pricing and forwarding scheme from the perspective

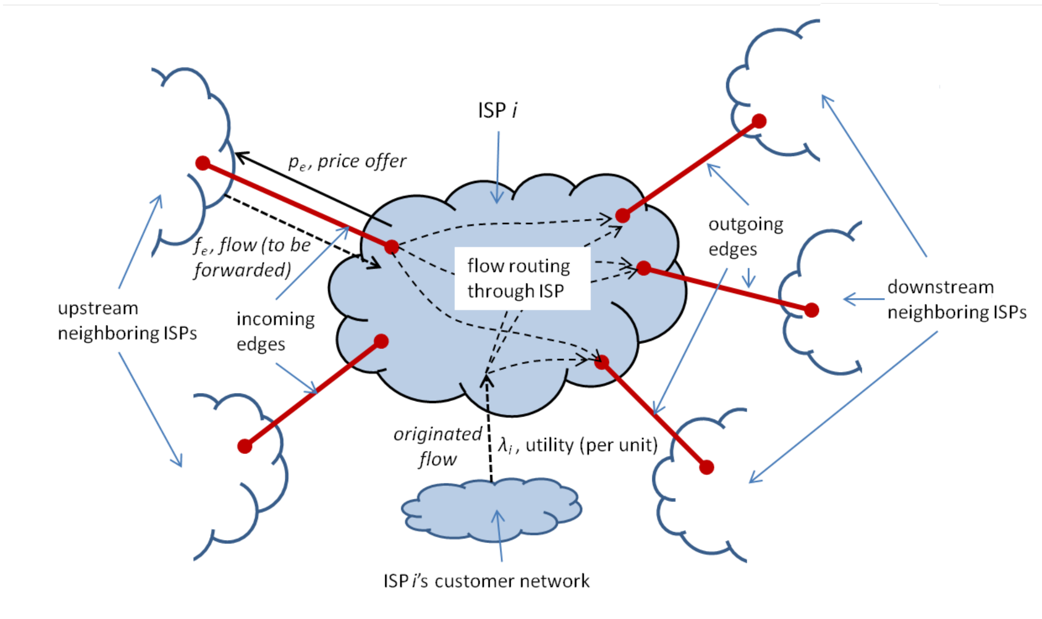


Figure 4.1: Price setting, flow generation and forwarding by ISP i.

of ISP i.

So apart from gaining utility by sending their own flow, ISPs also gain utility for receiving flows from neighboring ISPs and lose utility by paying the next-hop ISPs that receive their flow. Also, observe that for the flow to reach the destination, it is essential that intermediate players forward the flow reliably. Hence every player is required to forward all incoming flow (alternatively, we can think of there being a very large penalty for accepting payment for incoming flow that the player has no intention of forwarding). The forwarding of all flow is always possible due to our assumptions about graph capacities.

Consider now the incentives of an ISP i once the prices p_e on inter-domain links have been set. Let E_i^{in} and E_i^{out} be the set of incoming and outgoing edges for ISP i's subnetwork respectively, and let f_i be the amount of traffic that i sends, i.e., traffic bound for t originating at node w_i. Then, ISP i's utility is

$$\sum_{e \in E_i^{in}} f_e \cdot p_e - \sum_{e \in E_i^{out}} f_e \cdot p_e + \lambda_i \cdot f_i. \tag{4.1}$$

Thus, once the prices are set, and ISPs upstream of ISP i choose how much traffic to send on edges E_i^{in}, ISP i's goal is simply to choose how to forward incoming traffic through its subnetwork S_i (as well as how much of its own traffic to send), in order to maximize the above expression. To do this, the ISP may have some internal traffic engineering policy (e.g., min-hop routing). We assume that all ISPs forward traffic, and decide how much of their own traffic to send, in the optimal way; in other words, they choose the alternative which maximizes expression (4.1), while following the constraints of their internal routing policy.

There is one detail to add to our model definition: what if several possible flows yield the same highest utility for an ISP i? Which of these alternatives would ISP i choose, i.e., what is the tie-breaking rule? In our model, we leave this up to the player, instead of forcing some tie-breaking rule. In other words, the strategy of ISP i is (p_i^{in}, γ_i), where p_i^{in} is a vector of prices p_e for each edge in E_i^{in}, and γ_i is an arbitrary tie-breaking rule that decides between several possible flows of the same quality. In other words, ISP i's first priority when deciding where to forward traffic and how much of its own traffic to send, is to maximize its utility. If there are several flows that maximize its utility, it breaks ties according to the tie-breaking rule γ_i. We make the following reasonable assumption about the behavior of internal routing policies and tie-breaking rules: *If the price on an edge (u, v) with $u \in S_i$ remains constant, while the prices on other edges of E_i^{out} increase, then the flow on edge (u, v) does not decrease.* In other words, my competitors charging higher prices only causes me to get more flow, not less.

We denote the collective strategy of all ISPs by $\{P, \gamma\}$. Now that we defined the ISP strategies, consider the outcome of the game, i.e., the flow of traffic that would be generated given ISP choices $\{P, \gamma\}$. To compute this flow, simply iterate over S_i in topologically sorted order (recall that our graph is a DAG). For each i, the incoming and outgoing prices are known, as well as the incoming flow on edges E_i^{in}. Thus, we assume that ISP i solves the above optimization problem (using γ_i to break ties in case several flows are optimal for it), and thus generates the flow inside the subnetwork S_i, as well as on the edges E_i^{out}. This process generates a unique flow; we denote the resulting flow by $f(P, \gamma)$, as this is the outcome of the strategy $\{P, \gamma\}$. The final utility of an ISP i given the choices of all ISPs, $\{P, \gamma\}$, is

$$utility_i(P, \gamma) = \sum_{e \in E_i^{in}} f_e \cdot p_e - \sum_{e \in E_i^{out}} f_e \cdot p_e + \lambda_i \cdot f_i,$$

where f_e is the flow on edge e in $f(P, \gamma)$, and f_i is the flow originating at node w_i in $f(P, \gamma)$.

Consider an ISP i who is computing its best response to a strategy $\{P, \gamma\}$. Notice that by changing its prices p_i^{in}, the resulting flow f_i^{in} may become completely different from $f(P, \gamma)$. If this were not the case, then i could always raise its incoming price, knowing that this would increase its utility since the flow would remain the same. In essence, players in this game (ISPs) anticipate changes in flow that result from price changes, but myopically assume that the prices of all other players remain the same when computing their own best response. Such behavior is reasonable in ISP routing settings, for example, since price setting takes place on a much slower time scale than routing.

4.1.2 Fixed Tie-Breaking Rules

When defining our model, we leave the tie-breaking rules γ_i as a player (ISP) choice. As we discuss in Section 4.3, our model has several nice properties, including existence of Nash equilibrium that is as good as a centrally optimal solution. Instead, however, we could have simply assumed a reasonable tie breaking rule for all ISPs. Unfortunately, as the following theorem shows, this lack of choice of the ISPs on the tie-breaking rules leads to instability.

Theorem 11 *If tie-breaking rules γ_i were assumed to be fixed, instead of part of a ISP's strategy, then there are networks where all Nash equilibria are arbitrarily bad compared to the socially optimal solution.*

Proof. We show that the example in Figure 4.2 has arbitrarily high price of stability when tie-breaking rules γ_i are assumed to be fixed. In the example, all edges have capacity 1. Every node is a separate player, i.e., the size of each set S_i equals 1. Thus, for the purposes of this example, we can refer to nodes instead of players.

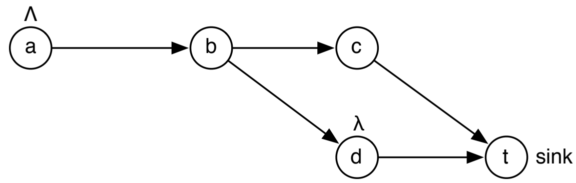


Figure 4.2: Example showing arbitrarily large price of stability for fixed tie-breaking rules.

Node a has source utility of Λ and d has source utility of λ. The other nodes have source utilities of 0. It is easy to verify that the optimal solution for this example consists of flow of size 1 on path $a \to b \to c \to t$ and a flow of size 1 on edge (d, t). This optimal solution gives a social welfare (i.e., sum of player utilities) of $(\Lambda + \lambda)$. Node b has a fixed tie-breaking rule γ_b where it prefers sending to player d over player c, whenever it has several ways of forwarding flow that results in the same utility for b.

Let any Nash equilibrium solution for this instance in which node a sends some of its own flow be called a *good* equilibrium. We first show that this instance does not admit a good equilibrium. In order to prove this by contradiction, suppose that such an equilibrium does exist and $f(a, b) > 0$. We denote price of edge (b, c) as p_c and that of edge (b, d) as p_d.

If $p_d = 0$ then the preference of edge (b, d) over edge (b, c) ensures that $f(b, d) > 0$. But now node d has a profitable deviation where it can increase p_d to λ thereby receiving more utility by either sending its own flow or forwarding flow by charging a higher price. Hence $p_d > 0$. Now consider the values that p_c can take.

- $p_c < p_d$: We have already shown that $p_d > 0$. Let $p_c = p_d - \delta$ where $\delta < p_d$. In this case, node b will forward flow $f(a, b)$ on edge (b, c). Now consider the deviation for node c where it increases p_c to $p_d - \delta/2$. Node b will still forward the flow $f(a, b)$ on edge (b, c), which implies utility of node c would increase, making it a profitable deviation.

- $p_c \geq p_d$: Again, the preference of edge (b, d) over edge (b, c) ensures that node b will forward flow $f(a, b)$ on that edge (b, d). This means utility of node $c = 0$. Now consider the deviation for node c where it decreases p_c to $p_d - \epsilon$

where $\epsilon < p_d$. Now flow $f(a,b)$ will be routed to edge (b,c) giving a utility of $f(a,b) \cdot p_c$ to node c. Since $f(a,b)$, $p_c > 0$, this is a profitable deviation for node c.

Therefore there does not exist a stable price for edge (b,c) when $f(a,b) > 0$, and hence a good equilibrium does not exist for this instance. Now consider a strategy ψ, where all edge prices are $\geq \Lambda$ (except for edges to t, which always have price 0 since t is not a player). In this solution node d sends its own flow and the social utility is λ. It is easy to verify that this strategy is in Nash equilibrium. Since good equilibria do not exist and d is the only other node with positive source utility, ψ is a Nash equilibrium strategy with the highest social utility. Hence:

$$\text{Price of Stability} = \frac{\Lambda + \lambda}{\lambda}.$$

Thus if $\Lambda \gg \lambda$ then price of stability can be arbitrarily large, and any existing Nash equilibrium is much worse than the social optimum. The above example can be adjusted such that, with fixed tie-breaking rules, no Nash equilibrium exists.

Note that, if instead of having fixed orderings, b were able to pick its ordering γ_b to prefer node c over d, then prices of Λ on (a,b) and λ on (b,c) and (b,d) result in a Nash equilibrium with the same quality as the social optimum. ■

4.1.3 Well-Provisioned Networks

For most of our theoretical results, we need the assumption that internal subnetworks (individual ISP networks) are *well-provisioned*. To be well-provisioned, a network S_i has to be such that, given an incoming flow f_i^{in} and outgoing prices p_i^{out}, the flow that would maximize i's utility would simply be to forward flow on the cheapest outgoing edges, and to send its own flow on outgoing edges with left-over capacity as long as their price p_e is less than λ_i. In other words, a network is well-provisioned when the solution to each ISP's flow optimization problem (i.e., how the ISP should split its traffic among its downstream neighbors) does not need to take the internal capacity into account, since it is large enough to handle the traffic. For example, if each internal edge of S_i had capacity at least as large as the total capacity of E_i^{out}, then S_i is automatically well-provisioned.

While we will assume that the ISP networks are well-provisioned in order to prove good theoretical properties, our simulations show that, even without this assumption, the price dynamics behave extremely well, and converge to solutions with good social welfare.

4.2 Price-Flow Equilibrium Analysis

In this section, we show that for general networks, Nash equilibrium may not exist. In later sections, we consider the special case when all ISP networks are well-provisioned. We will show that in this case, good Nash equilibria always exist, and give several theoretical characterizations.

Theorem 12 *For our pricing game, there are networks which admit no Nash equilibrium.*

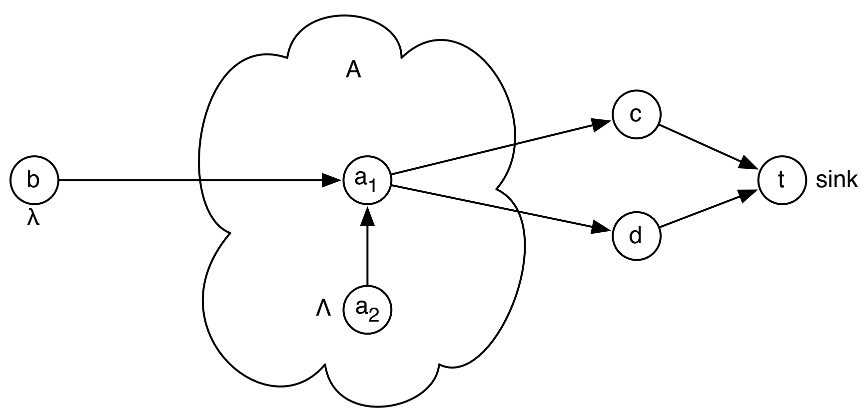


Figure 4.3: Example showing that Nash Equilibrium may not exist in the cloud model.

Proof. In the example illustrated in Figure 4.3, player A controls nodes a_1 and a_2; the rest of the nodes are each controlled by a separate player. Node a_2 has a source utility of Λ. Player b has source of utility of λ, where $\lambda < \Lambda$. Players c and d do not have any source utility. All edge capacities are 1.

We will use the following notation throughout. Price of edge (b, a_1) will be denoted by p_b and the quantity of flow on the edge will be denoted by f_b. Similarly, the price of edges (a_1, c) and (a_1, d) will be denoted by p_c and p_d respectively whereas the quantity of flow will be denoted by f_c and f_d. We will mark the difference in the values of these quantities after a strategy deviation by using $'$. For example the flow on edge (a_1, c) after deviation will be denoted by f'_c. Utility of a player x will be denoted by $u(x)$.

Unless otherwise mentioned, the quantity ϵ is an infinitesimally small amount greater than 0. This also implies $(\Lambda - \lambda) \gg \epsilon$. The value ϵ is typically used to avoid invoking the tie-breaking rule. For example, if player A wants to draw a flow of size 1 from player b, the maximum price that can set on edge (b, a_1) is $p_b = \lambda$. At this price though the tie breaking rule comes into play that may result in $f_b < 1$. In such a case, player A sets p_b to $\lambda - \epsilon$ which guarantees flow of size 1.

Now let us assume that this example accepts a Nash equilibrium solution. Observe that only nodes a_2 and b send their own flow. Consider the following cases for the flow generated in the Nash equilibrium solution:

Case 1: Neither a_2 nor b send their own flow

Since a_2 is not sending its own flow, it must be true that p_c, $p_d \geq \Lambda$. In this situation player c can reduce price on edge (a_1, c) so that $p'_c = \Lambda - \epsilon$. This would prompt player A would send its own flow form node a_2 on edge (a_1, c). This would result in player c's utility increasing from 0 to $\Lambda - \epsilon$ making it a profitable deviation.

Hence there does not exist a Nash equilibrium strategy where neither a_2 no b send their own flow.

Case 2: b sends own flow but a_2 does not

Since b is sending its own flow, $p_b \leq \lambda$. Utility of player A is as follows:

$$u(A) = f_b \cdot p_b - f_c \cdot p_c - f_d \cdot p_d.$$

Now consider a deviation for player A where she increases price on edge (b, a_1) such that $p'_b > \lambda$. This would result in reducing f'_b to 0. Player A then replaces the lost flow from b by sending its own flow from node a_2 and sends the same quantity of flow to nodes c and d. Player A's new utility is:

$$
\begin{aligned}
u(A)' &= f_b \cdot \Lambda - f_c \cdot p_c - f_d \cdot p_d \\
&> f_b \cdot \lambda - f_c \cdot p_c - f_d \cdot p_d \\
&\geq f_b \cdot p_b - f_c \cdot p_c - f_d \cdot p_d \\
&= u(A).
\end{aligned}
$$

This is a valid deviation for player A, hence there does not exist a Nash equilibrium strategy where b sends own flow but a_2 does not.

Case 3: **a_2 sends own flow but b does not**

We claim that in this case $p_c = p_d = 0$. Note that the maximum flow that a_2 can send is of size 1. This implies that if $f_d > 0$ then $f_c < 1$ and vice versa. Let us assume that $f_d > 0$. This also means that $p_d \leq p_c$. Now consider the following implications of this assumption:

1) If $p_d > 0$ and $p_c > p_d$ then $u(c) = 0$ as it will not receive any flow. But player c can change price of edge (a_1, c) to $p_c' = p_d - \epsilon$, thereby obtaining flow of size f_d and utility $(p_d - \epsilon) \cdot f_d$.

2) If $p_d > 0$ and $p_c = p_d$ then utility of player c is $u(c) = p_c \cdot f_c = p_d \cdot f_c$. Now player c can change price of edge (a_1, c) to $p_c' = p_d - \epsilon$ for $0 < \epsilon < \frac{p_d \cdot f_d}{f_c + f_d}$. This would imply that flow f_d would now be routed to c and its new utility will be:

$$
\begin{aligned}
u(c)' &= (p_d - \epsilon) \cdot (f_d + f_c) \\
&> \left(p_d - \frac{p_d \cdot f_d}{f_c + f_d}\right)(f_d + f_c) \\
&= p_d \cdot f_c \\
&= u(c).
\end{aligned}
$$

3) If $p_d = 0$ and $p_c > p_d$ then d would increase price on edge (a_1, d) to $p_d' = \epsilon$ such that $0 < \epsilon < p_c$ and gain utility of value $\epsilon \cdot f_d$ which is more than her previous utility: 0.

4) Since all the above cases have valid deviations, Nash equilibrium can exist only if $p_c = p_d = 0$.

The above argument holds symmetrically for the case where $f_c > 0$. This proves our claim that $p_c = p_d = 0$.

Consider the utility of player A: $u(A) = \Lambda(f_c + f_d)$. Since b is not sending any of its own flow, $p_b \geq \lambda$. We have already shown that $p_c = p_d = 0$ and $f_c + f_d \leq 1$. This means player A can increase its utility by sending the complete 1 unit of flow from node a_2 and setting $p'_b = \lambda - \epsilon$. This will result in b sending its flow f_b of size 1. The resulting utility of player A: $u(A)' = \Lambda \cdot 1 + (\lambda - \epsilon) > u(A)$. Which means this would be a valid deviation for player A.

Hence there does not exist a Nash equilibrium strategy where a_2 sends own flow but b does not.

Case 4: Both a_2 and b send their own flow

We first look at the case where $f_c + f_d < 2$. Again, we claim that in this case $p_c = p_d = 0$. A similar analysis as the one in **Case 3** can be carried out. It is easy to see that if $f_c < 1$ then $p_c \geq p_d$ and vice versa. Let us assume that $f_c < 1$. Now consider the following implications of this assumption:

1) Consider the case where $p_d > 0$ and $p_c > p_d$. If $f_c = 0$ then the same argument as given in **Case 3.1** holds.

   If $f_c > 0$ then $f_d = 1$. Let $p_d = p_c - \delta$ where $\delta < p_c$. Now player d can set $p'_d = p_c - \delta/2$, thereby obtaining the same amount of flow at a higher price and hence increase its utility.

2) If $p_d > 0$ and $p_c = p_d$ then the argument of **Case 3.2** holds exactly.

3) If $p_d = 0$ and $p_c > p_d$ then d would increase price on edge (a_1, d) to $p'_d = \epsilon$ such that $0 < \epsilon < p_c$ and gain utility of value $\epsilon \cdot f_d$ which is more than her previous utility: 0.

4) Since all the above cases have valid deviations, Nash equilibrium can exist only if $p_c = p_d = 0$.

Given $p_c = p_d = 0$, player A will send all of its own flow from node a_2 if it previously wasn't and also set price $p'_d = \lambda - \epsilon$ to draw a flow of size 1 from player b. It is easy to see that this is a beneficial deviation for player A. This results in $f'_c + f'_d = 2$.

Finally, we consider the case where $f_c + f_d = 2$. Since player b is sending its own flow, $p_b \leq \lambda$. If $\min(p_c, p_d) > \lambda$, then this cannot be a Nash equilibrium, since A can gain utility by increasing the price p_b, and reducing the flow f_b. This is because if $\min(p_c, p_d) > \lambda$, then A is losing utility from flow received from b. Thus, in every Nash equilibrium, it must be that $\min(p_c, p_d) \leq \lambda$.

W.l.o.g. let $p_c = min(p_c, p_d) \leq \lambda$. Then, player c can gain utility by setting $p'_c = \Lambda - \epsilon$. Player A will still send its own flow of size 1 and is obligated to forward flow of size 1 which it receives from b. This clearly is a beneficial deviation for player c as its utility jumped from being $\leq \lambda$ to $\Lambda - \epsilon$. Thus, no Nash equilibrium exists when $f_c + f_d = 2$.

Conclusion The four considered cases are the only possible configurations of flow. Since a Nash equilibrium strategy cannot exist in any of these configurations, there does not exist one for this example. ■

Due to the above result, in Section 4.3 and Section 4.4, we assume that subnetworks S_i are well-provisioned.

4.3 Well-Provisioned Networks: Equilibrium Characterization

When a subnetwork is well-provisioned, a player i's incentives are equivalent to the case when the entire subnetwork S_i consists of a single node. Also, this enables us to make more precise observations about the behavior of players in the game. For example we can define a notion of valid flows, which are flows that every player forwards in order to maximize its utility. Specifically, given the prices p_v^{out} and flows f_v^{in}, we define the set of valid resulting flows f_v^{out} to be $\mathcal{F}_v(f_v^{in}, p_v^{out})$, which are all flows satisfying the following conditions:

- $\forall e \in E_v^{out} : f_e \leq c_e$ (usual capacity constraint);
  $f_v = \sum_{e \in E_v^{in}} f_e - \sum_{e \in E_v^{out}} f_e \geq 0$ (usual flow conservation)

- $\forall e \in E_v^{out} : f_e > 0$ only if for every $e' \in E_v^{out} \backslash e$ with $p_{e'} < p_e$, e' is saturated (send on cheapest edges first),

- $\forall e \in E_v^{out} : p_e < \lambda_v$ implies that e is saturated (send own flow if profitable), and
  $p_e > \lambda_v$ and $f_e > 0$ imply that $f_v = 0$ (don't send own flow if unprofitable).

Any way of forwarding flow to maximize v's utility obeys these conditions. The last condition holds since if v is sending its own flow, but $f_e > 0$ for some edge $e \in E_v^{out}$ with $p_e > \lambda_v$, then v could re-distribute its flow so that it is sending its own flow on edge e, and then improve its utility by sending less of its own flow. We can also precisely define valid tie-breaking rules as follows: since each valid flow in $\mathcal{F}_v(f_v^{in}, p_v^{out})$ corresponds to a tie-breaking rule selected by player v, we associate these tie-breaking rules with a flow generation function γ_v which, given the incoming flows and out going prices, produces an outgoing flow. The set of these flow generation functions is denoted by Γ_v:

$$\Gamma_v := \{\gamma_v | \, \forall f_v^{in}, p_v^{out} : \gamma_v(f_v^{in}, p_v^{out}) = f_v^{out} \in \mathcal{F}_v(f_v^{in}, p_v^{out})\}.$$

In other words, Γ_v contains all functions that generate only valid out-flows. Hence the strategy set of each player v is $\mathcal{R}_+^{|E_v^{in}|} \times \Gamma_v$, and a strategy of the player is given by the tuple $\{p_v^{in}, \gamma_v\}$ where $p_v^{in} \in \mathcal{R}_+^{|E_v^{in}|}$ and $\gamma_v \in \Gamma_v$. We denote the collective strategy of all players by $\{P, \gamma\}$.

For well provisioned networks, we first characterize the Nash equilibrium strategies and then give an efficient algorithm that constructs a Nash equilibrium strategy that is as good as the optimal solution. This holds for an arbitrary number of sources with elastic demands of heterogenous value. We also show that in case of a single source, under some reasonable behavioral assumptions for the players, the price of anarchy is 1, and in fact player prices at equilibrium are unique. We will also consider the more general case where the source utility can be an arbitrary concave function $\Lambda_v(f_v)$, and the prices can be arbitrary convex functions $\Pi_e(f_e)$. We show that the above results still hold if arbitrary convex prices are allowed, and thus allowing non-linear prices does not impact the quality of equilibrium solutions. On the other hand, if source utilities can be non-linear functions, then we show that pure Nash equilibrium may no longer exist, even for discrete pricing models.

4.3.1 Characterization of Nash Equilibiria

We first prove useful sufficient conditions for a strategy to be a Nash equilibrium.

Theorem 13 *In the case where all ISPs are well-provisioned, if flow $f(P, \gamma)$ and prices P satisfy the following conditions, then strategy $\{P, \gamma\}$ is a Nash equilibrium:*

(a) For every node u, the price on all edges of E_u^{out}, except edge (u, t) if it exists, is the same. Let this price be denoted by y_u.

(b) If $f_u > 0$ then $y_u = \lambda_u$; if $f_u = 0$ then $y_u \geq \lambda_u$.

(c) For $\forall v \neq t$, if edge (u, v) has a positive flow on it, then $y_u \geq y_v$.

(d) For $\forall v \neq t$, if edge $e = (u, v)$ is unsaturated ($f_e < c_e$), then $y_u \leq y_v$.

We first prove the following lemmas, which state that if a player deviation includes increasing its price on some incoming edge, then no flow is sent on this edge after such a deviation.

Lemma 5 *Let $\{P, \gamma\}$ satisfy the conditions of Theorem 13. Consider a unilateral deviation by player v where v changes its strategy from (p_v^{in}, γ_v) to $(p_v^{in'}, \gamma_v')$ such that for some edge $e = (u, v)$, $p_e' > p_e$. Let $\{P', \gamma'\}$ denote the resulting collective strategy of all players. Then the flow on edge e in $f(P', \gamma')$ equals 0.*

Proof. We will let f denote the flow $f(P, \gamma)$, and f' denote the flow $f(P', \gamma')$. Consider first the case where node u is not sending its own flow in $f(P', \gamma')$, i.e., $f_u' = 0$. Since $p_e' > p_e$, condition (a) of Theorem 13 says that $\forall e' \in E_u^{out} \backslash e$, $p_{e'} < p_e'$. Also because of the non-monopolistic property we know that total capacity on the set of edges $E_u^{out} \backslash e$ is at least as much as the capacity on edges E_u^{in}. Therefore the flow generating algorithm will route all incoming flow to u on edges of $E_u^{out} \backslash e$ in $f(P', \gamma')$.

Now suppose instead that $f'_u > 0$. By condition (b), we know that $y_u \geq \lambda_u$, and thus that $p'_e > \lambda_u$. Thus no valid flow $f(P', \gamma')$ will send any flow on edge e (since node u loses utility by sending flow on edge e). Hence in $f(P', \gamma')$ there would be no flow on edge e. ■

We are now ready to prove Theorem 13.

Proof. Let $\{P, \gamma\}$ satisfy the conditions of Theorem 13, and suppose to the contrary that for some player v, there exists a strictly improving unilateral deviation from $\{P, \gamma\}$, where v changes its strategy from (p_v^{in}, γ_v) to $(p_v^{in'}, \gamma'_v)$. Denote by $\{P', \gamma'\}$ the collective player strategy after v's deviation, and let $f = f(P, \gamma)$, $f' = f(P', \gamma')$.

Let the set of incoming edges of v be partitioned into the following:

- $E_{inc} \subseteq E_v^{in}$ such that $\forall e \in E_{inc} : p'_e > p_e$.
- $E_{dec} \subseteq E_v^{in}$ such that $\forall e \in E_{dec} : p'_e < p_e$.
- $E_{free} \subseteq E_v^{in}$ such that $\forall e \in E_{free} : p'_e = p_e$ and $f_e < c_e$.
- $E_{sat} \subseteq E_v^{in}$ such that $\forall e \in E_{sat} : p'_e = p_e$ and $f_e = c_e$.

Note that $E_{inc} \cup E_{dec} \cup E_{free} \cup E_{sat} = E_u^{in}$. Now, utility of node v in strategy (P, γ) can be stated as follows:

$$utility_v(P, \gamma) = \sum_{e \in E_{inc} \cup E_{dec} \cup E_{free} \cup E_{sat}} f_e \cdot (p_e - y_v) + f_v \cdot (\lambda_v - y_v) + c_{(v,t)} \cdot y_v. \quad (4.2)$$

This is because for every unit of flow that node v is forwarding or sending, it must pay y_v to an outgoing edge, except for $c_{(v,t)}$ units of flow, which can be sent to t with price 0 (we will use the convention that $c_{(v,t)} = 0$ is edge (v, t) does not exist).

Notice that each term $f_e(p_e - y_v)$ is nonnegative due to condition (c), and the term $f_v(\lambda_v - y_v)$ equals zero due to condition (b). Similarly, the utility of v in strategy $\{P', \gamma'\}$ is the same as in Equation 4.2 with f' replacing f and p'_e replacing p_e. Note that, since we assumed that every edge (v, t) is saturated (since node v can always forward or send its own flow to fill up edge (v, t) to capacity without decreasing its utility), then the term $c_{(v,t)} \cdot y_v$ is present both before before and after the deviation. We now analyze the effect of price changes from p_e to p'_e on the sum in Equation 4.2 for each of the four subsets of E_v^{in}.

- E_{inc}: According to Lemma 5, all edges with price increase will not have any flow in $f(P', \gamma')$. Hence the term $\sum_{e \in E_{inc}} f'_e \cdot (p'_e - y_v)$ equals $0 \leq \sum_{e \in E_{inc}} f_e \cdot (p_e - y_v)$.

- E_{dec}: Since $p'_e < p_e$ for edges in E_{dec}, then the only way that it is possible for $f'_e \cdot (p'_e - y_v)$ be greater than $f_e \cdot (p_e - y_v)$ is if $f'_e > f_e$. This implies that edge e is not saturated in f, i.e, $f_e < c_e$. According to condition (d), edges in E_{dec} that have free capacity in $f(P, \gamma)$ have $p_e \leq y_v$. Thus $(p'_e - y_v) < (p_e - y_v) \leq 0$, and so it is not possible that $f'_e \cdot (p'_e - y_v) > f_e \cdot (p_e - y_v)$ because each term $f_e \cdot (p_e - y_v)$ is nonnegative. Therefore, $\sum_{e \in E_{dec}} f'_e \cdot (p'_e - y_v) \leq \sum_{e \in E_{dec}} f_e \cdot (p_e - y_v)$.

- E_{free}: For an edge $e \in E_{free}$, we know by condition (d) that $p'_e = p_e \leq y_v$. Since each term $f_e(p_e - y_v)$ is nonnegative, this implies that $0 = f_e(p_e - y_v) \geq f'_e(p'_e - y_v)$. Thus $\sum_{e \in E_{free}} f'_e \cdot (p'_e - y_v) \leq \sum_{e \in E_{free}} f_e \cdot (p_e - y_v)$.

- E_{sat}: Since these edges are saturated in $f(P, \gamma)$, then $f'_e \leq f_e$. Also from condition (c) we know that $p'_e = p_e \geq y_v$. It follows that $\sum_{e \in E_{sat}} f'_e \cdot (p'_e - y_v) \leq \sum_{e \in E_{sat}} f_e \cdot (p_e - y_v)$.

- f_v: As mentioned above, we know because of condition (b) that $f_v \cdot (\lambda_v - y_v) = 0$. The only way that $f'_v \cdot (\lambda_v - y_v) > 0$ is if $\lambda_v - y_v > 0$, which contradicts condition (b). Thus, this term equals 0 both before and after v's deviation.

Hence the change in node v's strategy can only result in a drop in utility of node v. Thus the collective strategy $\{P, \gamma\}$ is a Nash equilibrium. ■

Theorem 13 gives sufficient conditions for a strategy to be a Nash equilibrium. We will call such strategies *uniform*, since all the outgoing prices are the same for every node in such a solution. As we will show later, good uniform Nash equilibria always exist, and can be efficiently computed.

Definition 2 *Uniform Nash equilibrium: Any Nash equilibrium strategy that satisfies the conditions of Theorem 13 is a* **uniform** *Nash equilibrium.*

The above is a reasonable property to expect from a Nash equilibrium, as neighboring ISPs at the outgoing edges of a ISP i are essentially competing for ISP i's traffic. Thus, if some edges of E_i^{out} are cheaper than the others, then they may raise their prices, expecting that the incoming flow from S_i would not change. On the other hand, neighboring ISPs at more expensive edges of E_i^{out} may lower their prices in order to obtain more traffic from S_i and thus make a larger profit.

We extend this results by partially characterizing the structure of all uniform Nash equilibria in well-provisioned networks. Specifically, we give almost matching sets of necessary and sufficient conditions for the existence of such equilibrium.

Theorem 14 *If all ISPs are well-provisioned, then every uniform Nash equilibrium must satisfy properties (a)-(c).*

(a) For every player i, the price on all edges of E_i^{out}, except edges to t if they exist, is the same. Let this price be denoted by y_i.

(b) If $f_i > 0$ then $y_i = \lambda_i$; if $f_i = 0$ then $y_i \geq \lambda_i$.

(c) For all edges (u, v) with $u \in S_j$ and $v \in S_i$, if (u, v) has a positive flow on it, then $y_j \geq y_i$.

(d) For all edges (u, v) with $u \in S_j$ and $v \in S_i$, if (u, v) is unsaturated ($f_e < c_e$), then $y_j \leq y_i$.

Proof. Consider a uniform Nash equilibrium strategy $\{P, \gamma\}$; it satisfies condition (a) by definition. For the following discussion we will consider ISPs i and j and edge $e = (u, v)$ where $u \in S_i$ and $v \in S_j$. We now prove that condition (b) must hold in every uniform Nash equilibrium. First, it is clear that if $y_i < \lambda_i$, then i would saturate all outgoing edges with its own flow to maximize its utility, and thus f_i would be greater than 0. Thus, $f_i = 0$ implies that $y_i \geq \lambda_i$ in a uniform Nash equilibrium. Similarly, it is clear that if $f_i > 0$, then $y_i \leq \lambda_i$, since otherwise ISP i would not send its own flow, and thereby improve its utility. All that is left to show is that when $f_i > 0$, we cannot have that $y_i < \lambda_i$. If $y_i < \lambda_i$ then consider edge $e = (u, v)$ such that $p_e = y_i$. Here, $f_e = c_e$ since i will saturate this edge with its own flow. Note that such an edge does exist since $f_i > 0$. In this case ISP j can increase the payment it receives from i by increasing the price of edge (u, v) to be arbitrarily close to λ_i. This deviation will not affect the flow on outgoing edges of i since all outgoing prices equal $y_i < \lambda_i$, and so all outgoing edges of i would still be saturated. This means the flow in the network remains the same and utility of ISP j strictly increases. Hence strategy $\{P, \gamma\}$ would not be a Nash equilibrium unless condition (b) is satisfied.

Now let us suppose to the contrary that there exists a uniform Nash equilibrium strategy $\{P, \gamma\}$ that does not satisfy condition (c). Let j be the first ISP in topological sort order for whom $y_i < y_j$ and there exists a flow on edge (u, v). Now consider a deviation for ISP j where it increases price of all incoming edges to y_j if they were lower earlier. After deviation, no matter if the flow on edge (u, v) increases or decreases, ISP j will be better off since it was losing money on the flow before. Since the price of (u, v) increased while the prices of all other edges remained constant, then due to our assumption about tie-breaking rules, we know that no outgoing edge, other than (u, v), will experience a decrease in flow (although they may experience an increase in flow). We need to examine how this affects the utility of ISP j. W.l.o.g., consider a path $p = \{u, w_1, w_2, \ldots, w_k, v\}$ which experiences an increase in flow after deviation. Note that after deviation, all incoming prices of ISP j are $\geq y_j$. Combined with the fact that condition (b) implies $y_j \geq \lambda_j$, it means, even in the worst case, j will not lose money for receiving flow from path p. Thus the deviation will be profitable for ISP j. And since such deviations can not exist in a Nash equilibrium solution, all uniform Nash equilibrium solutions satisfy condition (c).

So far it has been shown that a uniform Nash equilibrium strategy satisfies conditions (a)-(c). We now show an example of a uniform Nash equilibrium which satisfies conditions (a)-(c) but does not satisfy condition (d).

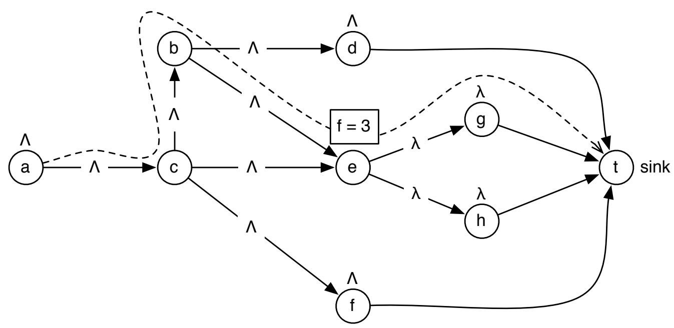


Figure 4.4: Example before deviation.

Consider the example given in figure 4.4. Every node is a separate ISP. Edge (c, e) has capacity 1, edges (e, g), (e, h), (g, t) and (h, t) have capacity 4 while the remaining edges have capacity 3. ISPs a, d and f have source utility Λ and ISPs g and h have source utility λ where $\Lambda \gg \lambda$. Now consider a pricing strategy as given in Figure 4.4. Let us assume that the tie-breaking rules induce a flow of size 3 as shown in the figure, while other ISPs with positive source utilities saturate their outgoing edges. It is easy to verify that this strategy satisfies conditions (a)-(c) but edge (c, e) does not satisfy condition (d). Here $y_c = \Lambda > y_e = \lambda$ even though edge (c, e) is unsaturated. The obvious choice for ISP e is to lower the price on edge (c, e) in order to gain flow on the edge. If ISP e reduces the price of the edge (c, e) by a small amount, then the tie-breaking rule for c may result in a very different flow as shown in Figure 4.5, since how the flow is apportioned among the outgoing edges of c with prices Λ is allowed to depend on whether c is forwarding 3 units of flow on these edges, or only 2 units of flow.

Player e ends up losing the profitable flow from edge (b, e) while receiving smaller amount of flow on edge (c, e). This is clearly not profitable for player e, and so is not an improving deviation. Through examination of all other players, it is not difficult to see that the original strategy is in Nash equilibrium, even though condition (d) is not satisfied. ■

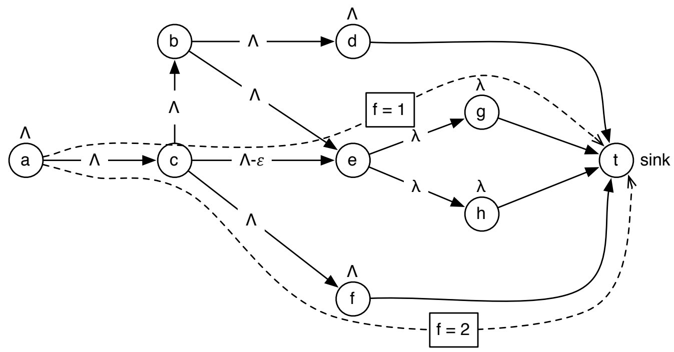


Figure 4.5: Example after deviation.

The above theorem implies that the conditions of Theorem 14 essentially characterize all uniform Nash equilibria for the ISP pricing and forwarding game. Unfortunately, as we showed in the proof of Theorem 14, there exist uniform Nash equilibria that do not obey property (d), and so this property remains sufficient but not necessary for the existence of uniform Nash equilibrium.

4.3.2 Price of Stability

By an optimal solution to this game, we will mean one in which the sum of the utilities of all players is maximized. Since the price paid by players to each other cancels out in the sum, optimal solutions are ones in which $\sum_v \lambda_v f_v$ is maximized. We will call a flow f^* *socially optimal* if $\sum_v \lambda_v f_v^*$ is maximum over all flows that obey capacity constraints and where f_v^* flow originates at node v, with all flow ending at the sink t (it is easy to see that, without loss of generality, all edges incident on t are saturated). Clearly, $\sum_v \lambda_v f_v^*$ is the social welfare in an optimal solution, since if all prices are set to 0, and γ is such that f^* is the resulting flow, then this results in social welfare of $\sum_v \lambda_v f_v^*$. We now prove that flow f^* can also be achieved by a Nash equilibrium solution, i.e., that the price of stability of this game is 1.

Theorem 15 *Given a socially optimal flow f^*, there exists a collective strategy $\{P, \gamma\}$ such that $f(P, \gamma) = f^*$ and $\{P, \gamma\}$ is a uniform Nash Equilibrium. In other words, the price of stability is* 1 *for well-provisioned networks.*

Proof. We will prove this theorem by constructing a strategy $\{P, \gamma\}$ such that

$f(P, \gamma) = f^*$, and then showing that P and $f(P, \gamma)$ satisfy the conditions of Theorem 13.

We first set the prices P. Let V_S be the set of nodes that send their own flow in f^* (i.e., $f_v^* > 0$), and V_{NS} be the set of nodes that do not send their own flow in f^*. Since our goal is to form prices that satisfy condition (a) of Theorem 13, we will set all prices on outgoing edges in P to be the same for each node: denote these prices as y_v for node v. For every vertex $v \in V_S$, set $y_v = \lambda_v$, thus satisfying condition (b) as well. We now need to assign prices to the vertices of V_{NS}.

Let G^r be the residual graph of f^*, i.e., it contains an edge (v, u) for every edge $e = (u, v)$ with $f_e^* > 0$ (call these *backward* edges), and an edge (u, v) for every edge $e = (u, v)$ with $f_e^* < c_e$ (call these *forward* edges). The following lemma is a simple consequence of the fact that f^* is socially optimal.

Lemma 6 *If $u \in V_S$ is reachable from v in G^r, then $\lambda_u \geq \lambda_v$.*

Proof. To see that $\lambda_u \geq \lambda_v$, consider the path from v to u in G^r. This path is an augmenting path of the flow f^*. Since $u \in V_S$, we know that $f_u^* > 0$. We can thus use a flow path from u to t together with the augmenting path from v to u in order to augment the flow f^* by decreasing the flow f_u^* by $\varepsilon > 0$, and increasing the flow f_v^* by ε, while still satisfying all capacity constraints. If $\lambda_u < \lambda_v$, this would result in a flow with higher social welfare than f^*, giving us a contradiction. Thus, $\lambda_u \geq \lambda_v$. This is the only place where we use the fact that f^* is a socially optimal flow. ■

To assign prices to nodes of V_{NS}, we now proceed to obtain upper and lower bounds on the y_v values of nodes $v \in V_{NS}$.

Lemma 7 *In every uniform Nash equilibrium strategy $\{P^*, \gamma^*\}$ with $f(P^*, \gamma^*) = f^*$ and prices y_v, we have that for all $u \in V_S$ and $v \in V$ with u being reachable from v in G^r, it must be that $\lambda_u \geq y_v$.*

Proof. We can show this inductively on the distance from v to u in G^r. Since this is a uniform Nash equilibrium and $u \in V_S$, then by condition (b) we know that $\lambda_u = y_u$. Now consider a node v such that there is an edge (v, w) in G^r with

u being reachable from w in G^r by a path of length at most k. By the inductive hypothesis, $\lambda_u \geq y_w$. If (v, w) is a forward edge, then by condition (d), we know that $y_v \leq y_w \leq \lambda_u$, as desired. If instead (w, v) is a backward edge, then by condition (c), we similarly know that $y_v \leq y_w \leq \lambda_u$. ■

Let R_v^h be the set of nodes $u \in V_S$ such that u is reachable from v in G^r. Then, using Lemma 7 we know that $h_v = \min_{u \in R_v^h} \lambda_u$ is an upper bound for y_v in any uniform Nash equilibrium that results in flow f^*.

Lemma 8 *In every uniform Nash equilibrium strategy $\{P^*, \gamma^*\}$ with $f(P^*, \gamma^*) = f^*$ and prices y_v, we have that for all $u \in V_S$ and $v \in V$ with v being reachable from u in G^r, it must be that $\lambda_u \leq y_v$.*

Proof. The proof is the same as the proof of Lemma 7, with conditions (c) and (d) reversed. ■

Let R_v^l be the set of nodes $u \in V_S$ such that v is reachable from u in G^r. Then, using Lemma 8 we know that $l_v = \max_{u \in R_v^l} \lambda_u$ is a lower bound for y_v in any uniform Nash equilibrium that results in flow f^*.

We now set the y_v values for $v \in V_{NS}$ in the following manner: For every vertex v, $y_v = h_v$.

Now that we have fully defined the prices P, we can choose flow generation functions $\gamma_v \in \Gamma_v$ for all v such that $f(P, \gamma) = f^*$. To do this, simply set γ_v to be such that $\gamma_v(f_v^{in*}, \vec{y_v}) = f_v^{out*}$, where f_v^{in*} and f_v^{out*} are the flows into and out of v in flow f^*, and $\vec{y_v}$ is a vector of prices which equals y_v on every edge of E_v^{out}. To show that this is a valid flow generation function, we just need to prove that $f_v^{out*} \in \mathcal{F}_v(f_v^{in*}, \vec{y_v})$, so we check all the necessary conditions listed in Section 4.1. Clearly, the capacity and conservation conditions are satisfied since f^* satisfies them. All edges have the same outgoing price, so v's utility remains the same no matter which edges its flow is forwarded on. Finally, notice that for all v, we have that $y_v \geq \lambda_v$, since if $v \in V_S$ then $y_v = \lambda_v$, and if $v \in V_{NS}$, then this also holds due to Lemma 6, which implies that $h_v \geq \lambda_v$. Thus, with these prices it is never strictly profitable for a node v to send its own flow, and $p_e = y_v$ can only be strictly greater than λ_v when $v \in V_{NS}$, i.e., when $f_v^* = 0$, as desired. Thus, we have fully defined a strategy $\{P, \gamma\}$ such that $f(P, \gamma) = f^*$. All that is left to prove is that $\{P, \gamma\}$ is a uniform Nash equilibrium:

Condition (a). We know that condition (a) is satisfied since all edges of E_v^{out} were given the same price y_v.

Condition (b). Condition (b) is satisfied by the prices generated by our algorithm, since we set prices of all $v \in V_S$ to λ_v, and by Lemma 6, $y_v = h_v \geq \lambda_v$ for all $v \in V_{NS}$.

Condition (c). Consider an edge $e = (u, v)$ with $f_e^* > 0$. Then, edge (v, u) is a backward edge in G^r. If $u \in V_S$ and $v \in V_{NS}$, then Lemma 7 immediately implies that $y_u = \lambda_u \geq h_v = y_v$.

Since G^r has an edge (v, u), then all nodes reachable from u in G^r are also reachable from v. Thus by Lemma 7, if $u, v \in V_{NS}$, then $h_u \geq h_v$, and so $y_u \geq y_v$, as desired.

If $u \in V_{NS}$ and $v \in V_S$, then let w be the node of V_S reachable from u in G^r such that $h_u = \lambda_w$. Since w is reachable from u, then w is also reachable from v in G^r. Therefore, by Lemma 6, we know that $y_u = \lambda_w \geq \lambda_v = y_v$.

Finally, if both u and v are in V_S, then Lemma 6 immediately implies that $y_u = \lambda_u \geq \lambda_v = y_v$, as desired. Hence condition (c) is satisfied.

Condition (d). Consider an edge $e = (u, v)$ with $f_e^* < c_e$. Then, edge (u, v) is a forward edge in G^r. If $v \in V_S$ and $u \in V_{NS}$, then Lemma 7 immediately implies that $y_u = h_u \leq \lambda_v = y_v$.

Since G^r has an edge (u, v), then all nodes reachable from v in G^r are also reachable from u. Thus by Lemma 7, if $u, v \in V_{NS}$, then $h_u \leq h_v$, and so $y_u \leq y_v$, as desired.

If $v \in V_{NS}$ and $u \in V_S$, then let w be the node of V_S reachable from v in G^r such that $h_v = \lambda_w$. Since w is reachable from v, then w is also reachable from u in G^r. Therefore, by Lemma 6, we know that $y_v = \lambda_w \geq \lambda_u = y_u$.

Finally, if both u and v are in V_S, then Lemma 6 immediately implies that $y_u = \lambda_u \leq \lambda_v = y_v$, as desired. Hence condition (d) is satisfied.

Hence we have shown that a collective strategy $\{P, \gamma\}$ can be computed in polynomial time such that it is a uniform Nash equilibrium and $f(P, \gamma) = f^*$. ■

4.3.3 Reasonable Assumptions and Price of Anarchy

Consider a game where all players with non-zero λ_v value are not neighbors of the sink. Now consider a strategy for this game where every vertex charges a very high price (say bigger than the highest λ_v value). Given this pricing strategy, no vertex will send its own flow and still every vertex will have no incentive to deviate, i.e., the strategy will be in Nash equilibrium. This is because no vertex would unilaterally reduce the prices of its incoming edges, given that they will have to pay a large amount to forward any flow sent to them. Nodes that have edges incident to the sink will not change their prices as there is no hope of obtaining any flow and hence, any profit. In this Nash equilibrium strategy the total utility of players is 0 and hence the price of anarchy is unbounded. These "bad equilibria" cannot be eliminated even after introducing pairwise deviations.

In order to eliminate such unrealistic solutions from consideration, work dealing with similar scenarios made some reasonable assumptions about player behavior. For example, [3] assumes that if a player does not receive any flow on its incoming edge, then she never charge an unnecessarily large price for this edge. In this section, we make the same assumption on the players' pricing strategy:

Property 1 *If a vertex v does not receive any flow on edge (u, v), then it sets $p_{(u,v)}$ to be the price of the cheapest unsaturated outgoing edge of v, if one exists.*

We call pricing strategies that satisfy this property *reasonable*. This property simply says that given an edge (u, v) that has no flow on it, node v will charge the minimum price such that potential flow on this edge will not result in loss of utility for v. Below we show that, at least for single-source games (i.e., games where only one node has a non-zero λ value), this additional assumption on player behavior causes all equilibria to become as good as the optimum solution.

Theorem 16 *For a single source game where players form reasonable pricing strategies, the price of anarchy is* 1.

Proof. Let the source node be s. Any node that is not reachable from s can never receive a flow and hence the strategy of such a node will not have consequence on the outcome of the game. Hence we assume w.l.o.g. that no such node exists. Note that if there exists an edge from the source to sink then it will always be saturated. Hence we also assume w.l.o.g. that no such edge exists.

Due to the non-monopolistic property of the network, the capacity of the minimum cut, and hence the size of the largest $s - t$ flow, will be the sum of the capacities of the outgoing edges of s. So in the socially optimal solution, all outgoing edges of s are saturated.

In order to prove the theorem we make use of the following notation. Let N_s be the set of neighboring vertices of the source s. If c_e is the capacity of the edge e then the size of the socially optimal flow is $\sum_{v \in N_s} c_{(s,v)}$. Let N_t be the set of nodes adjacent to the sink t.

Lemma 9 *In any Nash equilibrium for a single source game where players form reasonable pricing strategies, for all $(u, v) \in E$ such that $u \neq s$, $p_{(u,v)} = 0$.*

Proof. We prove this lemma by induction. Consider a topological sort order of the network, with the first element in the order being node s followed by the nodes belonging to N_s. The last element will be the sink node t whereas the penultimate element will be a node v such that $v \in N_t$. According to the topological ordering, all outgoing edges of v will be incident to t and hence all outgoing edges of v will have price $p_e = 0$ in any Nash equilibrium strategy, say $\{P, \gamma\}$. Let this be the base case for induction.

Now consider an arbitrary node v_i in the ordering $s, v_1, v_2, \ldots, v_k, t$. We assume that the prices of all outgoing edges of $v_{i+1}, v_{i+2}, \ldots, v_k$ in $\{P, \gamma\}$ are 0. Suppose to the contrary that in strategy $\{P, \gamma\}$, the price of edge (v_i, v_j), where $(i < j \leq k)$, is non-zero, and assume wlog that (v_i, v_j) is such an edge with largest price. Then consider the following two cases:

Case 1. Under strategy $\{P, \gamma\}$, there does not exist any flow on the edge (v_i, v_j). In this case, Property 1 tells us that node v_j would set the price of (v_i, v_j) to 0. This contradicts our assumption that (v_i, v_j) has non-zero price.

Case 2. Under strategy $\{P, \gamma\}$, there does exist a non-zero flow on edge $e = (v_i, v_j)$. This implies that node v_i is not sending its own flow, by the last property of the flow generation function (i.e., it would be unprofitable for v_i to send its own flow since its λ value equals 0, and it sends positive flow on edge e with price greater than 0). From the non-monopolistic property of the network, and since v_i is not sending its own flow, we know that there exists another node $v_{j'}$ such that the edge $e' = (v_i, v_{j'})$ is not saturated. Also from the inductive hypothesis we know that the price of all outgoing edges of $v_{j'}$ is 0. Let $f = f(P, \gamma)$.

In this case, consider a deviation by node $v_{j'}$ in which it changes the price on edge e' to $p_e - \epsilon$ while keeping the rest of its strategy the same, where $p_e > 0$ is the price of edge e, and

$$\epsilon < \frac{p_e \min\{f_e, c_{e'} - f_{e'}\}}{c_{e'}}.$$

Call this new collective strategy $\{P', \gamma\}$. We claim that the utility of $v_{j'}$ strictly increases after this deviation. Consider the flow generation of $f(P', \gamma)$. All outgoing prices and incoming flow vectors are the same as in $\{P, \gamma\}$ until node v_i, so the incoming flow into node v_i is the same in both $f(P, \gamma)$ and $f(P', \gamma)$.

First consider the change in flow on e'. By setting its price to be smaller than p_e, $v_{j'}$ has guaranteed that it receives at least $\min\{f_e, c_{e'} - f_{e'}\}$ extra flow on edge e', since flow is sent on edges with smallest price first. Thus the utility of $v_{j'}$ due to edge e' changes from at most $p_e \cdot f_{e'}$ (we assumed that p_e is the largest price of all edges leaving v_i), to at least $(p_e - \epsilon)(f_{e'} + \min\{f_e, c_{e'} - f_{e'}\})$. The utility due to this edge strictly increases, since $\epsilon(f_{e'} + \min\{f_e, c_{e'} - f_{e'}\}) \leq \epsilon c_{e'} < p_e \min\{f_e, c_{e'} - f_{e'}\}$.

Now consider the utility of $v_{j'}$ due to other incoming edges. All edges from nodes earlier than v_i in the topological ordering are sending the same amount of flow to $v_{j'}$ as before the deviation. All edges from nodes after v_i in the topological ordering have price 0, and so flow on them does not change the utility of node $v_{j'}$. Combined with the fact that all outgoing edges from node $v_{j'}$ have price 0, we know that the utility of $v_{j'}$ strictly increases after the deviation. This contradicts our assumption that strategy $\{P, \gamma\}$ is a Nash equilibrium.

Hence by induction we have shown that there does not exist any Nash equilibrium strategy where $p_{(u,v)} > 0$ for any $u \neq s$. This proves the lemma. ■

Using the result of Lemma 9, we will now show that in any Nash equilibrium strategy, the outgoing edges of s will be saturated. This will imply that every Nash equilibrium is socially optimum, since for a single source s, the social welfare of a solution with flow f is simply $\lambda_s \cdot f_s$.

Suppose to the contrary that in some Nash equilibrium strategy, the edge (s, v) is not saturated. In this case node v can price edge (s, v) just below λ_s so that (s, v) is saturated by the flow algorithm. This will always be a beneficial deviation for v since Lemma 9 tells us that the price on all outgoing edges of v is 0. This is clearly a contradiction and hence no such edge exists. ■

The above proof implies that prices at Nash equilibrium are unique. It also fairly easy to see that a strategy (P, γ) where all neighbors v of s price the edges (s, v) at λ_s is a Nash equilibrium satisfying Property 1, and it results in the socially optimum flow given the appropriate choice of functions γ.

4.4 Well-Provisioned Networks: Non-Linear Utility and Price Functions

In previous sections we analyzed the case where the utility of sending one unit of own flow (will also be referred to as 'per packet') was constant for the player. Note that we only consider well-provisioned networks in this section. Since they are a special case of general networks, all positive results in this section apply to well-provisioned networks, whereas the negative results extend to general networks as well. We will now study the case where the utility is a concave function of the amount of flow sent. This mirrors the fact that sending more flow usually has diminishing returns for the player. We denote this utility function as $\Lambda_v(f_v)$ where f_v is the total amount of own flow sent by node v and Λ_v is continuously differentiable, concave, and non-decreasing. Additionally, denote the derivative of Λ_v by λ_v: in the old model this was a constant, but now it is a non-increasing function.

Similarly when players receive flow on an incoming edge, the processing cost for each unit of flow generally increases with the total amount of flow. Hence we look at the case where the price charged by each player for incoming flow is a convex function. We denote this by $\Pi_e(f_e)$ where f_e is the flow on edge e and Π_e is a continuously differentiable convex non-decreasing function. Let π_e be the derivative of Π_e: in the old model this was called p_e and was a constant; now it is a non-decreasing function.

This more general model is formally defined as follows. Note that our assumptions of the graph being acyclic and non-monopolistic still hold.

Strategy: Once the price functions Π_e for every outgoing edge E_v^{out} and quantity of incoming flow f_v^{in} are set, player v is assumed to forward the flow and send its own flow in order to maximize its utility. In the earlier model where price per packet π_e was always constant, a node would start with sending flow on an edge with the cheapest price per packet and saturate it before moving on to other edges. In this model however, the price per packet on an edge may increase with the amount of flow present on the edge. Hence, if we ignore capacities, a node will always send flow on outgoing edges such that $\pi_e(f_e)$ is equal on all outgoing edges that have flow on them. It is easy to show that such a choice will minimize the node's cost.

Similarly, a node will only send its own flow if the utility of sending a packet is at least as much as the cost of sending it, otherwise sending own flow would result in lowering the player's utility. Hence a player would send its own flow until the utility of the marginal packet $\lambda_v(f_v)$ is equal to its price.

To state the above observations formally, consider a node v that has incoming edges E_v^{in} and outgoing edges E_u^{out}. Given the set of price functions Π_v^{out} and incoming flows f_v^{in}, we define the set of valid resulting flows f_v^{out} to be $\mathcal{F}_v(f_v^{in}, \Pi_v^{out})$, which are all flows satisfying the following conditions:

1. $\forall e \in E_v^{out} : f_e \leq c_e$ and $\sum_{e \in E_v^{in}} f_e + f_v = \sum_{e \in E_v^{out}} f_e$.

2. $\forall e \in E_v^{out}$, if $f_e > 0$ then $\forall e' \in E_v^{out} \backslash e : \pi_e(f_e) \leq \pi_{e'}(f_{e'})$ or $f_{e'} = c_{e'}$.

3. $\forall e \in E_v^{out} : \pi_e(f_e) < \lambda_v(f_v)$ implies that e is saturated, and if $f_v > 0$ then $\forall e \in E_v^{out}$, $f_e > 0 : \pi_e(f_e) \leq \lambda_v(f_v)$.

The first condition just says that the generated flow should obey flow conservation and edge capacities. The second condition says that flow is first sent to the edges with smallest marginal price. Finally, the third condition says that traffic is sent by node v if the marginal utility of sending a packet is less than the cost of sending it, and is not sent by v if the marginal utility of sending a packet is more than the cost of sending it. It is easy to show (through first order conditions for maximizing utility) that satisfying the above conditions maximizes utility for players. Note that these subsume the conditions in Section 4.1, replacing p_e with $\pi_e(f_e)$ and λ_v with $\lambda_v(f_v)$: hence this model is a strict generalization.

As in the previous model, each of these valid flows correspond to a tie-breaking rule selected by player v. We associate these tie-breaking rules with a flow generation function γ_v, which, given the incoming flows and outgoing prices, produces an outgoing flow. The set of these flow generation functions is denoted by Γ_v, defined in exactly the same manner as in Section 4.1. Hence the strategy of a player v is given by the tuple $\{\Pi_v^{in}, \gamma_v\}$. We denote the collective strategy of all players by $\{\Pi, \gamma\}$.

Outcome and Utility: Also like the previous model, the flow outcome can be easily determined if the nodes are considered in the topological sort order and is denoted by $f(\Pi, \gamma)$. Given the resulting flow $f(\Pi, \gamma)$, utility of player v is given by the following expression:

$$utility_v(\Pi, \gamma) = \sum_{e \in E_v^{in}} \Pi_e(f_e) - \sum_{e \in E_v^{out}} \Pi_e(f_e) + \Lambda_v(f_v).$$

4.4.1 Non-linear Prices

In this section we assume that utility functions Λ_v are linear, and show that all our results for linear price functions also hold for arbitrary convex price functions, thus showing that allowing players to set non-linear prices does not make the system any worse. To do this, we prove analogues of Theorems 13 and 16. By proving that the conditions from Theorem 13 imply that a solution is a Nash equilibrium, even when changing your strategy to an arbitrary price function is allowed, we immediately get the consequence that the price of stability is 1, since we already showed how to create an optimal solution satisfying these conditions in Theorem 15.

Theorem 17 *For instances with linear utility functions and non-decreasing, convex price functions: if flow $f(\Pi, \gamma)$ and pricing strategy Π satisfy the following conditions, then strategy $\{\Pi, \gamma\}$ is a Nash equilibrium:*

(a) For every node u, the price function on all edges of E_u^{out}, except edge (u,t) if it exists, is identical and linear, i.e., $\pi_{(u,v)}(x) = p_e$ for all x and constant p_e. Let the constant price per packet p_e be denoted by y_u.

(b) If $f_u > 0$ then $y_u = \lambda_u$; if $f_u = 0$ then $y_u \geq \lambda_u$.

(c) For $\forall v \neq t$, if edge (u,v) has a positive flow on it, then $y_u \geq y_v$.

(d) For $\forall v \neq t$, if edge $e = (u,v)$ is unsaturated ($f_e < c_e$), then $y_u \leq y_v$.

Proof. Let $\{\Pi, \gamma\}$ satisfy the conditions of Theorem 17, and suppose to the contrary that for some player v, there exists a strictly improving unilateral deviation from $\{\Pi, \gamma\}$, where v changes its strategy from (Π_v^{in}, γ_v) to $(\Pi_v^{in'}, \gamma_v')$. Denote by $\{\Pi', \gamma'\}$ the collective player strategy after v's deviation, and let $f = f(\Pi, \gamma)$, $f' = f(\Pi', \gamma')$.

For any edge $e = (u, v) \in E_v^{in}$ which has constant price per packet p_e before deviation, let Π_e' be the price function after deviation. We define the following terms:

- Let f_e^l be the flow of f_e' that "costs at most p_e". Formally, let $L = \{x | \pi_e'(x) \leq p_e\}$ be the set of flow amounts that cost less than p_e per packet, and let $\overline{L}$ be the supremum of L. Then, $f_e^l = \min(\overline{L}, f_e')$.

- Let $f_e^h = max(0, f'_e - f_e^l)$ be the rest of the flow of f'_e.

Note that $f_e^l + f_e^h = f'_e$. Since for all e, all functions $\pi_e(x) = p_e$ are constant by condition (a), then just as in the proof of Theorem 13, the utility of node v in strategy $\{\Pi, \gamma\}$ can be stated as follows:

$$utility_v(\Pi, \gamma) = \sum_{e \in E_v^{in}} f_e \cdot (p_e - y_v) + f_v \cdot (\lambda_v - y_v) + c_{(v,t)} \times y_v. \tag{4.3}$$

Notice that each term $f_e(p_e - y_v)$ is nonnegative due to condition (c), and the term $f_v(\lambda_v - y_v)$ equals zero due to condition (b).

Similarly, the utility of v in strategy $\{\Pi', \gamma'\}$ is the same as in Equation 4.3 except for the fact that the edge prices can now be arbitrary non-decreasing convex functions. It can be stated as follows:

$$utility_v(\Pi', \gamma') = \sum_{e \in E_v^{in}} [\Pi'_e(f'_e) - f'_e \cdot y_v] + f'_v \cdot (\lambda_v - y_v) + c_{(v,t)} \times y_v. \tag{4.4}$$

Note that, since we assumed that every edge (v, t) is saturated (since node v can always forward or send its own flow to fill up edge (v, t) to capacity without decreasing its utility), then the term $c_{(v,t)} \cdot y_v$ is present both before before and after the deviation.

We now analyze the effect of price function changes from $\Pi_e(x) = p_e \cdot x$ to $\Pi'_e(\cdot)$ on the sum in Equations 4.3 and 4.4 for each edge in E_v^{in}. For any edge $e \in E_v^{in}$, the first two terms of Equation 4.4 can be represented as follows:

$$\Pi'_e(f'_e) - f'_e \cdot y_v = \int_0^{f_e^l} \pi'_e(x)dx - f_e^l \cdot y_v + \int_{f_e^l}^{f'_e} \pi'_e(x)dx - f_e^h \times y_v. \tag{4.5}$$

We first claim that for $\forall e \in E_v^{in}$, $f_e^h = 0$. Suppose to the contrary that there exists an edge $e = (u, v) \in E_v^{in}$ such that $f_e^h > 0$. Condition (b) of Theorem 17 says that $\lambda_u \le p_e < \pi'_e(f'_e)$. Then from condition (3) of definition of flow generating function we know that node u does not send its own flow in f', i.e., node u only forwards incoming flow in f'. Since $f'_u = 0$, the non-monopolistic property for edge e requires that either $f'_e = 0$ or $\exists e' \in E_u^{out} \backslash e$ such that e' is not saturated in f'. Also, condition (a) of the theorem says that $\forall e' \in E_u^{out} \backslash e$, $\pi'_{e'} = p_e$. Now, since we have assumed $f_e^h > 0$, it has to be that $\exists e' \in E_u^{out} \backslash e$ such that $\pi'_{e'}(f'_{e'}) = p_e < \pi'_e(f'_e)$ where edge e' is not saturated. But this violates condition (2) of definition of the flow generating function.

Hence $f_e^h = 0$ and the term $\int_{f_e^l}^{f'_e} \pi'_e(x)dx - f_e^h \cdot y_v = 0$.

In order to analyze the term $\int_0^{f_e^l} \pi'_e(x)dx - f_e^l \cdot y_v$, we look at two cases.

- e is saturated in f_e: In this case $f_e^l \leq f_e = c_e$. Also, $\pi'_e(x) \leq p_e$ for $0 \leq x \leq f_e^l$ (by definition of f_e^l). Combined with the fact that $f_e \cdot (p_e - y_v) \geq 0$, it implies that $\int_0^{f_e^l} \pi'_e(x)dx - f_e^l \cdot y_v \leq f_e^l(p_e - y_v) \leq f_e \cdot (p_e - y_v)$.

- e is unsaturated in f_e: In this case condition (d) of the theorem says that $\pi'_e(x) \leq p_e \leq y_v$ for $0 \leq x \leq f_e^l$. Since the term $f_e(p_e - y_v)$ is nonnegative, this implies that $0 = f_e(p_e - y_v) \geq \int_0^{f_e^l} \pi'_e(x)dx - f_e^l \cdot y_v$.

Hence for every edge $e \in E_v^{in}$, $f_e \cdot (p_e - y_v) \geq \Pi'_e(f'_e) - f'_e \cdot y_v$.

Also we know because of condition (b) that $f_v \cdot (\lambda_v - y_v) = 0$. The only way that $f'_v \cdot (\lambda_v - y_v) > 0$ is if $\lambda_v - y_v > 0$, which contradicts condition (b). Thus, this term equals 0 both before and after v's deviation.

Hence the change in node v's strategy can only result in a drop in utility of node v. Thus the collective strategy $\{\Pi, \gamma\}$ is a Nash equilibrium. ■

In section 4.3.3 we showed that when prices have to be linear, the price of anarchy is 1 for networks with a single source under the mild assumption that players do not set large prices without a good reason (Property 1). In this section, we show that the same result holds if prices are allowed to be convex non-decreasing functions. Note that utilities are still linear. Since edge prices are allowed to be functions, we call pricing strategies that satisfy the following property as reasonable:

Property 2 *If a vertex v does not receive any flow on edge (u, v), then it sets $\Pi_{(u,v)}(x) = p_{(u,v)}x$ where $p_{(u,v)}$ is the cheapest marginal price of all unsaturated outgoing edges of v, if one exists.*

We now prove the theorem.

Theorem 18 *If node utilities are linear and edge prices are allowed to be convex non-decreasing functions then, for a single source game where players form reasonable pricing strategies, the price of anarchy is* 1.

Proof. Let the source node be s. Any node that is not reachable from s can never receive a flow and hence the strategy of such a node will not have consequence on the outcome of the game. Hence we assume w.l.o.g. that no such node exists. Note that if there exists an edge from the source to sink then it will always be saturated. Hence we also assume w.l.o.g. that no such edge exists.

Due to the non-monopolistic property of the network, the capacity of the minimum cut, and hence the size of the largest $s-t$ flow, will be the sum of the capacities of the outgoing edges of s. So in the socially optimal solution, all outgoing edges of s are saturated.

In order to prove the theorem we make use of the following notation. Let N_s be the set of neighboring vertices of the source s. If c_e is the capacity of the edge e then size of the socially optimal solution will be given by $\sum_{v \in N_s} c_{(s,v)}$. Let N_t be the set of nodes adjacent to the sink t.

Lemma 10 *If node utilities are linear and edge prices are allowed to be convex non-decreasing functions then, in any Nash equilibrium for a single source game where players form reasonable pricing strategies, for all* $(u, v) \in E$ *such that* $u \neq s$, $\Pi_{(u,v)} = 0$.

Proof. We prove this lemma by induction. Consider a topological sort order of the network, with the first element in the order being node s followed by the nodes belonging to N_s. The last element will be the sink node t whereas the penultimate element will be a node v such that $v \in N_t$. According to the topological ordering, all outgoing edges of v will be incident to t and hence all outgoing edges of v will have price $\Pi_e = 0$ in any Nash equilibrium strategy, say $\{\Pi, \gamma\}$. Let this be the base case for induction.

Now consider an arbitrary node v_i in the topological ordering $s, v_1, v_2, \ldots, v_k, t$. We assume that the prices of all outgoing edges of $v_{i+1}, v_{i+2}, \ldots, v_k$ in $\{\Pi, \gamma\}$ are 0. We will first show that all outgoing edges of v_i satisfy the following property:

Property 3 *If f_e is the flow on edge e then $\Pi_e(x) = p_e x$ for $0 \leq x \leq f_e$ for some constant $p_e \geq 0$.*

Suppose to the contrary that in strategy $\{\Pi, \gamma\}$, the price of edge (v_i, v_j), where $(i < j \leq k)$, does not satisfy Property 3. Let edge (v_i, v_j) have a concave non-decreasing price function $\Pi_e(x)$, and a flow of size f_e (possibly zero).

Then consider the following two cases:

Case 1. Under strategy $\{\Pi, \gamma\}$, there does not exist any flow on the edge (v_i, v_j), i.e., $f_e = 0$. In this case, Property 2 tells us that node v_j would set the price of (v_i, v_j) to 0. This contradicts our assumption that (v_i, v_j) does not satisfy Property 3.

Case 2. Under strategy $\{\Pi, \gamma\}$, there does exist a non-zero flow on edge $e = (v_i, v_j)$, i.e., $f_e > 0$. Now consider a deviation to a new price function for edge e: $\Pi'_e(x) = p'_e x$ for $0 \leq x \leq f_e$ and $\Pi'_e(x) = \Pi_e(x)$ for $x > f_e$ such that $p'_e < \pi_e(f_e)$ and $f_e \cdot p'_e > \int_0^{f_e} \pi_e(x)dx$. Note that such a constant p'_e exists since $\pi_e(x)$ is a monotonic non-decreasing function and $\pi_e(x) < \pi_e(f_e)$ for some $x \leq f_e$.

Call this new collective strategy $\{\Pi', \gamma\}$. We claim that the utility of v_j strictly increases after this deviation. Consider the flow generation of $f(\Pi', \gamma)$. All outgoing prices and incoming flow vectors are the same as in $\{\Pi, \gamma\}$ until node v_i, so the incoming flow into node v_i is the same in both $f(\Pi, \gamma)$ and $f(\Pi', \gamma)$.

First consider the change in the flow on e. We know that node v_i is only forwarding flow and not sending any of its own (since it is sending flow on an edge with non-zero price). Also, since edge e received a flow of size f_e in $f(\Pi, \gamma)$, condition (2.) of definition of flow generating function says that for all $e' \in E^{out}_{v_i} \backslash e$, either $\pi_{e'}(f_{e'}) \geq \pi_e(f_e) > p'_e$ or e' was saturated. Hence e would receive a flow of size at least f_e in $f(\Pi', \gamma)$. For node v_j, the utility due to this edge strictly increases, since $f_e \cdot p'_e > \int_0^{f_e} \pi_e(x)dx$.

Now consider the utility of v_j due to other incoming edges. All edges from nodes earlier than v_i in the topological ordering are sending the same amount of flow to v_j is before the deviation. All edges from nodes after v_i in the topological ordering have outgoing price 0, and so flow on them does not change the utility of node v_j. Combined with the fact that all outgoing edges from node v_j have price 0 and that v_j is not sending its own flow, we know that the utility of v_j strictly increases after the deviation. This contradicts our assumption the strategy $\{\Pi, \gamma\}$ is a Nash equilibrium. Therefore in every Nash equilibrium strategy, (v_i, v_j) must satisfy Property 3.

Hence all outgoing edges of v_i will satisfy Property 3. We will use this observation to show that all outgoing edges of v_i in fact have price functions $\Pi_e = 0$.

Suppose to the contrary that in strategy $\{\Pi, \gamma\}$, the price of edge (v_i, v_j), where $(i < j \leq k)$, is non-zero. We have already shown that this edge will satisfy Property 3. Assume w.l.o.g. that (v_i, v_j) is such an edge with the largest p_e as defined by Property 3. The argument that this is not possible is essentially the same as in Lemma 9. Hence by induction we have shown that there does not exist any Nash equilibrium strategy where $\Pi_{(u,v)} \neq 0$ for any $u \neq s$. This proves the lemma. ∎

Using the result of Lemma 10, we will now show that in any Nash equilibrium strategy, the outgoing edges of s will be saturated. This will imply that every Nash equilibrium is socially optimum, since for a single source s, the social welfare of a solution with flow f is simply $\lambda_s \cdot f_s$.

Suppose to the contrary that in some Nash equilibrium strategy, the edge (s, v) is not saturated. In this case node v can price edge (s, v) just below λ_s so that (s, v) is saturated by the flow algorithm. This will always be a beneficial deviation for v since Lemma 10 tells us that the price on all outgoing edges of v is 0. This is clearly a contradiction and hence no such edge exists. ∎

4.4.2 Non-linear Utilities

In this section we show that, if utility functions Λ_v can be non-linear, then there may not exist a pure Nash equilibrium.

Theorem 19 *If the player utilities Λ_v are concave non-decreasing functions, then pure Nash equilibrium may not exist.*

Proof. Consider the example in Figure 4.6.

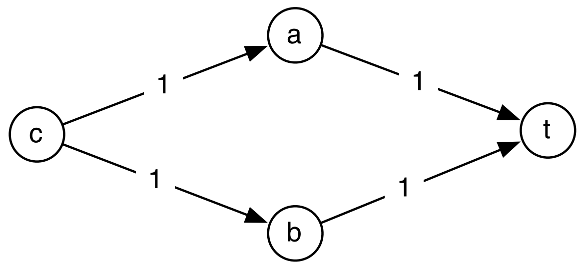


Figure 4.6: $\mathbf{\Lambda_c(f) = -9 \times f^2 + 37 \times f}$ for $\mathbf{f \leq 2}$. $\mathbf{\Lambda_a = \Lambda_b = 0}$. All edges have capacity 1.

Lemma 11 *For any Nash equilibrium strategy for example in Figure 4.6, there exists another equilibrium strategy where the price functions for edges (c, a) and (c, b) are linear.*

Proof. Note that in all Nash Equilibrium solutions, there will always be some flow on edge (c, a). If there is no flow on edge (c, a) then node a's utility if 0, whereas by lowering the price $\Pi_{(c,a)}$ sufficiently, a can ensure that it receives some flow on the edge, which provides positive utility. The same argument symmetrically holds for edge (c, b).

We first show that in Nash equilibrium strategy $\{\Pi, \gamma\}$ for this example, all edges must satisfy Property 3.

If this is not the case then let $\{\Pi, \gamma\}$ be a Nash equilibrium strategy where we assume wlog that edge (c, a) has a concave non-decreasing price function $\Pi_e(x)$, and a positive flow of size f_e and (c, a) does not satisfy Property 3. The utility of node a can be stated as :

$$utility_a = \Pi_e(f_e) = \int_0^{f_e} \pi_e(x)dx. \tag{4.6}$$

Now consider a deviation to a new price function for edge (c, a): $\Pi'_e(x) = p'_e x$ for $0 \leq x \leq f_e$ and $\Pi'_e(x) = \Pi_e(x)$ for $x > f_e$ such that $p'_e < \pi_e(f_e)$ and $f_e \cdot p'_e >$

$\int_0^{f_e} \pi_e(x)dx$. Note that such a constant p'_e exists since $\pi_e(x)$ is a monotonic non-decreasing function and $\pi_e(x) < \pi_e(f_e)$ for some $x \leq f_e$ (since (c, a) does not satisfy Property 3).

Note that in $\{\Pi', \gamma\}$, the price function of edge (c, b) is the same whereas the marginal price for edge (c, a) has decreased. By condition (3.) of definition of the flow generating function we know that:

$$\lambda_c(f_c) \geq \pi_e(f_e) > p'_e. \tag{4.7}$$

This means that node c would send a flow of size at least f_c after deviation. If edge (c, b) was saturated in the flow $f(\Pi, \gamma)$ then edge (c, a) would receive a flow a size at least f_e after deviation. If (c, b) was not saturated in flow $f(\Pi, \gamma)$ then from condition (2.) of definition of flow generating function we know that $\pi_{(c,b)}(f_{(c,b)}) \geq \pi_e(f_e) > p'_e$. Therefore, in this case too, edge (c, a) would receive a flow a size at least f_e after deviation.

And since $f_e \cdot p'_e > \int_0^{f_e} \pi_e(x)dx$, utility of node a as given by Equation 4.6, will increase. This contradicts the assumption that strategy $\{\Pi, \gamma\}$ was in Nash equilibrium. Therefore in all Nash equilibrium pricing strategies, edges will satisfy Property 3.

Now consider a new pricing strategy $\bar{\Pi}$ constructed from a Nash equilibrium strategy $\{\Pi, \gamma\}$ as follows: for every edge e with flow f_e on it and $\pi_e(f_e) = p_e$ let $\bar{\Pi}_e(f) = f \cdot p_e$ for $f > 0$. We have already shown that in $f(\Pi, \gamma)$, $\forall e \in E$: for $0 < f \leq f_e$, $\Pi_e(f) = f \cdot p_e$.

Note that the only difference between the pricing strategies Π and $\bar{\Pi}$ is that $\forall e \in E$, $\pi_e(f) \geq p_e$ for $f > f_e$ where as $\bar{\pi}_e(f) = p_e$ for $f > f_e$. Therefore there exists a tie-breaking strategy $\bar{\gamma}$ such that $f(\bar{\Pi}, \bar{\gamma}) = f(\Pi, \gamma)$. Since the utility of players a and b depend only on the flow received from node c, their utility is identical in both strategies.

Now, if there exists a profitable deviation for say node a in strategy $\{\bar{\Pi}, \bar{\gamma}\}$ then the same deviation will be profitable in strategy $\{\Pi, \gamma\}$. For example since the utility of node a depends only on flow from node c, it could reduce price of edge (c, a) to obtain more flow. But this deviation would also be valid in $\{\Pi, \gamma\}$. Since we assumed $\{\Pi, \gamma\}$ to be a Nash equilibrium strategy, no such deviation exists. Hence $\{\bar{\Pi}, \bar{\gamma}\}$ is also a Nash equilibrium strategy, thus proving statement of the Lemma. ■

Since we have shown in Lemma 11 that for every Nash equilibrium strategy, there exists one with linear prices, we need only show that such Nash equilibria do not exist.

Suppose such an equilibrium exists for this example: let p_a be the price per packet of edge (c, a), and p_b be the price per packet of edge (c, b) in this equilibrium: the prices of the other two edges must be 0 since they are edges to the sink.

Note that in all Nash Equilibrium solutions, there will always be some flow on edge (c, a). If there is no flow on edge (c, a) then node a's utility if 0, whereas by lowering the price $p_{(c,a)}$ sufficiently, a can ensure that it receives some flow on the edge, which provides positive utility. The same argument symmetrically holds for edge (c, b).

Lemma 12 *In any Nash equilibrium strategy for the example in Figure 4.6, edges (c, a) and (c, b) will have the same price.*

Proof. Suppose to the contrary that $p_a > p_b$. Then, by increasing its price by $\epsilon < p_a - p_b$, node b will still receive the same amount of flow on edge (c, b) according to the flow generating algorithm, since (c, a) has positive flow on it at price p_a, and so (c, b) will be saturated at both price p_b and $p_b + \epsilon$. This deviation increases the utility of node b, and thus this cannot be a Nash equilibrium. ■

Lemma 13 *In any Nash equilibrium strategy for the example in Figure 4.6, edges (c, a) and (c, b) will be saturated.*

Proof. By Lemma 12 we know that in any Nash equilibrium, $p_a = p_b$. Now consider a Nash equilibrium strategy with resulting flow f, where edge (c, a) is not saturated. We know that $f_{(c,a)} > 0$ and $f_{(c,b)} > 0$. Consider a deviation by node a where it sets its incoming price to $p_a - \epsilon$ for $\epsilon < p_a \min\{1 - f_{(c,a)}, f_{(c,b)}\}$. Now edge (c, a) becomes the preferred edge and hence receives more flow (flow that was being routed to (c, b)). Specifically, edge (c, a) must now receive at least $\min\{1 - f_{(c,a)}, f_{(c,b)}\}$ more flow, and thus node a now receives at least $(p_a - \epsilon) \min\{1, f_{(c,a)} +$

$f_{(c,b)}\}$ utility. This is strictly larger than the utility a received before the deviation, since $p_a f_{(c,a)} + \epsilon \min\{1, f_{(c,a)} + f_{(c,b)}\} < p_a \min\{1, f_{(c,a)} + f_{(c,b)}\}$. This contradicts our assumption that the strategy is in Nash equilibrium, and the same argument symmetrically holds for node b. ■

Now consider the flow amount f_c sent by node c in a Nash equilibrium in Example 4.6. We know that if $p = p_a = p_b$ is the price per packet set on both edges out of c, then node c will send a flow of size at most f_c such that $\lambda_c(f_c) = p$ (since any more flow would be unprofitable). Hence, we can define the inverse function such that $\lambda_c^{-1}(p) = f_c$. Specifically, here $\Lambda_c(f) = -9f^2 + 37f$ and hence $\lambda_c(f) = -18f + 37$ and $\lambda_c^{-1}(p) = (37 - p)/18$. Lemma 13 says that in any Nash equilibrium both edges (c, a) and (c, b) have to be saturated. For that be true, node c has to send 2 units of its own flow. This means that the price per packet on the outgoing edges of c cannot be more than 1, and thus the utility of node a in any Nash equilibrium can be no more than 1.

Now consider a deviation where node a increases its incoming price p_a to 10. In this situation, a will still receive a flow of size $\lambda_c^{-1}(10) - 1 = 0.5$ giving it a utility of $10 * 0.5 = 5$. This means that for all strategies that satisfy the conditions laid down by Lemma 13, there exists an improving deviation for player a. Hence there does not exist any Nash equilibrium solution for this example. ■

4.4.2.1 Discrete Prices

The non-existence of Nash equilibrium in the previous example relied heavily on the fact that the prices could be changed by an infinitesimal amount. We now show that even if the prices were to be discrete, the existence of Nash equilibrium is not guaranteed. Let us assume that players are allowed to pick discrete prices and say the smallest unit is δ (e.g., we can say that players can pick integer prices, and thus $\delta = 1$).

Theorem 20 *If utility functions are non-linear then there exists an instance that does not have a pure Nash equilibrium strategy, even with discrete price structures.*

Proof. Consider the example in Figure 4.6 and let δ be the smallest unit of discrete price that players allowed to pick. For this proof, we will assume that δ is much smaller than 1: we can handle the general case with simple re-scaling of the function Λ_c and the edge capacities. Let $e = (c, a)$ and $e' = (c, b)$. We can still assume that prices are linear, so let p_e be the price per packer on edge e, and $p_{e'}$ be the price per packet on edge e'. We first prove the following lemma.

Lemma 14 *If $f_e > 0$ and $f_{e'} > 0$ in a Nash equilibrium solution, then $|p_e - p_{e'}| \leq \delta$.*

Proof. Suppose to the contrary that $p_e < p_{e'} - \delta$. According to the flow generating algorithm, e will be saturated since a higher-priced edge, e', has flow on it. Now by increasing the price of edge e by δ, node j can receive the same amount of flow but at a higher price thus making it an profitable deviation. This is a contradiction since the strategy was supposed to be in Nash equilibrium. ■

We first claim that there does not exist a Nash equilibrium strategy in which node c emits a flow of size less than or equal to 1. In order to prove this claim, suppose to the contrary that ≤ 1 flow is being emitted. Then, without loss of generality, let edge (c, a) be the edge that receives less flow (can be equal) out of the two. This implies that $p_e \geq p_{e'}$. Recall that the function $\lambda_c^{-1}(p) = (37 - p)/18$ gives the total amount of flow emitted by c when the price of both of its outgoing edges is p. The flow on edge e is at most $\lambda_c^{-1}(p_e)/2$ when $p_e = p_{e'}$. When $p_e = p_{e'} + \delta$ instead, then it is still true that the flow on edge e is at most $\lambda_c^{-1}(p_e)/2$. To see this, notice that this is trivially true if $\lambda_c^{-1}(p_e) \geq 2$, since the capacity of edge e is 1. Otherwise, the flow on edge e is at most $\lambda_c^{-1}(p_e) - 1 < \lambda_c^{-1}(p_e)/2$, since we can assume that $\lambda_c^{-1}(p_e) < 2$. Thus we can give an upper bound on the utility of node a as follows:

$$utility_a \leq \frac{p_e \lambda_c^{-1}(p_e)}{2} = \frac{p_e \cdot (37 - p_e)}{2 \cdot 18}.$$

Given this upper bound for utility of a, it can be shown through simple differentiation that the maximum occurs at $p_e = 18.5$. Note that since we are assuming that $f_c \leq 1$, then $p_e \geq 19$. Therefore, $utility_a \leq \frac{19 \cdot (37-19)}{2 \cdot 18} = 9.5$. But in this case node a can improve its utility by reducing the price of edge (c, a) to 18 and thereby getting completely saturated and obtaining a utility of 18. This is clearly a contradiction since we assumed the strategy to be in Nash equilibrium.

This means that if there exists an equilibrium, then $f_c > 1$, and thus price $p_e < 19$. Consider such an equilibrium strategy. From Lemma 14 we know that the price of the two edges (c, a) and (c, b) can differ by at most δ. Let us suppose first that the prices are the same.

Now, if the amount of flow on edge (c, a) is f_a then the utility of node a is $f_a p_e$. If a deviates by reducing its price by δ, then it will be saturated (since $p_e < 19$, and thus c sends at least 1 unit of flow). Since we are assuming that the current strategy is Nash equilibrium, this deviation should not be profitable. Hence:

$$f_a p_e \geq p_e - \delta.$$

The same holds for node b:

$$f_b p_e \geq p_e - \delta.$$

Adding the two inequalities gives the following:

$$\begin{aligned}
& p_e(f_a + f_b) \geq 2(p_e - \delta) \\
\Longrightarrow\ & p_e \lambda_c^{-1}(p_e) \geq 2p_e - 2\delta \\
\Longrightarrow\ & p_e \frac{(37 - p_e)}{18} \geq 2p_e - 2\delta \\
\Longrightarrow\ & p_e^2 - p_e - 36\delta \leq 0 \\
\Longrightarrow\ & p_e \leq \frac{1 + \sqrt{1 + 144\delta}}{2}.
\end{aligned}$$

Let us assume that δ is sufficiently small so that $p_e \leq 3$. At this price, even if node a receives all the flow it can, its utility will be at most 3. But this means that it can profitably deviate by increasing its price to 10, giving it a flow of size 0.5 and thus utility 5.

This contradicts the assumption that the strategy is in Nash equilibrium. A similar analysis works if we assume that the price of edges (c, a) and (c, b) differ by δ. ■

4.5 Simulation Study: Convergence of Dynamic Pricing Updates

We showed in Section 4.3 that for well-provisioned networks, good Nash equilibria exist and can be computed efficiently. Moreover, we partially characterized the set of possible uniform Nash equilibria using a few simple properties. In this section, we will show that, even if the network is not well-provisioned (and even if monopolies exist), simple price dynamics behave extremely well on average. Specifically, our extensive simulations show that price dynamics in our setting converge quickly, and to solutions that achieve good social welfare. We present results from our simulation study that explores the convergence of a simple, best-response based incremental price update policy, and shows that for a large fraction of randomly generated network topologies, it converges quickly to a solution that yields near-optimal network utility. We first perform the study on well-provisioned ISPs, which allows us to simulate networks of up to 500 ISPs. Subsequently, we study networks where ISPs have general topologies (i.e., they are not necessary well-provisioned); for this case, we study networks with up to 50 ISPs, and each ISP comprising of up to 50 nodes (routers). Note that our analytical model and results in Sections 4.1, 4.2, 4.3 assumed a "non-monopolistic" network topology, which ensures that there is enough competition in the network, and thus no single ISP can charge exorbitantly high prices. In our experimental study, we simulated networks where this assumption is made to hold (by choosing link/edge capacities appropriately), as well as networks where the non-monopolistic assumption does not hold. Our extensive simulation experiments show that the nature of the convergence and performance results do not differ significantly across these two kinds of networks. As expected, however, the convergence and performance results for monopolistic network topologies are slightly worse (on average) than that for non-monopolistic networks with similar parameters. In the following, therefore, we only present the simulation study on networks where the non-monopolistic assumption may not hold, which yields more conservative performance/convergence results, and has wider applicability in practice.

4.5.1 Network Generation, Dynamic Price Update Policy, and Convergence Criteria

For well-provisioned networks, we experimented with two different models of inter-ISP connectivity: (i) *Uniformly-random connectivity*, (ii) *BRITE-generated connectivity*. Since we consider a single, specified destination node/ISP, the forwarding network must constitute a directed, acyclic graph (DAG), which is created in the two cases as follows. For uniformly-random connectivity, the specified number of ISPs is created and numbered uniquely (numbers representing the topological order). Then each ISP is given a random number of outgoing edges, chosen uniformly at random between 2 and 6 (inclusive). The endpoints are randomly chosen from the subset of ISPs further down in the topological order. If the chosen number of outgoing edges is greater than or equal to the number of possible endpoints (i.e., number of ISPs further down in the topological order), then only one edge is created per endpoint (note that this ensures connectivity to the destination ISP). Note that the number of incoming edges of ISPs get determined automatically through the above procedure, and can be greater than 6 (no constraint is placed on them). For BRITE-generated connectivity, we use the BRITE simulator [121] to generate the initial network (with average node degree set to 4), from which the destination-specific directed acyclic network is created. Note that the BRITE simulator uses the Barabasi-Albert model [118] to generate scale-free networks that are more representative of router and ISP connectivity in the Internet. In this case, the generated graph is given direction by first choosing a sink (destination) randomly from the ISPs, and then doing a breadth-first search from the sink, thus dividing the ISPs into layers, where all ISPs in the same layer have the same shortest distance to the sink. The edges inside each layer are directed similarly to the uniformly-random connectivity case: by creating a random ordering of ISPs and directing edges from lower-numbered to higher-numbered ISPs. An edge between layers is directed from the ISP that is farther away from the sink to the ISP that is closer to the sink.

Once the connectivity between ISPs is determined (as described above), each ISP is first given a total outgoing capacity equal to its total incoming capacity plus a random amount chosen uniformly from $[0, 1]$ (so ISPs must be traversed in topological order). This outgoing capacity is then divided randomly between the ISP's outgoing edges. If the ISPs are not well-provisioned, the intra-ISP network topologies are also chosen in a similar manner; we discuss this in Section 4.5.3 when we discuss the simulation results for general-topology ISP networks. Finally, the utility values (per unit flow) λ_i are chosen uniformly at random as an integer on $[0, 30]$.

For any given set of prices, the flows are determined by traversing the network of ISPs in the topological order. Given the incoming flows (which is already determined when the ISP is considered), the ISP decides how much of its own flow to generate and how to direct all the flow (that it receives on its incoming edges, as well as what it generates) by solving a linear program, to maximize its overall utility given current prices and incoming flow.

The incremental price update policy used by the ISPs on its incoming edges (i.e., edges in the network that connect an ISP to its upstream neighbors), works as follows. Given the current set of prices, an ISP determines the estimated flow (calculated the same way as described above) on the edge if (i) the price on this edge was reduced by unit amount, and (ii) if the price on this edge was increased by unit amount. This is further used to determine the expected change in utility in both cases (i) and (ii). The ISP then increases or decreases the edge price by one if it is expected to increase the ISP's utility; the price remains the same if no gain in utility can be made. Two other simple and intuitive methods of changing price were tested, and did not provide any observable benefits; thus we only present the results for the price update policy as described above. The pricing update procedure was run for 300 cycles (iterations), where a cycle means that each node is given the opportunity to alter its prices (the order in which they are given this opportunity is a random permutation which remains consistent between cycles). We say that convergence is attained at time (cycle) t (< 200) if the following criteria holds between t and $\max(300, t + 100)$: (i) the minimum total utility in the network must be at least 0.9 times the maximum total utility, and (ii) the slope of the linear-least-squares-regression line on total utility can be at most 0.00002 times the maximum total utility. (The total utility, and its maximum and minimum, are based on the measured values at the end of each cycle of the simulation.) In other words, if (i) and (ii) does not hold for at least 100 consecutive cycles (of the 300 cycles of simulation time), we assume that the utility/prices have not converged. All average results are calculated by averaging over 200 randomly generated networks (uniformly-random or BRITE-generated), with similar parameters.

4.5.2 Networks of Well-Provisioned ISPs

We now present the simulation results for the case of well-provisioned networks, for both uniformly-random and BRITE-generated connectivity.

4.5.2.1 Frequency of Convergence

We first study how frequently our pricing update policy converged to result in stable total utility/prices/flow values. Figures 4.7 and 4.8 plot the ratio of networks of a given size in which the total utility values converged, to the total number of networks tested (we simulated on 200 different networks for each size, as noted earlier). We observe that the total utility values converged in a majority of the networks, and the convergence frequency is much better (almost 100%) for BRITE-generated topologies. This is particularly significant since BRITE-generated topologies are closely representative of inter-ISP connectivity topologies in the Internet. Recall that the networks simulated do not in general satisfy the non-monopolistic assumption under which the equilibrium existence and efficiency results have been shown (in Section 4.3). Therefore, these simulation results show that despite the monopolistic nature of some (possibly many) ISPs, simple best-response based incremental pricing update policies can result in stable prices and flows. (We will see shortly that the resulting stable flows are "efficient" as well.) Note that our convergence criteria ((i) and (ii), as described above) are quite strict and the simulation time is fairly short (300 cycles of simulation time, out of which the total utility must not vary significantly over a 100-cycle period, for convergence). With less strict convergence criteria and longer simulation period, the frequency of convergence is likely to be higher. Also note that frequency of convergence goes down as the number of ISPs increase; this is clearly observed in the case of uniformly-random connectivity between ISPs (Figure 4.7). This is partly due to the fact that the convergence time increases with increasing number of ISPs, as we intuitively expect (we support this observation below when we show convergence time results). This naturally implies that convergence frequency as measured within a fixed time-limit of 300 cycles is likely to reduce as the number of ISPs become larger.

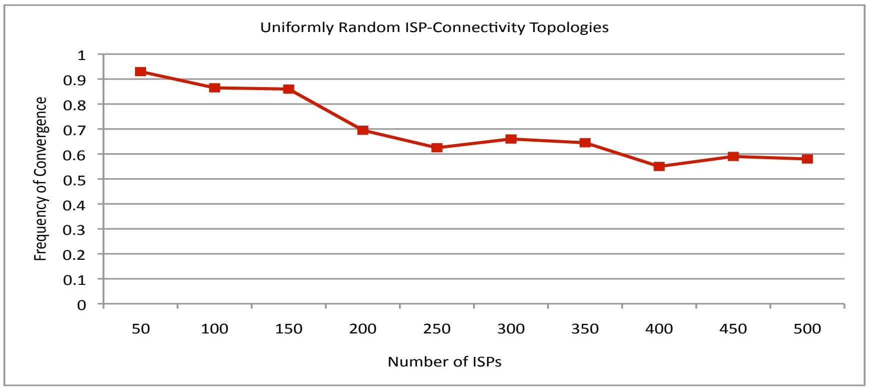


Figure 4.7: Frequency of Convergence (Uniformly-random connectivity).

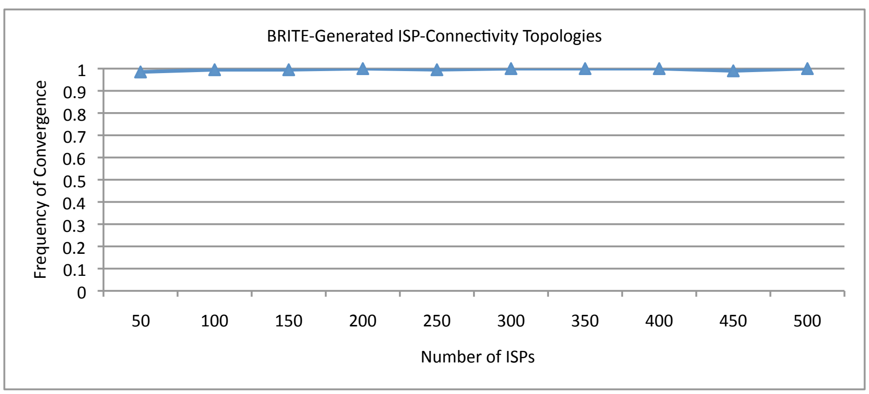


Figure 4.8: Frequency of Convergence (BRITE-generated connectivity).

4.5.2.2 Time of Convergence

Next we study the average convergence time, counting only the networks for which the pricing update policy converged, as discussed above. The results are shown in Figures 4.9 and 4.10. We observe that the average convergence time is quite short: for networks with 500 ISPs, it is about 80 cycles for uniformly-random connectivity, and less than 30 cycles for BRITE-generated connectivity. Note that each step (iteration) of the pricing update procedure requires local information exchange between ISPs (an ISP exchanges price and flow information with neighboring ISPs); so the message exchange and computation required in each cycle can be done quickly if executed in a distributed manner. The short convergence time for BRITE networks is also part of the reason why we see a very high convergence rate (frequency) in Figure 4.8.

The convergence time increases with increasing number of ISPs, as we intuitive expect. However the corresponding rate of increment (slope) is fairly modest, particularly for BRITE-generated topologies. The number of ISPs (ASes) in the Internet runs into thousands (tens of thousands) [120]; yet, our results (we did extensive simulations for up to 500 ISPs) hint that stable and efficient prices (and the corresponding flows) should be computable across all ISPs/ASes in the Internet in a reasonable amount of time.

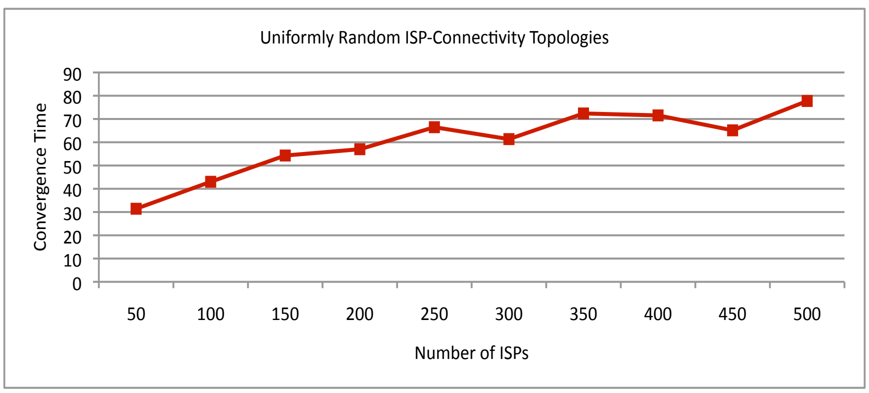


Figure 4.9: Convergence Time (Uniformly-random connectivity).

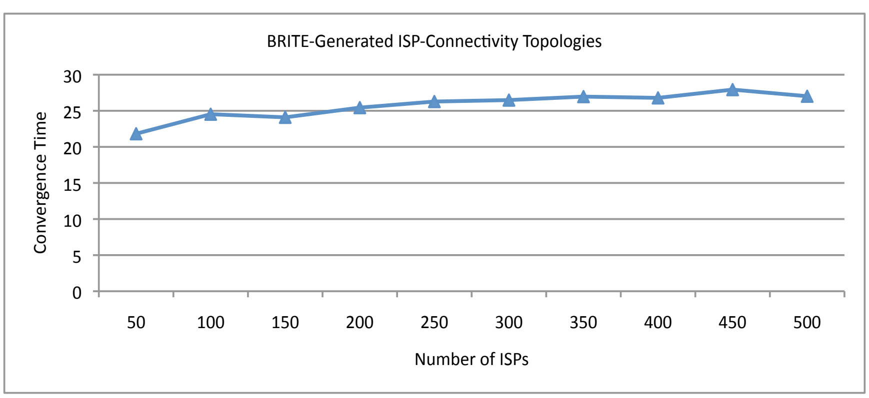


Figure 4.10: Convergence Time (BRITE-generated connectivity).

4.5.2.3 Utility upon Convergence vs Optimum Utility

Next we study the network utility attained on convergence, and discuss how it compares with the optimal utility. Note that the network utility represents the aggregate weighted source-to-destination flow value, where the traffic originating at ISP i is weighted by its utility value (per unit flow) λ_i. Thus this network utility can be viewed as generalized end-to-end throughput attained in the network.

Figures 4.11 and 4.12 show the network utility attained on convergence, as a fraction of the optimum utility. Note that the optimum utility is calculated by solving a weighted maxflow problem in the entire network (from all ISP sources to the single destination ISP). The network utility attained on convergence corresponds to the utility values averaged over the last 100 cycles.

For both types of networks, we observe that the attained utility remains quite close to the optimum, on the average, for networks up to 500 ISPs. While the ratio of attained utility and optimal utility reduces slightly as the number of ISPs increase, this reduction is quite modest – it reduces from 90% to 80% as we increase the number of ISPs from 50 to 500. Note that since the non-monopolistic assumption required for Theorem 15 may not hold in the simulated networks, the price of stability need not be unity. However, our simulations show that there exist stable price-flow solutions that attain high utility in networks on the average, many of which may not satisfy the non-monopolistic assumption.

Note that all the results reported so far all correspond to average statistics; we will look into detailed statistics on the attained utility (for smaller number of ISPs) in Section 4.5.3.

4.5.2.4 Network Utility and Price Evolutions with Time

We now look at how the network utility and edge prices evolve over time in a sample (typical) network, to gain better understanding of the convergence process. For this set of simulations, we consider a single network of 1000 ISPs generated randomly according to the uniformly-random connectivity model. After each cycle, the total utility in the network and the prices of the edges were recorded.

Figure 4.13 shows how the total utility value evolves over time. The figure

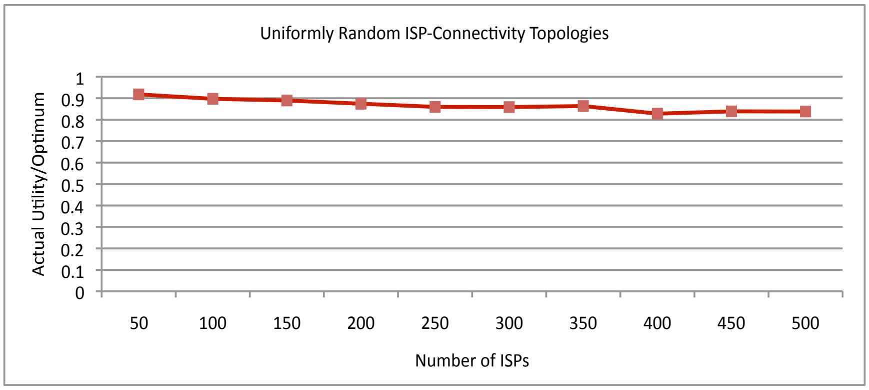


Figure 4.11: Ratio of Average Utility attained on convergence and Optimal Utility (Uniformly-random connectivity).

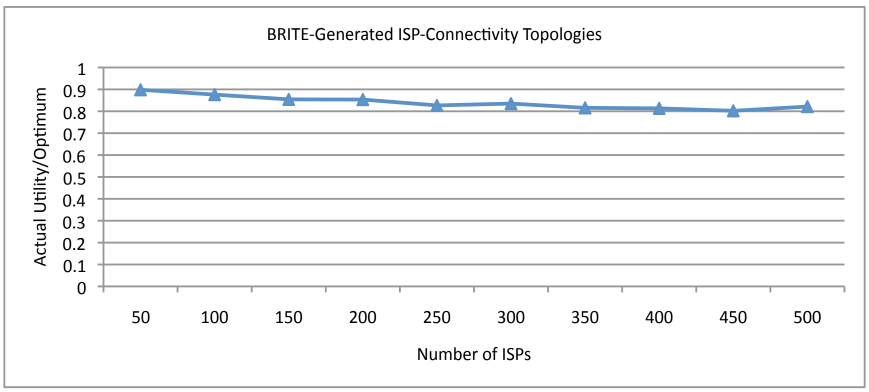


Figure 4.12: Ratio of Average Utility attained on convergence and Optimal Utility (BRITE-generated connectivity).

shows that the network utility reaches a more-or-less stable value fairly quickly (in only 60-70 cycles, which is quite significant, considering the network has 1000 ISPs, and our price update procedure is decentralized); but exhibits small fluctuations around that value for the rest of the simulation time. Thus although convergence occurs fast, it is not perfectly smooth convergence, as some small fluctuations in utility values remain - this behavior was observed to be typical, in our simulation experiments. These fluctuations in network utility are a result of small fluctuations in the price and flow values, and is better understood by looking at sample price evolution charts, as we show next.

Figure 4.14 shows how the incoming edge prices of an ISP (the prices that are set by the ISP) evolve over time, for two different ISPs. Note that each curve in the figure corresponds to a different edge (price variable). In both cases, the price convergence process is quite fast, i.e., convergence occurs within 60 cycles. The price convergence process for ISP 771 is “smooth”, i.e., once the prices converge to their stable values, they do not deviate from those values. We have observed that this kind of behavior is typical for most ISPs. The price convergence process for ISP 997 is however quite different; even though the prices seem to “converge” to some steady-state values, they occasionally fluctuate around those values, and these fluctuations do not die down over time. This kind of fluctuation in price occur for a small fraction of ISPs, in many of our simulations. A closer inspection reveals that the range of these fluctuations is (+/-)1, which is also the step size of the pricing updates. This implies that these fluctuations can be reduced by making the step size of pricing updates smaller, although typically at the cost of having longer convergence times.

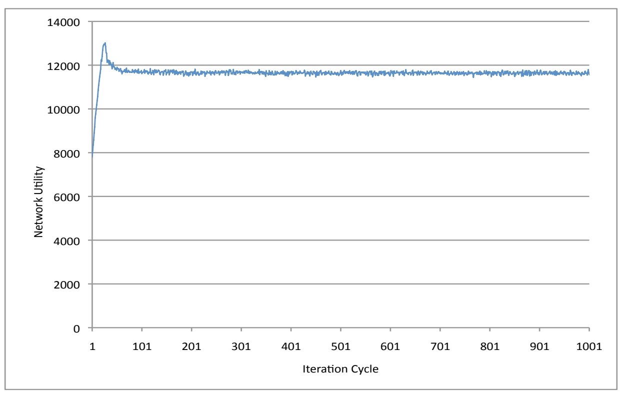
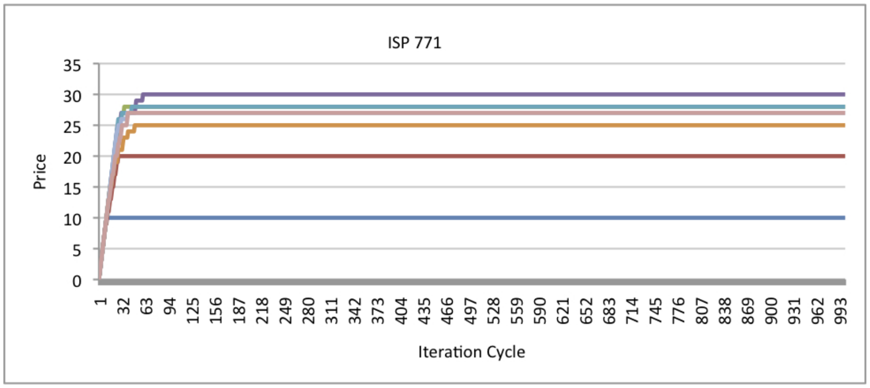


Figure 4.13: Time graph showing the Convergence of the Network Utility for a sample network.

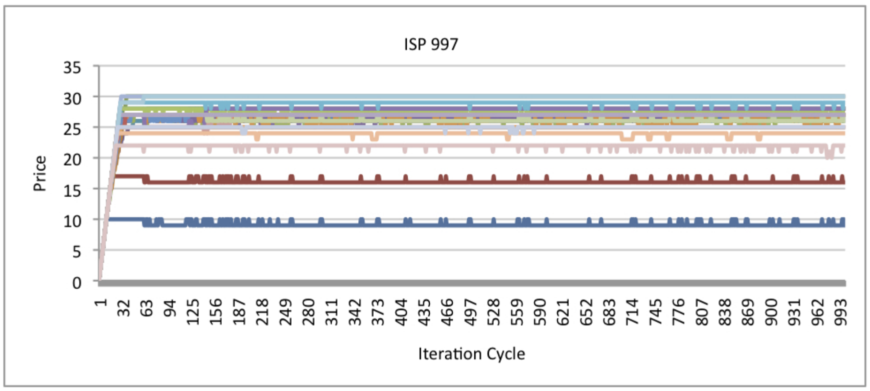
4.5.3 Networks of ISPs with General Topologies

4.5.3.1 Preliminaries

Next we perform our simulation study on networks where ISPs have general topologies, and therefore may not be well-provisioned. We have experimented with both uniformly-random and BRITE-generated connectivity models at both the intra-ISP and inter-ISP levels. Below, we only present results for the case where both intra- and inter-ISP topologies are determined by BRITE; our findings for the other cases are similar. In this case, BRITE is first used to create a network of ISPs



(a)



(b)

Figure 4.14: Time graph of the price changes on the incoming edges of two ISPs in the sample network (the ISP number refers to its position in the topological order).

- in the same way as described in Section 4.5.1. Subsequently, BRITE is used to create a network of router nodes for each of these ISPs. If two ISPs are connected in the upper level (inter-ISP) network, an edge will be put between two routers belonging to the two ISPs (one from each ISP). These "gateway" routers are chosen by finding the lowest degree non-leaf router node in each of the two ISPs. Once the network has been created, the edges are given direction and capacity as follows. The edges between ISPs are assumed directional, and are directed based on the breadth-first-search from the sink ISP (as in Section 4.5.2). Edges within an ISP are bidirectional (can carry flow in both directions) and have capacity one. The capacities of the inter-ISP edges are determined in an iterative manner, traversing the ISPs in the reverse topological order. When considering a specific ISP, a maxflow through (and including) that ISP to the destination is computed, assuming all incoming edges of the ISP have infinite capacity (note that since we are traversing the ISPs in the reverse topological order, the capacities of all edges from that ISP to the sink are already determined by then). Then, for each incoming edge, a random number r, $0.9 \leq r \leq 1.0$, is created, and the edge is given capacity $r \times f$, where f is the amount of flow on this edge in the previously found maxflow. This ensures that every ISP can always forward all traffic that enters its subnetwork on ingress links.

Additionally, for each ISP, a "source node" is chosen at random from all the nodes in the ISP. All flow generated inside an ISP is assumed to originate at that source node; the corresponding λ value is chosen as before - uniformly at random as an integer on $[0, 30]$. For a given set of prices, the flow finding algorithm works as in Section 4.5.1, except that the intra-ISP topology (the intra-ISP network connectivity and link capacities) must be additionally considered in the linear program that finds the flow maximizing the ISP's utility. Prices are changed in the network as before: there is a consistent random permutation of the ISPs, this permutation is iterated through and each ISP is given the opportunity to raise or lower the price on each of its incoming edges based on estimated change in its own utility.

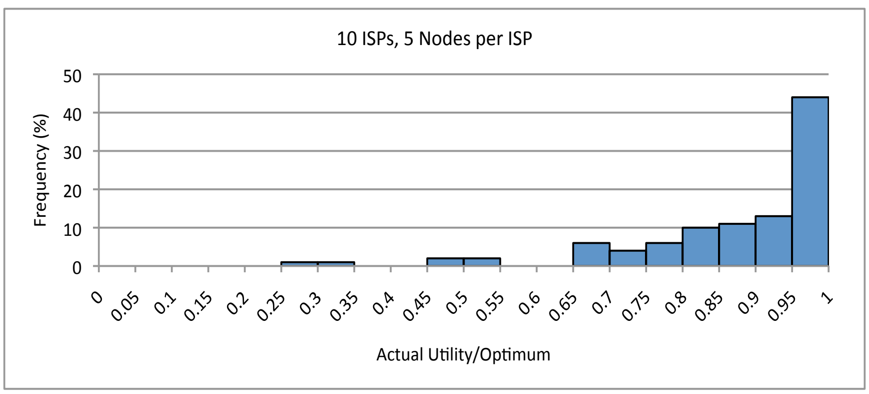
In the general-topology ISP model that we consider here, the intra-ISP topologies need to be considered in our simulations and numerical computations; this practically limits us from simulating networks with very large number of ISPs as before, or having a large number of nodes (routers) per ISP. Below, we show the results for networks up to 50 ISPs with up to 50 nodes per ISP.

The method of determining convergence is the same as that discussed in Section 4.5.1, except that the simulation time is 200 cycles (instead of 300) and convergence is determined based on whether conditions (i) and (ii) holds for 75 consecutive cycles (instead of 100).

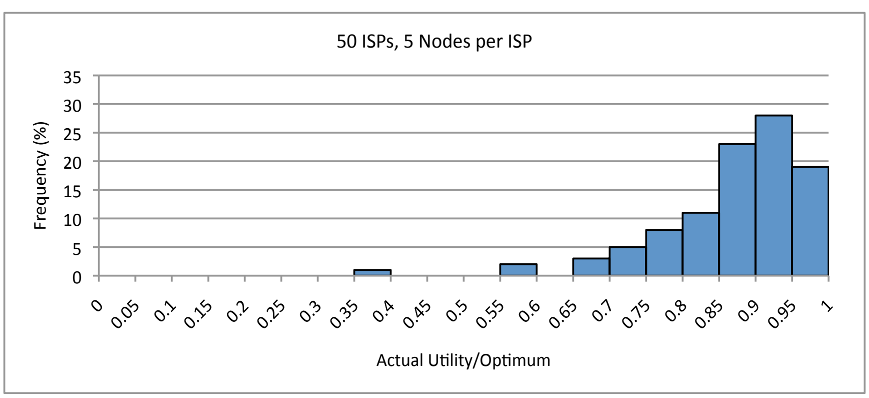
4.5.3.2 Utility upon Convergence vs Optimum Utility

Figures 4.15 and 4.16 show the histograms of network utilities as observed over simulations over 200 network topologies, for each setting of parameters. The attained utility shown is calculated by averaging over the last 75 cycles in the 200-

cycle simulation period. The average utility was observed to be close to 90% of the optimum in all cases, which is consistent with our observations for the case of well-provisioned ISPs (Section 4.5.2). The figures show that for an overwhelming majority of the ISPs, the utility attained on convergence was 80% or more (of the optimum). From the figures, we also observe that, interestingly, the tail seems to improve as the number of ISPs in the network, or the number of nodes (routers) per ISP, increases. For example, for networks of 50 ISPs and 50 nodes per ISP, all of the 200 networks simulated resulted in a (attained utility / optimal utility) ratio of 0.5 or more, on convergence. This implies that the utility ratio distribution attained on convergence of our pricing update policy is likely to improve (i.e., the utility ratio is likely to be more consistently high) as we scale up in terms of number of ISPs, or number of nodes per ISP.

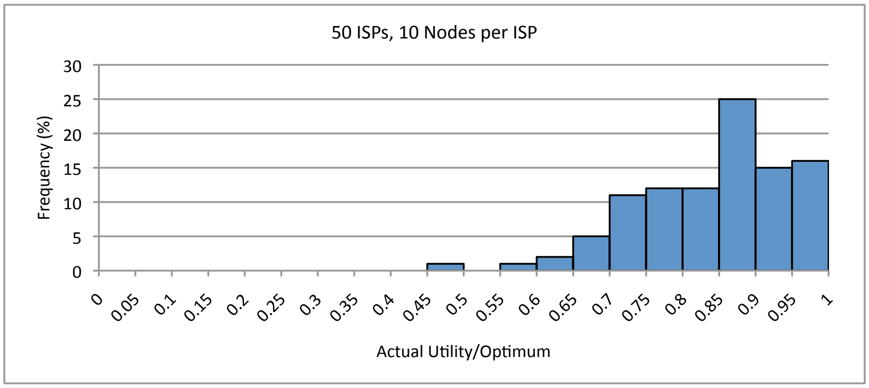


(a)

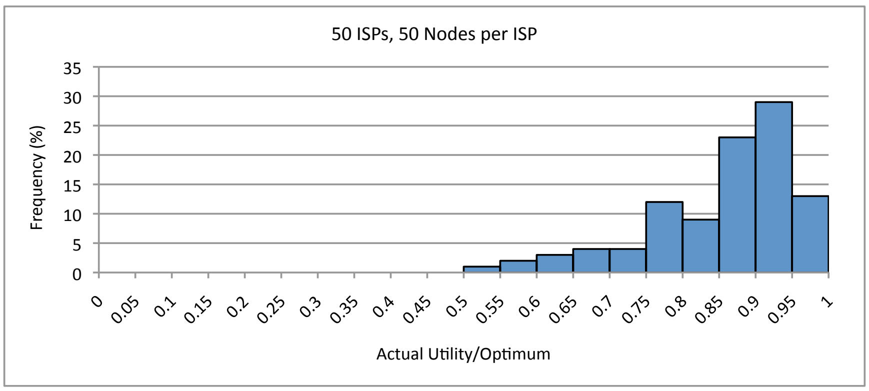


(b)

Figure 4.15: Histogram of Network Utility, for different number of ISPs (2 cases).



(a)



(b)

Figure 4.16: Histogram of Network Utility, for different ISP sizes (2 cases).

4.5.3.3 Convergence Time and Frequency

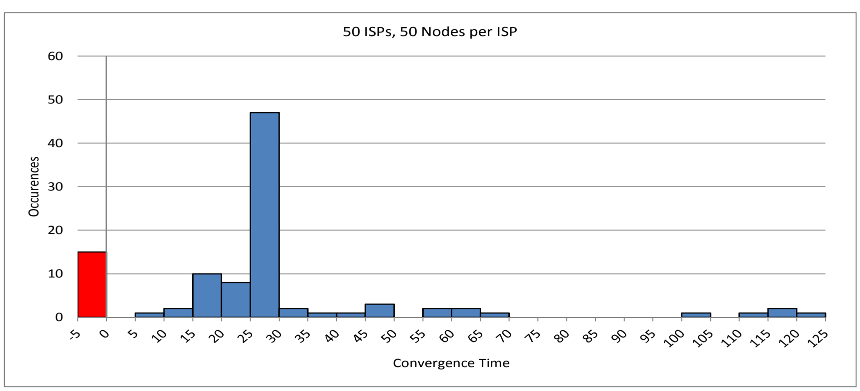


Figure 4.17: Histogram of Convergence Time.

Finally, we look at a representative result of the convergence time and frequency by looking at random BRITE networks with 50 ISPs and 50 nodes per ISP. The result is shown in Figure 4.17. Note that the bar corresponding to the negative value of the convergence time in the figure (in red) corresponds to the frequency (percentage) of non-convergence within the simulation time of 125 cycles. We notice here that for a significant majority of the networks, the pricing update procedure converges very quickly, say within 30 cycles. However, a fairly long tail for the convergence time is observed here - sometimes the pricing updates take fairly long to converge, but this happens for a small fraction of networks. This also explains the high rate of non-convergence observed here (about 15%), which is also partly a result of the short simulation time (125 cycles); the rate of non-convergence would reduce as the total simulation time (the time limit for convergence) is increased. We also observe that the convergence time tail becomes longer (and the non-convergence frequency becomes higher) with increases in the number of ISPs or the number of nodes per ISP; the growth rates are however fairly modest. Note that these facts are consistent with our observations in Section 4.5.2 for well-provisioned ISP networks.

CHAPTER 5
Discussion and Future Directions

5.1 The Firefighter Problem

In our work, we looked at two models: one where the rate of spread of vaccination is the same as the rate of spread of infection; and the other where vaccination does not spread. We provided several positive and negative theoretical results for these models. These models capture the essence of the hardness of the problem but are 'simple' in the sense that they do not capture complexities like probabilistic spread, incubation period for diseases, etc. There is a large potential for research in cases where the Firefighter model is applied to specific areas of application by appropriately subsuming the features and characteristics of the application. Few of the topics on which further work can be carried out are listed below.

- Approximation the MinBurn objective.

In our work we have looked at objectives like MaxSave and MinBudget. Another relevant objective is MinBurn, in which given a budget B, the goal is to find a valid vaccination strategy that minimizes the number of nodes infected. The optimal solution for the MinBurn objective coincides with the optimal solution for the MaxSave objective. But owing to the fact that these problems are NP-complete, we are confined to study their approximate solutions which can be very different from each other.

Unlike the saved nodes, the burnt nodes do not form submodular sets as a function of vaccinated nodes, which was the essential property used for deriving our results for the MaxSave objective. Hence new approach would be necessary in order to solve this problem. One approach that we have found to be useful is to a technique similar to the one used to solve the Unbalanced Graph Cut problem by authors in their paper [47] .

- Is there a possibility of bi-criteria approximation?

Another interesting short term objective is to look at bi-criteria approximations. Instead of saving the maximum possible nodes, say OPT, for a given budget B, an (α, β)-approximation will save at least $\alpha \cdot OPT$ nodes while not exceeding a modified budget of B/β.

Sometimes trying to find an optimal solution for a given objective might prove to be excessively intractable or may lead to a bad result like an high integrality gap or worse - inapproximability. For such objectives it might be the case that some constraint in the model is limiting the range to valid solutions thus forcing a bad result. In these situations it makes sense to relax such a constraint and make it part of a bi-criteria approximation so a that a more favorable result might be obtained. The analysis for bi-criteria approximation may also shed some light on the structural relation between the two said criteria in the model.

- What if only the nodes adjacent to currently infected nodes be vaccinated?

An interesting variation to model is a constraint in which the vaccination is only possible on nodes adjacent to already infected nodes. This constraint may model the requirement of a particular social setting in which the problem is being considered.

Imagine that the Fire-fighter problem is being studied in an epidemiology setting where a disease has to be curbed from spreading. When a person gets infected, individuals that interact with this person face a very high risk of contracting the infection. It may then be considered imprudent or ethically unacceptable that other individuals in a community be given a priority over them for being vaccinated. It may also be the case that associates of an infected individual are most easily accessible for carrying out the vaccinations.

Such constrained situations can be effectively studied in this model.

- Vaccination spreads at a different rate than the infection.

In our work we looked at two models: one where the rate of spread of the vaccination is the same as the rate of spread of the infection; and the other where the rate of spread of vaccination is 0, i.e. it does not spread. In reality the rate of spread of the vaccination may lie somewhere in the middle.

It is not difficult to imagine scenarios where the antidote spreads at rate either faster or slower than the original infectious process. A rival idea for example may require much more convincing or debating and hence may spread at a slower rate. On the other hand a computer virus patch may be disseminated at a faster rate.

The lemmas that were essential to our analysis of the spreading model fail to hold for these intermediate cases. Hence the study of such models give rise many interesting open problems.

- Is the problem easier on DAGs?

Though it is an restrictive assumption on the input graph, the absence of cycles in DAG reduces the number of paths available for the infection to spread. This makes DAGs interesting candidate for analysis of the firefighter problem.

- To isolate the infection as early as possible.

When the input graph is very large, possibly infinite, another objective worth considering would be to cut off the infection as soon as possible. Cutting off the infection implies that even if no further vaccinations are carried out, the infection will not spread to the complete graph. The aforementioned objective tries to achieve this goal in as few time-steps as possible. This may be a good objective for situations where the nature or rate of spread of the infection is uncertain and a quarantine is desirable as soon as possible.

- Variable cost of vaccination.

In the model considered in our work, every vaccination costs one unit of the available budget. We could also consider an interesting generalization where the cost of a vaccination is function of the time-step at which it is administered. Most natural candidate for such a function would be one that decrease or does not increase with time. This models the reality that as time passes, the pool of available resources required for administering vaccines increases; and hence its cost falls.

Objectives like MaxSave can be studied on this model where the budget itself can also be a function of time.

- Edges with associated probabilities.

If there exists a vulnerable node next to an infected node then in the next time-step the infection spreads to the vulnerable node. This behavior signifies that the probability of the infection being transmitted is 1.

This model can be generalized to incorporate susceptibility of individuals to the infection. Every edge can have an associated probability which gives the probability of infection spreading through this edge.

In scenarios where the spreading entity is an idea, these probabilities could signify the amount of trust shared by the adjacent nodes.

- Placing sentries on nodes that send alerts on encountering infection.

A related framework for studying the spread of infection and its subsequent containment is as follows:

Place sentries initially on a subset of nodes of the graph. Whenever a sentry encounters the infection, it starts sending out alerts to all nodes so that they can take requisite precautionary measures. In this model the sentries are placed before the source of infection is known.

One of the objectives worth considering in this model is to place a fixed number of sentries such that the number of infected nodes resulting from the worst possible source of infection is minimized.

- Infected nodes become contagious after a period of incubation.

Some diffusive processes may need to incubate inside the host before they become contagious. Some diseases have a significant incubation time before the symptoms start showing and the disease becomes infectious. It is also not difficult to imagine that when new ideas, such as adoption of a new product, spread there may exist a considerable lag between adoption and spread. In the case of adopting a new product, the user may want to try it for some time before recommending it to his acquaintance.

Such a lag between initial contact and infectious behavior adds a significant level of complexity to the model. Additionally, the lag time can be function of the node, i.e. can be different for different nodes.

Study of objectives like MaxSave and MinBudget on such general models would be an interesting long term research goal.

- Infected nodes recover on their own.

Another property of diseases that we have not considered in our simplified model is that some patients who contract the disease recover on their own. This property has been extensively studied in epidemiology literature under the Susceptible-Infectious-Recovered model.

After a node has reached the *recovered* stage, it no longer is infectious and neither is it susceptible. The effect is then similar to disconnecting the node from the graph altogether. Further generalization can be made by making the time to recover vary from node to node.

Also note that a similar effect is observed if the node 'dies' due to the infection instead of recovering.

5.2 Diffusion in Complex Networks

The paradigm developed by us in which instances of complex diffusion model are simplified in order to leverage properties like submodularity to efficiently calculate good seed sets has a wide range of applications. This technique can be applied to other general diffusion models developed for a specific scenarios. Of course, properties like submodularity may not be be applicable everywhere; however, the goal is to make the model theoretically tractable enough so that new or existing results can be used to achieve the objective efficiently. Some of the possible directions for further work in this area are as follows:

- Predicting optimal threshold for Projected Greedy calculation.

The efficiency of our method depends on finding a good seed set using the simplified model. We generate several seed sets for a tunable parameter, in this case the threshold t for Projected Greedy calculation, and select the one that gives us the best performance in simulations. It will be highly advantageous if we could make an accurate prediction about the optimal threshold t_{opt}. This would help us reduce the over all running time for out heuristic, although only by a constant factor. There are a number of general model parameters like graph type, upper and lower thresholds, time-steps to evacuation, information fusion parameters, and probability of successful message transmission that can affect the optimal threshold value. We have already seen that the t_{opt} value tends to decrease as λ_d value increases. We can intuitively expect that the probability of successful message transmission p_t most probably does not have any effect on the t_{opt} value. Increasing p_t may merely increase the proportion of nodes evacuated for all threshold values.

It will however be interesting to study the effect of upper and lower thresholds and also the gap between them, on t_{opt}. Relevant future work can consist of obtaining simulation results for different sets of these parameter values and performing linear regression to measure their effect on t_{opt}.

- Targeted immunization to contain diffusive spread.

The paradigm used by us can be applied to numerous other problems. One such goal is to contain or stop a diffusive process. Faced with an epidemic that is spreading through a population, and a limited supply of vaccine (or simply a lack of time to administer it), it is necessary to decide whom to vaccinate. This is also true for many other diffusive processes, such as canceling an evacuation alert, propagating computer patches that address vulnerabilities, and viral counter-marketing, where a firm attempts to undermine a word-of-mouth marketing campaign by its competitors. Efficient heuristics for targeted immunization in a variety of settings can be grouped at a high level into non-infectious vaccination, infectious vaccination, and real-time vaccination. These problems were already introduced and motivated in discussion on the Firefighter problem. In fact, the Firefighter problem is one possible theoretical model for real-time vaccination.

Non-Infectious Vaccination One problem that can be considered is the so-called prophylactic vaccination, i.e., vaccination that is done before an epidemic begins, in order to lessen the impact of such an epidemic. This is essentially a question about cuts in graphs, since vaccinating a node in this context means that this node cannot spread the information/infection to its neighbors, and thus is essentially removed from the network. Several recent papers modeled vaccination using graph cuts. For example, the work of Hayrapetyan et al. [47] and others [48, 122, 46, 44, 80], fully utilizes the social network structure to cut off and contain various diffusive processes in a social network. The papers mentioned above are mostly concerned with efficient approximation algorithms for simple diffusion processes (e.g., deterministic independent cascade [39]). Note that most of this work is not based on properties such as submodularity (which does not hold for this context), but instead utilize mostly techniques from optimization and traditional approximation algorithms. Two of the conclusions that are clear from existing work is that designing good approximation algorithms for general diffusion models is difficult, and that the method of finding a good set to immunize depends crucially on whether the initial set of infected nodes is known, random, adversarial, etc. We can apply our design methodology to form efficient heuristics for finding a set of nodes to immunize. This will likely require extending the theoretical results mentioned above to slightly more general models of diffusion, such as the case with transmission probabilities, threshold models [39, 42], the case when the starting point of the infection is unknown, and when people are able to recover (or die) from the infection. This theoretical work provides a large toolkit of simplified diffusion processes that admit provably good approximation algorithms. At this point, our methodology can be applied so that, given an instance of the general diffusion process in Section 3.1, one could form an appropriate instance of a simplified diffusion process (choosing a good one from our toolkit), use existing heuristics on this simplified instance, and then transform the results into results for the general instance. Choosing exact simplified instances that would behave similarly to the general instances forms the bulk of the work, and requires both theoretical insight and extensive simulations. In addition to attempting to form a good seed set to immunize, one can also consider combined objectives: How do we seed the network for maximizing spread, but it such a way that it is easy to cancel the spread of information (e.g., in the case of a false evacuation alert)? Given that the network is optimized for spreading information, how can we contain it? In general, there is an intuitive tradeoff between the ability of a network to spread information, and the ability to impede this spread. It would be very interesting to determine if this intuitive tradeoff arises in real-world network topologies, and is actually a true tradeoff.

Infectious Vaccination Another important model of immunization that can be studied is one where vaccination is also a process that spreads through the network. In the case of ideas propagating through a social network, this represents the fact that an antidote to a harmful idea is often another idea, which can be just as infectious. This also occurs in the propagation of computer patches through a social network, and in canceling evacuation alerts. For this case, immunizing a node does not just mean removing it from the network, but also means starting a competing cascade. In Section 2.2, by using results about submodular functions on partition matroids, we were able to provide an algorithm that saves at least 63.2% of the maximum possible, for a simple deterministic diffusion model. Similar results become much more difficult to prove once we add some natural complexities to the problem, such as having nodes leave the network after reaching a certain threshold of information. However, we believe that results about submodularity can still be used to great advantage here, since while for a deterministic model in this context, the number of un-infected people is not a submodular function of the set we choose to vaccinate, it is still submodular for certain distributions in the probabilistic model. Probabilistic diffusion processes also give rise to a new natural objective: to minimize the probability of a large epidemic (instead of minimizing the expected size of an epidemic). Unlike in the case of non-infection vaccination, we believe that similar techniques to the ones we used in Chapter 3 have a chance of succeeding here; although here the goal is to stop the diffusive process instead of to maximize it, the same properties of submodularity often hold for simple version of our model.

Real-Time Vaccination We have studied the Firefighter problem Chapter 2. This problem is extremely interesting from the theoretical point of view since, at its essence, the goal is to form a cut over time so that in total the nodes vaccinated at all the time-steps stop the infection. Our methodology can be applied to vaccination in real-time (i.e., approximation algorithms for simple models, choose appropriate simplifications, and use simpler models to design heuristics for more complex ones). Doing this is likely to require very different techniques from prophylactic vaccination, as the area of cuts over time is not nearly as developed as that of regular cuts. In general, the dynamic component of the seeding adds an extremely interesting (though complex) dimension to the space of possible heuristics.

5.3 Strategic Pricing in Next-hop Inter-domain Routing

Research in the field of strategic next-hop routing is relatively new and there are many issues that need to be tackled effectively. Modeling a meaningful game that incorporates characteristics observed in real systems without including too many restrictive assumptions was a major bone of contention that we came across. As we have seen, issues such as monopolies and tie-breaking rules required careful consideration. Inter-domain routing on the Internet is an important application of strategic next-hop routing and more research needs to be carried out to include its complexities in a game theoretic framework. Following are some of the topics that can be researched further.

- Relation between step-size and $max(\lambda_i)$.

We observed in Section 4.5 that given our dynamic price update strategy, networks may not converge to an equilibrium. The criteria for defining 'convergence' is very strict. Even so, we observe that some networks converge to a steady state value, but then fluctuate around that value. An important observation is that the fluctuations are of size ± 1, which is the step-size. This could mean that the granularity of step-size is not small enough for a player to reach it's true steady state value. Further simulations can be carried out to study how the ratio of $max(\lambda_i)$ to step-size affects this phenomenon. These simulations can help determine whether a sufficiently small value of step-size as compared to $max(\lambda_i)$ will help avoid these fluctuating steady states.

On the other hand, reducing the step size will lead to increase in convergence time since it will take more iterations for prices to reach their steady state values.

Hence a smarter update method of adapting the step-size is worth studying. One such approach could be to begin with large step size and then reduce its value with increasing iterations.

- Heterogenous price updates.

  We have assumed that all ISPs update their prices using the same step-size. In realistic scenarios, ISPs may not agree on the step-size. It is reasonable to assume that they may select a step-size that is proportional to their λ_i values. It would be interesting to study how heterogenous values of step-size affects results observed in our simulation.

- Price of Anarchy.

  In our work, we have shown that Price of Anarchy for networks with single source is 1, under certain reasonable assumptions. An immediate open problem is: What about PoA with multiple sources? How bad can a Nash equilibrium get? Results from our simulation would suggest that the price of anarchy may not be very large given that most networks converge to outcomes that are close to the social optimal. However, there is always a possibility that there exists a really bad example that results in a high Price of Anarchy.

- Players owning non-adjacent nodes.

  Another interesting generalization that can be studied is the case where players can own arbitrary number of nodes in the network. This models the more realistic scenario where an ISP owns multiple Autonomous Systems in the Internet. Many interesting issues arise in this situation. First of all, the conditions on networks have to be well defined to make sure that monopolies do not exist. One player owning multiple nodes makes condition for monopoly tough to assess and define. Also, since players can now incur loss at one of their nodes while at the same time obtaining a larger profit at another, conditions for Nash equilibrium will depend much more on the actual network structure.

- Multiple sinks.

In our game, traffic originates at multiple sources and is directed towards a single destination. A natural extension here is to consider multiple sinks. However, defining a coherent game for multiple sinks is not a trivial extension of the single sink version. There are several issues that need to be tackled. We need to make sure that traffic for any sink cannot cycle in the network since this leads to outcomes that are not well defined. If players can allocate fractional capacities on their incoming edges to traffic depending on its destination, then it becomes difficult to avoid monopolies. For the single sink version, the non-monopolistic property of edges avoids this situation. Hence more work is required to determine an appropriate model for the multi-sink version of the next-hop routing game.

- Two and three stage games.

We have defined a one stage game where given the pricing strategy and tie-breaking strategy of players, a unique outcome can be determined. Alternatively, this game can also be modeled as a two stage game where in the first stage players decided only the pricing strategy. Then in the next stage, the players forward flow and send their own flow on their outgoing edges in a manner that maximizes their profit. This model obviates the need for having a tie-breaking strategy. It can be argued that the original game described in Chapter 4 is a "general" version of this two stage game. That is to say, all equilibrium pricing strategies in the two stage version of the game can be mapped to equilibrium pricing strategies in the original game but not vice-versa. The concept of two stage game is advantageous in that it captures in some way the different time scales on which the two decisions are made. The price setting on edges is an abstraction for contracts between ISPs. Once made, these contracts are set for longer time periods as compared to routing decisions that can changed much frequently.

In fact, one can also define a three stage game where the stages are as follows:

i. Players decide to augment or reduce the capacities on their incoming edges.

ii. Players decide prices for their incoming edges.

iii. Players decide to route flow.

This game also takes into consideration the fact that players can physically change the capacity on their incoming edges. It is reasonable to assume that such decisions are made on a much longer time scale as compared to the contractual and routing decisions. Note that the there is only one set of players. We can divide the strategies into different stages only because the types of decisions made by players are inherently different. Also, the decisions made at one stage affects the strategy at next level.

5.4 Concluding Remarks

In this thesis we have made several contributions to the study of diffusive processes and selfish next-hop routing. Using the model of the Firefighter problem, we provided theoretical analysis and algorithms with worst case performance bounds for the problem of inhibiting the spread of malicious entities in networks using limited resources. We have answered some of the open questions, especially for the non-spreading model. Building on this work, we developed a method of applying theoretical results to more complex diffusion models. We also looked at the problem of maximizing efficiency of Internet and Internet like systems by optimally allocating scarce resources using diffusion of prices through the network.

As described in previous sections of this chapter, there are a number of interesting directions that can be further pursued. But, we believe that the work in this thesis is an important first step towards gaining a more comprehensive understanding of these fields.

BIBLIOGRAPHY

[1] E. Anshelevich, F. Shepherd, and G. Wilfong, "Strategic network formation through peering and service agreements," *Games and Econ. Behavior*, vol. 73, no. 1, pp. 17–38, Sept. 2011.

[2] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker, "A BGP-based mechanism for lowest-cost routing," *Distrib. Comput.*, vol. 18, pp. 61–72, Jul. 2005.

[3] C. H. Papadimitriou and G. Valiant, "A new look at selfish routing," in *Proc. of ICS*. Beijing, China: Tsinghua University Press, 2010, pp. 178–187.

[4] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in *Proc. of INFOCOM 2001. 20th Annu. Joint Conf. of the IEEE Comput. and Commun. Soc. IEEE*, vol. 3, Anchorage, Alaska, 2001, pp. 1300 –1309 vol.3.

[5] K. Kar, M. Kodialam, and T. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 12, pp. 2566 –2579, dec 2000.

[6] R. Guerin, A. Orda, and D. Williams, "QoS routing mechanisms and OSPF extensions," in *Global Telecommun. Conf., 1997. GLOBECOM '97., IEEE*, vol. 3, Phoenix, AZ, Nov. 1997, pp. 1903 –1908 vol.3.

[7] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," *RFC 4271*, no. 4271, pp. 1–105, 2006.

[8] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Accurate and efficient SLA compliance monitoring," *ACM SIGCOMM Comput. Commun. Review*, vol. 37, no. 4, p. 109, 2007.

[9] M. Yuksel, A. Gupta, K. Kar, and S. Kalyanaraman, *Next-Generation Internet Architectures and Protocols*, B. Ramamurthy, G. N. Rouskas, and K. M. Sivalingam, Eds. Cambridge, UK: Cambridge University Press, December 2010.

[10] C. Labovitz, S. Iekel-johnson, A. Arbor, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *Commun.*, vol. 40, no. 4, pp. 75–86, 2010.

[11] A.-L. Barabási, R. Albert, and H. Jeong, "Mean-field theory for scale-free random networks," *Physica A: Statistical Mechanics and its Applications*, vol. 272, no. 1-2, pp. 173–187, 1999.

[12] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, "Stochastic models for the web graph," in *Proc. of the 41st Annu. Symp. on Found. of Comput. Sci.*, ser. FOCS '00. Washington, DC: IEEE Computer Society, 2000, pp. 57–65.

[13] S. Eubank, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, and N. Wang, "Structural and algorithmic aspects of massive social networks," in *Proc. of the 15th Annu. ACM-SIAM Symp. on Discrete Algorithms*, ser. SODA '04. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2004, pp. 718–727.

[14] D. J. Watts and S. H. Strogatz, "Collective dynamics of small world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[15] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos, "Epidemic thresholds in real networks," *ACM Tran. on Inform. and Syst. Security*, vol. 10, no. 4, pp. 1–26, 2008.

[16] C. Borgs, J. Chayes, A. Ganesh, and A. Saberi, "How to distribute antidote to control epidemics," *Random Structures and Algorithms*, vol. 37, no. 2, pp. NA–NA, 2010.

[17] M. E. J. Newman, "The spread of epidemic disease on networks," *Phys. Rev. Lett.*, vol. 66, p. 016128, 2002.

[18] R. Chellappa and A. Jain, Eds., *Markov Random Fields: Theory and Application.* Waltham, Mass: Academic Press, 1993.

[19] N. Bailey, *The Mathematical Theory of Infectious Diseases and its Applications.* London: Griffin, 1975.

[20] M. A. Nowak and R. M. May, *Virus Dynamics: Mathematical Principles of Immunology and Virology.* Oxford: Oxford University Press, 2000, vol. 291, no. 5506.

[21] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. of INFOCOM 2005. 24th Annu. Joint Conf. of the IEEE Comput. and Commun Soc. IEEE*, vol. 2, Miami, FL, March 2005, pp. 1455 – 1466 vol. 2.

[22] C. Moore and M. E. J. Newman, "Epidemics and percolation in small-world networks," *Phys. Rev. E*, vol. 61, pp. 5678–5682, May 2000.

[23] R. Pastor-Satorras and A. Vespignani, "Epidemic dynamics in finite size scale-free networks," *Phys. Rev. E*, vol. 65, p. 035108, Mar 2002.

[24] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, ser. KDD '07. New York, NY: ACM, 2007, pp. 420–429.

[25] J. Aspnes, N. Rustagi, and J. Saia, "Worm versus alert: who wins in a battle for control of a large-scale network?" in *Proc. of the 11th Int. Conf. on Principles of Distributed Syst.*, ser. OPODIS'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 443–456.

[26] G. Baier, T. Erlebach, A. Hall, E. Köhler, P. Kolman, O. Pangrác, H. Schilling, and M. Skutella, "Length-bounded cuts and flows," *ACM Trans. Algorithms*, vol. 7, pp. 4:1–4:27, December 2010.

[27] N. Sonnerat and A. Vetta, "Defending planar graphs against star-cutsets," *Electron. Notes in Discrete Math.*, vol. 34, pp. 107–111, 2009.

[28] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions, II," in *Polyhedral Combinatorics*, ser. Mathematical Programming Studies, M. L. Balinski, A. J. Hoffman, R. W. Cottle, L. C. W. Dixon, B. Korte, M. J. Todd, E. L. Allgower, W. H. Cunningham, J. E. Dennis, B. C. Eaves, R. Fletcher, D. Goldfarb, J.-B. Hiriart-Urruty, M. Iri, R. G. Jeroslow, D. S. Johnson, C. Lemarechal, L. Lovasz, L. McLinden, M. J. D. Powell, W. R. Pulleyblank, A. H. G. Rinnooy Kan, K. Ritter, R. W. H. Sargent, D. F. Shanno, L. E. Trotter, H. Tuy, R. J. B. Wets, E. M. L. Beale, G. B. Dantzig, L. V. Kantorovich, T. C. Koopmans, A. W. Tucker, and P. Wolfe, Eds. Philadelphia, PA: Springer Berlin Heidelberg, 1978, vol. 8, pp. 73–87.

[29] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM J. on Comput.*, vol. 40, no. 6, pp. 1740–1766, 2011.

[30] T. W. Valente, "Network models of the diffusion of innovations," *Comput. & Math. Organization Theory*, vol. 2, pp. 163–164, 1996.

[31] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proc. of the 13th Int. Conf. on World Wide Web*, ser. WWW '04. New York, NY: ACM, 2004, pp. 491–501.

[32] S. Hill, F. Provost, and C. Volinsky, "Network-based marketing: Identifying likely adopters via consumer networks," *Stat. Sci.*, vol. 21, pp. 256–276, 2006.

[33] J. L. Iribarren and E. Moro, "Information diffusion epidemics in social networks," *Phys. Rev. Lett.*, vol. 103, p. 038702, 2009.

[34] A. Java, P. Kolari, T. Finin, and T. Oates, "Modeling the Spread of Influence on the Blogosphere," University of Maryland, Baltimore County, Baltimore, MD, Tech. Rep., March 2006.

[35] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, May 2007.

[36] J. Leskovec, A. Singh, and J. Kleinberg, "Patterns of influence in a recommendation network," in *Proc. of the 10th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining*, ser. PAKDD'06. Singapore: Springer-Verlag, 2006, pp. 380–389.

[37] D. Strang and S. A. Soule, "Diffusion in organizations and social movements: From hybrid corn to poison pills," *Annu. Review of Sociol.*, vol. 24, no. 1, pp. 265–290, 1998.

[38] X. Wan and J. Yang, "Learning information diffusion process on the web," in *Proc. of the 16th Int. Conf. on World Wide Web*, ser. WWW '07. New York, NY: ACM, 2007, pp. 1173–1174.

[39] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, ser. KDD '03. New York, NY: ACM, 2003, pp. 137–146.

[40] D. Kempe, J. Kleinberg, and . Tardos, "Influential nodes in a diffusion model for social networks," *Automata Languages and Programming*, vol. 3580, no. 1, pp. 99–120, 2006.

[41] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *Proc. of the 3rd Int. Conf. on Internet and Network Econ.*, ser. WINE'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 306–311.

[42] P. A. D. Jr. and F. S. Roberts, "Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion," *Discrete App. Math.*, vol. 157, no. 7, pp. 1615 – 1627, 2009.

[43] N. Chen, "On the approximability of influence in social networks," *SIAM J. on Discrete Math.*, vol. 23, no. 3, pp. 1029–1037, 2008.

[44] G. Giakkoupis, A. Gionis, E. Terzi, and P. Tsaparas, "Models and algorithms for network immunization," Department of Computer Science, University of Helsinki, Tech. Rep. C-2005-75, 2005.

[45] D. Chakrabarti and C. Faloutsos, "Epidemic spreading in real networks: an eigenvalue viewpoint," *Proc. of 22nd Int. Symp. on Rel. Distributed Syst. 2003*, vol. pages, no. 22, pp. 25–34, 2003.

[46] R. Engelberg, J. Könemann, S. Leonardi, and J. S. Naor, "Cut problems in graphs with a budget constraint," *J. of Discrete Algorithms*, vol. 5, pp. 262–279, June 2007.

[47] A. Hayrapetyan, D. Kempe, M. Pál, and Z. Svitkina, "Unbalanced graph cuts," in *Proc. of the 13th Annu. European Conf. on Algorithms*, ser. ESA'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 191–202.

[48] J. Aspnes, K. Chang, and A. Yampolskiy, "Inoculation strategies for victims of viruses and the sum-of-squares partition problem," *J. of Comput. and Syst. Sci.*, vol. 72, no. 6, pp. 1077–1093, 2006.

[49] B. Hartnell, "Firefighter! an application of domination," presented at the 25th Manitoba Conf. on Combinatorial Math. and Comput., University of Manitoba, Winnipeg, Canada, 1995.

[50] G. MacGillivray and P. Wang, "On the firefighter problem," *J. of Combinatorial Math. and Combinatorial Comput.*, vol. 47, pp. 83–96, 2003.

[51] P. Fogarty, "Catching fire on grids," Master's thesis, Dept. of Math., Univ. of Vermont, Burlington, Vermont, 2003.

[52] M. Develin and S. G. Hartke, "Fire containment in grids of dimension three and higher," *Discrete Appl. Math.*, vol. 155, pp. 2257–2268, October 2007.

[53] S. Crosby, A. Finbow, B. Hartnell, R. Moussi, K. Patterson, and D. Wattar., "Designing fire resistant graphs," *Congr. Numerantium*, vol. 173, pp. 207–222, 2005.

[54] C. Leizhen and W. Weifan, "The surviving rate of a graph for the firefighter problem," *SIAM J. on Discrete Math.*, vol. 23, no. 4, pp. 1814–1826, 2009.

[55] A. King and G. MacGillivray, "The firefighter problem for cubic graphs," *Discrete Math.*, vol. 310, no. 3, pp. 614 – 621, 2010, sixth Czech-Slovak International Symposium on Combinatorics, Graph Theory, Algorithms and Applications.

[56] L. Cai, E. Verbin, and L. Yang, "Firefighting on trees: $(1-1/e)$ approximation, fixed parameter tractability and a subexponential algorithm," in *Algorithms and Computation*, ser. Lecture Notes in Computer Science, S.-H. Hong, H. Nagamochi, and T. Fukunaga, Eds. Hong Kong SAR China: Springer Berlin / Heidelberg, 2008, vol. 5369, pp. 258–269.

[57] B. Hartnell and Q. Li, "Firefighting on trees: how bad is the greedy algorithm?" *Congr. Numerantium*, no. 145, pp. 192–197, 2000.

[58] P. Chalermsook and J. Chuzhoy, "Resource minimization for fire containment," in *Proc. of the 21st Annu. ACM-SIAM Symp. on Discrete Algorithms*, ser. SODA '10. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010, pp. 1334–1349.

[59] J. H. Sorensen, "Modeling human response to a chemical weapons accident," in *DIMACS Working Group on Modeling Social Responses to Bio-terrorism Involving Infectious Agents*, New Bruncswick, NJ, 2003.

[60] J. H. Golden and C. R. Adams, "The tornado problem: Forecast, warning, and response," *Natural Hazards Review*, vol. 1, no. 2, pp. 107–118, 2000.

[61] M. K. Lindell, C. S. Prater, and W. G. Peacock, "Organizational communication and decision making for hurricane emergencies," *Natural Hazards Rev.*, vol. 8, no. 3, pp. 50–60, 2007.

[62] M. K. Lindell and C. S. Prater, "Critical behavioral assumptions in evacuation time estimate analysis for private vehicles: Examples from hurricane research and planning." *J. of Urban Planning and Develop.*, vol. 133, no. 1, pp. 18–29, 2007.

[63] J. H. Sorensen, "Hazard warning systems: review of 20 years of progress," *Natural Hazards Rev.*, vol. 1, no. 2, pp. 119–125, 2000.

[64] D. Mileti, *Disasters by Design: A Reassessment of Natural Hazards in the United States*. Washington, DC: The National Academies Press, 1999.

[65] C. on Disaster Research in the Social Sciences: Future Challenges and N. R. C. Opportunities, *Facing Hazards and Disasters: Understanding Human Dimensions*. Washinton, D.C.: The National Academies Press, 2006.

[66] M. K. Lindell and C. S. Prater, "Estimating evacuation time components: Lessons from nuclear power plants, hurricanes, and the first world trade center bombing," in *National Institute for Standards and Technology Workshop on Building Occupant Movement During Fire Emergencies*, Gaithersburg, MD, 2004, pp. 91–95.

[67] R. W. Perry and M. K. Lindell, "Understanding citizen response to disasters with implications for terrorism," *J. of Contingencies and Crisis Management*, vol. 11, no. 2, pp. 49–60, 2003.

[68] J. H. Sorensen and D. S. Mileti, "Decision-making uncertainies in emergency warning system organizations," *Int. J. of Mass Emergencies and Disasters*, vol. 5(1), pp. 33–61, 1987.

[69] J. Sorensen, B. Vogt, and D. Mileti, *Evacuation: an assessment of planning and research*. Oak Ridge, Tennessee: Oak Ridge National Laboratory, 1987.

[70] B. Aguirre, *Planning, warning, evacuation, and search and rescue: a review of the social science research literature.* College Station, TX: Hazard Reduction & Recovery Center, 1994.

[71] A. W. Ding, "A theoretical model of public response to the homeland security advisory system," *The J. of Defense Modeling and Simulation: Applicat., Methodology, Technology*, vol. 3, no. 1, pp. 45–55, January 2006.

[72] G. O. Rogers and J. H. Sorensen, *Risk Analysis Prospects and Opportunities.* New York: Plenum Press, 1991, ch. Diffusion of Emergency Warning: Comparing Empirical and Simulation Results, pp. 117 – 134.

[73] J. Sorensen, U. S. D. of Energy, M. M. E. S. Inc., and O. R. N. Laboratory, *Assessment of the need for dual indoor/outdoor warning systems and enhanced tone alert technologies in the Chemical Stockpile Emergency Preparedness Program.* Oak Ridge, Tennessee: Oak Ridge National Laboratory, 1992.

[74] M. Lindell and R. Perry, *Communicating environmental risk in multiethnic communities*, ser. Communicating effectively in multicultural contexts. CA: Sage Publications, 2004.

[75] R. W. Perry, M. K. Lindell, and M. R. Greene, "Crisis communications: Ethnic differentials in interpreting and acting on disaster warnings," *Social Behavior and Personality: an Int. J.*, vol. 10, no. 1, pp. 97–104, 1982.

[76] R. Perry and A. Mushkatel, *Minority Citizens in Disasters.* Athens, GA: University of Georgia Press, 2008.

[77] T. Roughgarden, "Selfish routing with atomic players," in *Proc. of the 16th Annu. ACM-SIAM Symp. on Discrete Algorithms*, ser. SODA '05. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2005, pp. 1184–1185.

[78] ——, *Selfish Routing and the Price of Anarchy.* Cambridge, Mass.: MIT Press, 2005.

[79] D. Acemoglu, R. Johari, and A. Ozdaglar, "Partially optimal routing," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1148–1160, 2007.

[80] E. Anshelevich, B. Caskurlu, and A. Hate, "Strategic multiway cut and multicut games," in *Proc. of the 8th Int. Conf. on Approx. and Online Algorithms*, ser. WAOA'10. Liverpool, UK: Springer-Verlag, 2011, pp. 1–12.

[81] E. Anshelevich and G. Wilfong, "Network formation and routing by strategic agents using local contracts," in *Proc. of the 4th Int. Workshop on Internet and Network Econ.*, ser. WINE '08. Shanghai, China: Springer-Verlag, 2008, pp. 386–393.

[82] A. Archer and E. Tardos, “Frugal path mechanisms,” *ACM Trans. on Algorithms*, vol. 3, no. 1, pp. 991–999, 2002.

[83] A. Hall, E. Nikolova, and C. Papadimitriou, “Incentive-compatible interdomain routing with linear utilities,” in *Proc. of the 3rd Int. Conf. on Internet and Network Econ.*, ser. WINE’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 232–244.

[84] A. Hayrapetyan, E. Tardos, and T. Wexler, “A network pricing game for selfish traffic,” in *Proc. of the 24th Annu. ACM Symp. on Principles of Distributed Computing*, ser. PODC ’05. New York, NY: ACM, 2005, pp. 284–291.

[85] R. Johari, S. Mannor, and J. N. Tsitsiklis, “A contract-based model for directed network formation,” *Games and Econ. Behavior*, vol. 56, no. 2, pp. 201–224, Aug. 2006.

[86] H. Levin, M. Schapira, and A. Zohar, “Interdomain routing and games,” in *Proc of the 40th Annu. ACM Symp. on Theory of Comput.*, ser. STOC ’08. New York, NY: ACM, 2008, pp. 57–66.

[87] T. Roughgarden and E. Tardos, “How bad is selfish routing?” *J. ACM*, vol. 49, pp. 236–259, March 2002.

[88] J. Feigenbaum, M. Schapira, and S. Shenker, *Algorithmic Game Theory.* Cambridge, UK: Cambridge University Press, 2007, ch. Distributed Algorithmic Mechanism Design.

[89] A. Ozdaglar and R. Srikant, *Algorithmic Game Theory.* Cambridge, UK: Cambridge University Press, 2007, ch. Incentives and Pricing in Communication Networks.

[90] S. Chawla, F. Niu, and T. Roughgarden, “Bertrand competition in networks,” *SIGecom Exch.*, vol. 8, pp. 10:1–10:4, July 2009.

[91] S. Chawla and F. Niu, “The price of anarchy in bertrand games,” in *Proc. of the 10th ACM Conf. on Electron. Commerce*, ser. EC ’09. New York, NY: ACM, 2009, pp. 305–314.

[92] T. G. Griffin, F. B. Shepherd, and G. Wilfong, “The stable paths problem and interdomain routing,” *IEEE/ACM Trans. Netw.*, vol. 10, pp. 232–243, April 2002.

[93] S. J. Luis, “Network routing with path vector protocols: theory and applications,” in *Proc. of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Comput. Commun.*, ser. SIGCOMM ’03. New York, NY: ACM, 2003, pp. 49–60.

[94] H. Wang, H. Xie, Y. Yang, L. Li, Y. Liu, and A. Silberschatz, "Stable egress route selection for interdomain traffic engineering: model and analysis," in *13th IEEE Int. Conf. on Network Protocols, 2005. ICNP 2005.*, Boston, MA, Nov. 2005, pp. 14 – 29.

[95] Y. Yang, H. Xie, H. Wang, A. Silberschatz, A. Krishnamurthy, Y. Liu, and L. E. Li, "On route selection for interdomain traffic engineering," *IEEE Network*, vol. 19, no. 6, pp. 20 – 27, Nov.- Dec. 2005.

[96] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain traffic engineering with BGP," *IEEE Commun. Mag.*, vol. 41, no. 5, pp. 122–128, 2003.

[97] B. Quoitin, S. Tandel, S. Uhlig, and O. Bonaventure, "Interdomain traffic engineering with redistribution communities," *Comput. Commun.*, vol. 27, no. 4, pp. 355–363, 2004.

[98] R. Gao, C. Dovrolis, and E. W. Zegura, "Interdomain ingress traffic engineering through optimized AS-path prepending," in *Proc. of the 4th IFIP-TC6 Int. Conf. on Networking Technologies, Services, and Protocols; Performance of Comput. and Commun. Networks; Mobile and Wireless Commun. Syst.*, ser. NETWORKING'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 647–658.

[99] N. Feamster, J. Borkenhagen, and J. Rexford, "Guidelines for interdomain traffic engineering," *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 19–30, October 2003.

[100] W. Xu and J. Rexford, "MIRO: multi-path interdomain routing," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 171–182, Aug. 2006.

[101] G. Shrimali, A. Akella, and A. Mutapcic, "Cooperative interdomain traffic engineering using nash bargaining and decomposition," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 341–352, April 2010.

[102] M. Roughan and Y. Zhang, "GATEway: symbiotic inter-domain traffic engineering," in *Proc. of the 3rd Int. Conf. on Performance Evaluation Methodologies and Tools*, ser. ValueTools '08. ICST, Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 11:1–11:10.

[103] T. Basar, R. Srikant, and C. M. Simaan, "A stackelberg network game with a large number of followers," *J. of Optimization Theory and Applications*, vol. 115, pp. 479–490, 2002.

[104] T. Basar and R. Srikant, "Revenue-maximizing pricing and capacity expansion in a many-users regime," in *INFOCOM 2002. 21st Annu. Joint Conf. of the IEEE Comput. and Commun. Societies. Proc. IEEE*, vol. 1, New York, NY, 2002, pp. 294 – 301.

[105] D. Acemoglu and A. Ozdaglar, "Competition in parallel-serial networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1180–1192, 2007.

[106] A. Ozdaglar, "Price competition with elastic traffic," *Netw.*, vol. 52, no. 3, pp. 141–155, Oct. 2008.

[107] R. Cole, Y. Dodis, and T. Roughgarden, "Pricing network edges for heterogeneous selfish users," in *Proc. of the 35th Annu. ACM Symp. on Theory of Comput.*, ser. STOC '03. San Diego, CA: ACM, 2003, pp. 521–530.

[108] L. Fleischer, K. Jain, and M. Mahdian, "Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games," in *Proc. of the 45th Annu. IEEE Symp. on Found. of Comput. Sci.*, ser. FOCS '04. Washington, DC: IEEE Computer Society, 2004, pp. 277–285.

[109] X. Huang, A. E. Ozdaglar, and D. Acemoglu, "Efficiency and braess paradox under pricing in general networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 977–991, 2006.

[110] M. Schapira, Y. Zhu, and J. Rexford, "Putting bgp on the right path: a case for next-hop routing," in *Proc. of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets '10. Monterey, California: ACM, 2010, pp. 3:1–3:6.

[111] G. Huston, "Interconnection , peering and settlements," *Proc. of Internet Global Summit INET*, pp. 1–29, 1999.

[112] Y. Xi and E. Yeh, "Pricing, competition, and routing for selfish and strategic nodes in multi-hop relay networks," in *INFOCOM 2008. The 27th Conf. on Comput. Commun. IEEE*, Phoenix, AZ, April 2008, pp. 1463 –1471.

[113] S. Finbow, A. King, G. MacGillivray, and R. Rizzi, "The firefighter problem for graphs of maximum degree three," *Discrete Math.*, vol. 307, no. 16, pp. 2094 – 2105, 2007.

[114] G. Nemhauser and L. Wolsey, *Integer and combinatorial optimization*, ser. Wiley-Interscience series in discrete mathematics and optimization. Hoboken, NJ: Wiley, 1988.

[115] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, pp. 634–652, July 1998.

[116] T. M. R. Ahuja and J. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Upper Saddle River, New Jersey: Prentice Hall, 1993.

[117] C. Hui, "Modeling the diffusion of information in dynamic social and communication networks," Ph.D. dissertation, Decision Sci. and Eng. Syst., Rensselaer Poly. Inst., Troy, NY, 2011.

[118] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[119] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. of Modern Phys.*, vol. 74, pp. 47–97, Jan 2002.

[120] G. Huston, "AS names." [Online]. Available: http://bgp.potaroo.net/cidr/autnums.html [Accessed: 29 Mar. 2012]

[121] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Boston university representative internet topology generator." [Online]. Available: http://www.cs.bu.edu/brite/ [Accessed: 29 Mar. 2012]

[122] P.-A. Chen, M. David, and D. Kempe, "Better vaccination strategies for better people," in *Proc. of the 11th ACM Conf. on Electron. Commerce*, ser. EC '10. New York, NY: ACM, 2010, pp. 179–188.