Methods for Removing Links in A Network to Minimize the Spread of Infections

Apurba K. Nandi, Hugh R. Medal

Industrial and Systems Engineering

Mississippi State University

PO Box 9542

Mississippi State, MS 39762

Tel: (662)325-3923

Abstract

Minimizing the spread of infections is a challenging problem, and it is the subject matter in many different fields such as epidemiology and cyber-security. In this paper, we investigate the effectiveness of link removal as an intervention to minimize the spread of infections in a network. With that in mind, we develop four network interdiction models with different objectives and formulate these models as mixed integer linear programs. We also propose heuristic algorithms to solve the models. In effect, the network interdiction models act as link removal methods when used to minimize the spread of infections. We compare the effectiveness of these four methods with the effectiveness of an existing link removal method, a method based on a link centrality metric, and random link removal. Our results show that probability of transmission is an important factor in determining the effectiveness of link removal methods. Complete isolation of susceptible nodes from infected nodes is the most effective method in reducing the average number of new infections except when the probability of transmission is very low. In contrast, link removal to reduce the high probability transmission paths is the most effective method under most scenarios in increasing the average time to infect half of the susceptible nodes.

Keywords: network interdiction, integer programming, contamination minimization, spread of infections, link removal, edge manipulation

1. Introduction

The spread of harmful infections are omnipresent in real life networks: infectious diseases spread through social and transportation networks, computer viruses and malware spread in com-

Email addresses: akn77@msstate.edu (Apurba K. Nandi), hugh.medal@msstate.edu (Hugh R. Medal)

puter networks, and propaganda and rumors spread in online social networks. Minimizing the spread of these infections is very important because they can cause significant economic and social damage. Infectious diseases cause 10 million deaths each year globally, accounting for 23% of the total disease related deaths [32]. In the well-known influenza pandemic of 1918, 30 to 50 million people globally and more than 600,000 people in the United States are estimated to have died [13]. According to [24], 1000 to 1500 centrifuges of an Iranian nuclear power plant were destroyed by spreading a computer worm called Stuxnet. Lethal worms and viruses such as Stuxnet can easily fall in the hands of terrorist organizations and rogue nations.

It is well-documented that the ability of networks to maintain integrity when subject to attack (selective removal) or error (random removal) on the nodes or links depends on the particular network topology [14, 1]. Removal of a node in breaking the integrity of a network [14, 1] is analogous to immunization in reducing the spread of infections . Holme et al. [14] studied the performance of different types of complex networks after nodes and links were attacked, finding that Erdős–Rényi random networks [9] are the most robust, and scale-free networks [2] are the most vulnerable against attacks. Albert et al. [1] studied the error tolerance of different networks and demonstrated that scale-free networks are the most error tolerant but attack vulnerable. Satorras and Vespignani [25] showed that the error-tolerance property of scale-free networks does not allow a homogeneous approximation of connectivity in infectious disease spread modeling and results in overestimation of the epidemic threshold. Pastor-Satorras and Vespignani [21] even found the absence of an epidemic threshold in scale-free networks. Two implications of the work by Satorras and Vespignani [25] are that even on networks with very small average connectivity, epidemics can occur, and random immunization of nodes might not be an effective intervention. The above findings are practically relevant because many real world networks have the topological characteristics of the scale-free network, random network, and also small-world network [31]. For example, the worldwide air transportation network has the topological characteristics of a scale-free network [11], and some social networks have small-world network properties [10]. The underlying network is indeed an important factor to successfully analyze spread of infections.

It is possible to minimize or slow down the spread of infections in a network by manipulating the topology of the network [16, 28]. Two of the ways of manipulating the topology of a network to minimize spread are removing links and removing nodes. Tong et al. [28] and Kimura et al. [16] proposed greedy algorithms to minimize spread by removing a set of links. Enns et al. [8] proposed a network interdiction model with a non-linear programming formulation that minimizes the number of nodes at risk of infection. Unlike the removal of a set of links in [8, 16, 28], Shen et al. [26], Ventresca and Aleman [29], and Commander [5] proposed network interdiction models that removes a set of nodes to maximize the fragmentation of a network. The network interdiction model proposed by He et al. [12] removes a set of both links and nodes to minimize the total cost

of infection and prevention of infection. Their model can also be used as a node and link removal method to minimize spread by converting the cost of prevention into a budget and minimizing just the cost of infection.

In this paper, we study the problem of selecting a subset of links in a network to minimize the spread of infection. One of the reasons the link removal problem is very important because it is more fundamental than the node removal problem. Links can be selected and removed with finer precision to accomplish the desired objective. Marcelino and Kaiser [18, 19] compared several generic edge and node ranking metrics in reducing the global spread of influenza through the airline network. According to their results, link ranking metrics are usually more effective than node ranking metrics. Moreover, in many situations the node removal option is unattractive or not available at all, whereas there is still a way to remove links. For example, it might be very difficult to find and eliminate terrorists in a terrorist network, but there might be a way to block their communication channels, in effect removing links among them. The loss associated with completely shutting down an entire airport might be enormous, but it might be possible to temporarily suspend the flights between two specific airports during a global disease pandemic. Therefore, the link removal problem has potential applications in fields which were previously studied from the point of view of the node removal problem [4, 31, 3, 27, 17, 23].

Only a small amount of the previous research studied link removal methods that minimize the spread of infections [28, 8, 12, 16, 15, 18, 19]. The algorithm proposed in [28] is based on the finding that the leading eigenvalue of the network adjacency matrix determines whether a spread will turn into an epidemic [30, 22]. They report the effectiveness of their algorithm by showing that it reduces the leading eigenvalue more as compared with other eigenvalues. They also report the effectiveness based on the comparison of fraction of infected nodes produced by a simulation. However, they do not use information about which nodes are infected and which are not in the link selection mechanism. Also, they do not compare their effectiveness with any existing well-known link removal methods and link selection metrics. The algorithm proposed by Kimura et al. [16] is based on bond percolation. The main virtue of the algorithm proposed by Tong et al. [28] and the algorithm along with the speeding mechanism proposed by Kimura et al. [16] is that that they are fast. Kimura et al. [16] evaluates the effectiveness of their algorithm in terms of an indirect measure called contamination degree instead of a direct measure such as the average number of new infections used in this paper. The model proposed by Enns et al. [8] minimizes connectivity between infectious and susceptible nodes, but it is not clear from their work, how effective this model is as a link removal method in minimizing the spread of infections. Moreover, although Enns et al. [8] present a heuristic procedure, it is unclear if their non-linear programming formulation can be solve to optimality for problem instances much larger than the instances with 15 nodes that they solve using complete enumeration. They do not propose any exact procedure

other than complete enumeration.

In this paper, we propose mixed-integer programming formulations of four network interdiction models, each with different interdiction objectives. All the four models assume that only a subset of the links can be removed. The first model (MINCONNECT, sub-section 3.1) minimizes the number of connections between infected and susceptible nodes. The second model (MINATRISK, sub-section 3.2) minimizes the number of susceptible nodes having one or more connections with infected nodes. The third model (MINPATHS, sub-section 3.3) minimizes the number of paths between infected and susceptible nodes. And, the fourth model (MINWPATHS, sub-section 3.4) minimizes the weighted number of paths between infected and susceptible nodes. Our main goal is to minimize the spread of infections. So, after applying an interdiction model, we test the post-interdiction residual network using two different types of simulations: susceptible-infectious (SI) and susceptible-infectious-recovered (SIR). Then, we estimate the average number of new infections (measures the amount of spread) from the SIR simulation and the time to infect half of the susceptible nodes (measures the slowing down of the spread) from the SI simulation. To compare our link removal methods with existing methods, we also evaluate the effectiveness of random link removal, a link removal method proposed by Kimura et al. [16], and a method based on betweenness centrality.

Unlike the methods in [16, 28] which can only be applied to minimize spread, our interdiction models can have applications in fields similar to the ones discussed in [26, 29] to fragment a network. To our knowledge, the MINCONNECT model is the first interdiction model that takes into account connection of each pair of nodes in a link removal setting. This model has potential applications in problems such as synthesizing distributed firewall configurations in a computer network. The MINATRISK model with the same interdiction objective as that of Enns et al. [8] can be solved to optimality for networks larger than 150 nodes within reasonable time. These networks (150 nodes) are much larger than the 15 nodes networks that Enns et al. [8] solved to optimality. In addition, the 150 nodes networks are comparable with the 200 nodes instances that Enns et al. [8] solved using their approximate algorithm. Also, to our knowledge, no existing link removal method is based on controlling the speed of spread. The MINWPATHS model along with the efficient algorithm to solve it is more effective than existing methods in slowing down the spread.

Specifically, the contributions of this paper are as follows: 1) four new mixed-integer programming (MIP) network interdiction models for the infection control problem, 2) two new greedy algorithms for the MIP models, 3) comparison of our link removal methods with existing methods, and 4) recommendations, based on our results about which link removal method is the most appropriate for different infection control contexts (e.g., reduction of the spread versus slowing down of the spread and different probabilities of transmission).

The rest of the paper is organized as follows. In Section 2, we describe the problem of minimizing the spread of infections in a network and the link removal problem. In Section 3, we give mixed-integer programming formulations of our network interdiction models and prove several structural properties. In Section 4, we provide the solution algorithms. In Section 5, we discuss the computational tractability of our algorithms by reporting the results of a set of experiments. In Section 6, we compare all the link removal methods via simulation. Finally, in Section 7, we conclude our paper.

2. Problem Description

Our mathematical models consist of undirected graph $G = (N,A)$, where N is a set of nodes and $A = \{(i, j) : i \in N, j \in N, i < j\}$ is a set of links. Depending on the type of infection, a node might represent a computer, a person, an user account in a social media site, and so on. Similarly, a link might represent communication channels between two computers, social contact between two persons, friendship status between two user accounts, and so on. The models capture the state of a system prior to an outbreak, in which some of the nodes in the network are infected and the rest are susceptible to infection. Let $I \subseteq N$ be the set of infected nodes, and $S = N \backslash I$ be the set of susceptible nodes. The problem is to remove a set of links $L \subseteq A$ to minimize the infection spread in the resulting network, such that the cardinality of L is no greater than some integer parameter b. Parameter b represents the available budget.

In the previous paragraph the objective "minimize the infection spread" was intentionally vague. The spread of infections through a network is inherently stochastic, with a infectious node infecting its neighbors with some probability [16]. However, existing stochastic optimization approaches such as stochastic programming and simulation-optimization are often computationally intensive. Thus, this paper studies four deterministic optimization models, and each act as a proxy for stochastic optimization and as link removal methods to minimize the infection spread. In turn, this paper compares these four models along with some existing methods as measured by a stochastic simulation.

3. Model Formulations

Each of the models optimizes a different network interdiction objective. In terms of the objectives and the formulations, MINCONNECT and MINATRISK models are similar, and MINPATHS and MINWPATHS models are similar. MINCONNECT and MINATRISK models optimize the number of connections. In contrast, MINPATHS and MINWPATHS models[1] optimize the number of

[1] We briefly presented the third and the fourth (MINPATHS and MINWPATHS) models in a conference paper [20]. We describe them in detail in this paper.

transmission paths. Some of the parameters, variables, and constraints are common in the formulations of all the four models. Therefore, we used the same notations for all the common variables in all the formulations to avoid repetition.

We should mention here that any link between any pair of infected nodes is removed before building the corresponding formulations because they both are already infected leaving the link unable to transmit any infection. Since the removal is done before building the formulation, A in the formulation is actually a subset of the original set of links in the network.

The following terms will be used in explaining our four models:

Transmission path: If a path between two nodes (at least one of them is susceptible) contains no other infected nodes, it is a transmission path. If both of the nodes are susceptible, infection can transmit through the path when one of them becomes infected, and no other nodes on the path become infected at the same time.

Connection: Two nodes (at least one of them is susceptible) have one connection if there is at least one transmission path between them.

Susceptible node at risk of infection: A susceptible node is at risk of infection if it has connections with one or more infected nodes.

3.1. MINCONNECT *Model*

The first model, MINCONNECT, seeks to minimize spread by minimizing the number of connections between infected and susceptible nodes.

Let, N_i be the set of neighbors of node i, and $\Omega = \{(i,j) : (i,j) \in N \times N, j > i, (i,j) \notin I \times I\}$. Ω is the set of distinct pairs of nodes, and at least one of the nodes in each pair is susceptible. The decision variables are as follows.

$$x_{ij} = \begin{cases} 1 & \text{if } i\text{is connected to } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if link } (i,j)\text{is removed} \\ 0 & \text{otherwise} \end{cases}$$

The MINCONNECT model is formulated as follows.

$$\text{(MinConnect)} \quad \min \quad \sum_{i \in S} \sum_{j \in I} x_{ij} \tag{1a}$$

$$\text{s.t.} \quad x_{ij} + y_{ij} \geq 1 \quad \forall (i,j) \in A \tag{1b}$$

$$x_{ki} - x_{kj} + y_{ji} \geq 0 \quad \forall (k,i) \in \Omega, \quad \forall j \in N_i, k \neq j, j \notin I \tag{1c}$$

$$\sum_{(i,j) \in A} y_{ij} \leq b \tag{1d}$$

$$x_{ij} \geq 0 \quad \forall (i,j) \in \Omega \tag{1e}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A \tag{1f}$$

The objective function (1a) counts total number of connections between all the pairs of infected and susceptible nodes. Constraint (1b) makes sure that two neighboring nodes i and j are connected ($x_{ij} = 1$) if the link (i,j) between them is not removed. Constraint (1c) makes sure that a node k is connected to node i if node k is connected to node j and the link (i,j) is not removed. Constraints (1b) and (1c) together with the objective function (1a) ensure that $x_{ij} = 1$ if and only if there exists at least one path of unremoved links between i and j. There is no constraint (1c) for nodes i and k if both of them are infected. There is no constraint (1c) for nodes i and k through node j if node j is infected. Moreover, i and k might be neighbors, but constraint (1c) checks their connectivity through other neighbors. Constraint (1d) is the budget constraint. Note that x_{ij} variables are defined as binary. However, they are not required to be binary in the above formulation. Corollary 1 shows that x_{ij} variables will always have binary solutions.

Lemma 1. *An implied lower bound on x_{ij} is either 0 or 1 for all $(i,j) \in \Omega$ in any feasible solution of the* MINCONNECT *model.*

Proof. The proof is divided into three cases. Cases 1 and 2 can exist simultaneously for a pair of nodes (i,j), whereas case 3 exists in the absence of cases 1 and 2. □

Case 1. Nodes i and j have a single link transmission path. Constraint (1b) ensures that $x_{ij} \geq 0$ if $y_{ij} = 1$ and $x_{ij} \geq 1$ if $y_{ij} = 0$.

Case 2. Nodes i and j have one or more multiple link transmission paths. If $y_{pq} = 0$ where $(p,q) \in A$ for all of the links on a transmission path, then $x_{pq} \geq 1$ for all the links on that transmission path. From constraint (1c), $x_{ij} \geq 1$. If $y_{pq} = 1$ for at least one link on the transmission path, from constraint (1c), $x_{ij} \geq 0$. In this way, either $x_{ij} \geq 0$ or $x_{ij} \geq 1$ resulting from that transmission path. The same is true for all the transmission paths between nodes i and j.

Case 3. If there is no transmission path between nodes i and j, constraint (1e) ensures that $x_{ij} \geq 0$ for all $(i, j) \in \Omega$.

Considering all the three cases simultaneously, it is clear that the lower bound on x_{ij} is either 0 or 1 for all $(i, j) \in \Omega$ in any feasible solution of the MINCONNECT model.

Corollary 1. *An optimal solution to* MINCONNECT *exists such that* $x_{ij} \in \{0, 1\}$ for all $(i, j) \in \Omega$.

Proof. From Lemma 1, it is obvious that $x_{ij} \in \{0, 1\}$ for all $(i, j) \in \Omega$ in any optimal solution of the MINCONNECT model because it is a minimization problem with an objective function of the sum of x_{ij} variables. □

3.2. MINATRISK *Model*

The second model, MINATRISK, seeks to minimize spread by minimizing the number of susceptible nodes at risk of infection. The objective of this model is actually the same as the objective in [8]. However, the formulation in [8] is a non-linear program, while we present a mixed-integer linear programming formulation below.

All the parameters of this formulation are the same as the parameters in the formulation of the MINCONNECT model. And, the additional decision variable z_i is defined below.

$$z_i = \begin{cases} 1 & \text{if susceptible node } i\text{is at risk of infection} \\ 0 & \text{otherwise} \end{cases}$$

The MINATRISK model is formulated as follows.

$$\text{(MinAtRisk)} \quad \min \quad \sum_{i \in S} z_i \tag{2a}$$

$$\text{s.t.} \quad \text{(1b)} - -\text{(1f)} \tag{2b}$$

$$z_i - x_{ij} \geq 0 \quad \forall i \in S, j \in I \tag{2c}$$

The objective function (2a) counts the number of susceptible nodes at risk of infection. Constraint (2b) in this model includes the constraints (1b), (1c), (1d), (1e) and (1f) in the MINCONNECT model and serves the same purpose. The additional constraint (2c) makes sure that a susceptible node is at risk of infection if it is connected to one or more of the infected nodes. Note that although z_i for all $i \in S$ variables are defined as binary, they are not required to be binary in the above formulation. Lemma 2 proves that z_i variables will always have binary solutions.

Lemma 2. *An optimal solution to* MINATRISK *exists such that* $z_i \in \{0, 1\}$ for all $i \in S$.

Proof. In the MINATRISK formulation, $x_{ij} \geq 0$, or $x_{ij} \geq 1$ for all $(i, j) \in \Omega$ from Constraint (2b) based on Lemma 1. Now, from constraint (2c), $z_i \geq 0$ or $z_i \geq 1$ for all $i \in S$. Hence, $z_i \in \{0, 1\}$ for all $i \in S$ in the optimal solution since MINATRISK is a minimization problem with an objective function of the sum of z_i variables. □

Let, MINATRISK-Z be a variation of the MINATRISK formulation such that the binary constraint $y_{ij} \in \{0,1\}$ for all $(i,j) \in A$ in (2b) is replaced with $0 \leq y_{ij} \leq 1$ for all (i,j) in A, and the constraint $z_i \in \{0,1\}$ for all $i \in S$ is added. Lemma 3 shows that any optimal fractional y_{ij} solution of the MINATRISK-Z formulation can be converted to a binary solution by simply setting the fractional y_{ij} values to 0 without altering the optimal objective function value. This finding is particularly important because MINATRISK-Z is computationally much more efficient than the MINATRISK formulation (see Section 5).

Lemma 3. *Any optimal solution of the* MINATRISK-Z *formulation which has some fractional* y_{ij} *values can be converted to* $y_{ij} \in \{0,1\}$ *for all* $(i,j) \in A$ *by setting the fractional* y_{ij} *values to* 0 *without altering the optimal objective function value, and the optimal objective function value is equal to the optimal objective function value of the* MINATRISK *formulation.*

Proof. The proof is divided into two cases as follows. □

Case 1. Let us first examine the solution of the MINATRISK formulation if the binary constraint $y_{ij} \in \{0,1\}$ for all $(i,j) \in A$ is removed. Suppose, the set of all the susceptible nodes is divided into two sets $S1$ and $S2$. Nodes in $S1$ is connected to nodes in I through a set of links $L1$ where $|L1| > b$. And, nodes in $S2$ is connected to nodes in I through a set of links $L2$ where $|L2| = b$. From constraints (2b) and (2c), it is possible to have a solution such that $0 \leq y_{ij} = \frac{b}{|L1|} < 1$ for all $(i,j) \in L1$ and $y_{ij} = 0$ for all $(i,j) \in L2$ where $\sum_{(i,j)\in L1} y_{ij} = b$ if $|S1| \gg |S2|$. This is due to the reduction in the objective function $|S1| * \frac{b}{|L1|} > |S2|$. The quantity in the right side of the inequality is the reduction in the objective function if the solution is $y_{ij} = 1$ for all $(i,j) \in L2$ and $y_{ij} = 0$ for all $(i,j) \in L1$. So, the fractional solution is superior than the binary solution (Figure 1b). However, for the MINATRISK-Z formulation and for the same fractional solution, the reduction in the objective function is $0 < |S2|$ because $z_i = 1$ in any solution in which $z_i > 0$ from constraint $z_i \in \{0,1\}$ for all $i \in S$. So, the binary solution is superior than the fractional solution (Figure 1a). For the MINATRISK formulation, the binary solution is automatically selected.

Case 2. If $|L2| < b < |L1|$, and it requires more than $b - |L2|$ links to be removed to save one more node, the extra budget $b - |L2|$ cannot be used to save any more nodes after saving the nodes in $S2$. So, in the optimal solution if $0 < y_{ij} < 1$ for any $(i,j) \in L1$, they can be set to 0 without altering the objective function. Also, if $b > |L|$ in the trivial case, where L is the set of all the links connected to the infected nodes, the extra budget $b - |L|$ cannot be used because there is no remaining susceptible nodes to be saved (Figure 1c). So, in the optimal solution if $0 < y_{ij} < 1$ for any $(i,j) \in A \backslash L$, they can be set to 0 without altering the objective function. For the MINATRISK formulation, in

both $|L2| < b < |L1|$ and $b > |L|$ situations, links corresponding to the extra budget is automatically set to either 0 or 1.

In both cases, the optimal solution after conversion will be $y_{ij} \in \{0,1\}$ for all $(i,j) \in A$, and the optimal solution of the MINATRISK formulation is equal to the optimal solution of the MINATRISK-Z formulation.

3.3. MINPATHS *Model*

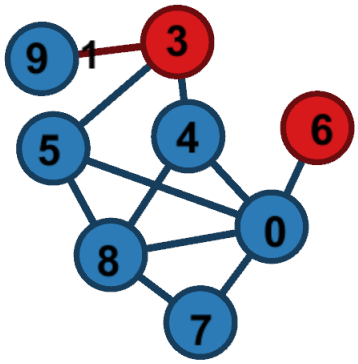
The objective of the MINPATHS model is to maximize the number of transmission paths removed from the network.

Let, P_{uv} be the set of transmission paths between infected node u and susceptible node v, and L_{uvw} be the set of links belonging to the w^{th} transmission path in P_{uv}. We use a modified depth first search algorithm to find all the L_{uvw}s. The additional decision variable used in this model is as follows.

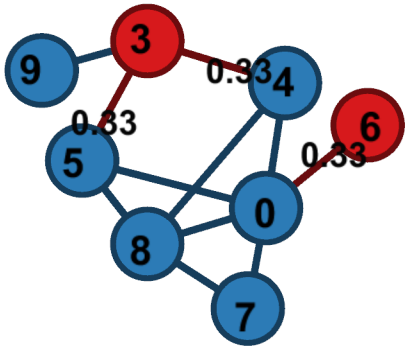
$$t_{uvw} = \begin{cases} 1 & \text{if the transmission path } w \text{in } P_{uv} \text{is removed} \\ 0 & \text{otherwise} \end{cases}$$

The MINPATHS model is formulated as follows.

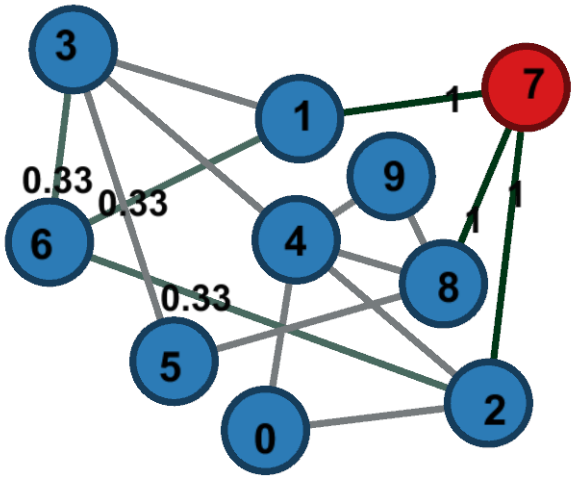
$$\begin{aligned}
\text{(MinPaths)} \quad \max \quad & \sum_{u \in I} \sum_{v \in S} \sum_{w \in P_{uv}} t_{uvw} && \text{(3a)} \\
\text{s.t.} \quad & t_{uvw} - \sum_{(i,j) \in L_{uvw}} y_{ij} \leq 0 \quad \forall u \in I, v \in S, w \in P_{uv} && \text{(3b)} \\
& t_{uvw} \leq 1 \quad \forall u \in I, v \in S, w \in P_{uv} && \text{(3c)} \\
& \sum_{(i,j) \in A} y_{ij} \leq b && \text{(3d)} \\
& y_{ij} \in \{0,1\} \quad \forall (i,j) \in A && \text{(3e)}
\end{aligned}$$



(a) Solution to MINATRISK-Z ($b = 1$). Objective function value = 5.



(b) Solution to MINATRISK-Z without the binary constraint ($b = 1$). Objective function value = 4.35.



(c) Solution to MINATRISK-Z ($b = 4$). Objective function value = 0.

Figure 1: Example solutions of the MINATRISK-Z formulation without and with the binary constraint. Figures 1a and 1b are for the same problem, and figure 1c is for a different problem. Red and blue nodes are infected and susceptible, respectively. Links with fractional values on them are partially removed; links without any value are not removed; and links with value 1 are completely removed.

Objective function (3a) counts the number of transmission paths removed from the network. Constraints (3d) and (3e) are the same as the constraints (1d) and (1f) respectively in the MIN-CONNECT model. Constraint (3c) makes sure that if none of the links on the w^{th} transmission path between infected node u and susceptible node v is removed, the transmission path is not removed. Constraint (3e) guarantees that the values of the path removal variables do not exceed 1. Note that although we do not use t_{uvw}s (path removal variables) as binary variables, the formulation guarantees binary values of this variable in any feasible solution of the model. This satisfies our definition of transmission paths.

There are potentially exponential number of paths between all the infected nodes and all the

susceptible nodes in a network. It means that the MINPATHS and the MINWPATHS models can only be built and solved for relatively small networks (less than 20 nodes). Thus, we propose greedy algorithms to solve them and demonstrate the performance of the algorithms in the computational experiments in section 5. Algorithm 2 solves the MINPATHS model. Algorithm for the MINWPATHS model can be found by simple modification of algorithm 2.

3.4. MINWPATHS *Model*

The objective of this model is to minimize the weighted number of transmission paths between all the infected nodes and all the susceptible nodes. The weight of a transmission path is the product of the transmission probabilities on each of the links on that path. Note that the MINWPATHS and the MINPATHS models are the same except their objective functions.

Let, p be the probability of an infected node transmitting infection to a neighboring susceptible node, and then p_{uvw} be the probability of transmission from infected node u to susceptible node v on the w^{th} transmission path between them. $p_{uvw} = p^{|L_{uvw}|}$. The MINWPATHS model is formulated as follows.

$$\text{(MinWPaths)} \quad \max \sum_{u \in I} \sum_{v \in S} \sum_{w \in P_{uv}} p_{uvw} t_{uvw} \tag{4a}$$

$$\text{(3b)–(3e)} \tag{4b}$$

Objective function (4a) ascertains the total number of weighted paths between all the infected and susceptible nodes. The constraint (4b) in this model includes the constraints (3b), (3c), (3d), and (3e) in the MINPATHS model and serves the same purpose.

4. Solution Algorithms

At first, we solve some of the problem instances of the optimization models using the commercial solver CPLEX [7] to understand the need for developing any algorithm. It is clear that our models cannot be solved for large problems (more than 300 nodes - average node degree 4) by a commercial solver within reasonable time (two hours). Motivated by the previous studies that successfully develop computationally efficient algorithms using Benders decomposition [6], we also develop and test algorithm based on these decomposition technique. However, Benders decomposition of the MINATRISK formulation is slower than its direct solution using CPLEX (Tables 1). Therefore, we developed Monte Carlo-based greedy algorithms for all the four models. In this section, we present the greedy algorithms, and in Section 5, we present and discuss the performances of both the greedy and decomposition algorithms.

4.1. Greedy Algorithm for the MINCONNECT *Model*

Algorithm 1 is the greedy algorithm for the MINCONNECT model. There are three nested loops in the algorithm. At every iteration of the most outer loop **(while)**, one of the links among all the remaining links in the network having the highest potential to minimize the number of connections between the infected and susceptible nodes is removed from the network. The potential of a link to minimize the number of connections is estimated by temporarily removing this link along with some other links to fill out the budgeted quota of links that can be removed and then, counting the number of connections between the infected and susceptible nodes in the resulting network.

Note that M is a parameter used in all the algorithms. It controls the number of times a set of links is randomly removed in the MINCONNECT and the MINATRISK models and the number of trees generated randomly from each of the infected nodes at the beginning in the algorithms for the MINPATHS and the MINWPATHS models. In other words, M is the number of replications in the random selection processes of the greedy algorithms.

Algorithm 1 Select a set of links to remove from a network to minimize the number of connections between infected and susceptible nodes

procedure GREEDY-MINCONNECT
  A network $G := (N, A)$
  A set of infected nodes $:= I$
  A budget $:= b$
  Number of samples $:= M$
  $N := N^0$, $A := A^0$
  A set of susceptible nodes, $S := N \setminus I$
  A set of links, $L := \emptyset$
  while $|L| < b$ **do**
    for $i := 1, |A|$ **do**
      $A := A \setminus A_i$
      $TC_{A_i} := 0$
      for $j := 1, M$ **do**
        $TL :=$ $b - |L| - 1$ randomly selected links$\in A$
        $A := A \setminus TL$
        $C_{A_i} :=$ Remaining connections between nodes$\in I$ and nodes$\in S$
        $TC_{A_i} := TC_{A_i} + C_{A_i}$
        $A := A \cup TL$
      end for
      $A := A \cup A_i$
      if $TC_{A_i} < TC_{best}$ **then**
        $i_{best} := i$
      end if
    end for
    $A := A \setminus A_{i_{best}}$
    $L := L \cup A_{i_{best}}$
  end while
  return L
end procedure

4.2. Greedy Algorithm for the MINATRISK *Model*

We do not present the greedy algorithm for the MINATRISK model separately because this algorithm and the greedy algorithm for the MINCONNECT model are very similar. The only difference is that i_{best} at any iteration of the **while** loop is selected based on TR_{A_i} rather than TC_{A_i} where TR_{A_i} is the total remaining number of susceptible nodes at risk of infection when link A_i is evaluated.

4.3. Greedy Algorithm for the MINPATHS *Model*

Algorithm 2 is the algorithm for the MINPATHS model. In the MINPATHS algorithm, after generating M trees, at each iteration of the main loop **(while)**, the link that removes the maximum

number of paths between the infected and susceptible nodes is removed from the network. At an iteration, the number of paths removed by a link is calculated by temporarily removing the link from the network and counting the number of paths removed from the remaining paths as a result of removing the link under consideration.

4.4. Greedy Algorithm for the MINWPATHS *Model*

We do not present the greedy algorithm for the MINWPATHS model separately, because this algorithm is similar to the one for the MINPATHS model. The difference is that i_{best} at any iteration of the **while** loop is selected based on WP_{A_i} rather than $P_{A_i}$where WP_{A_i} is the number of weighted paths removed from the total number of weighted paths when the link A_i is evaluated.

Algorithm 2 Select a set of links to remove from a network to minimize the number of paths between infected and susceptible nodes

procedure GREEDY-MINPATHS
  A network $G := (N, A)$
  A set of infected nodes $:= I$
  A budget $:= b$
  Number of trees $:= M$
  $N := N^0$, $A := A^0$
  A set of susceptible nodes, $S := N \setminus I$
  A set of links, $L := \emptyset$
  Randomly generate M trees of G from nodes in I to nodes in S
  $P :=$ all the paths between infected and susceptible nodes in M trees
  while $|L| < b$ **do**
    for $i := 1, |A|$ **do**
      $A := A \setminus A_i$
      $P_{A_i} :=$ number of paths removed from P
      if $P_{A_i} > P_{best}$ **then**
        $i_{best} := i$
      end if
      $A := A \cup A_i$
    end for
    $A := A \setminus A_{i_{best}}$
    $L := L \cup A_{i_{best}}$
  end while
  return L
end procedure

5. Computational Experiments

In this section, we investigate the computation times of the models for several different network sizes and problem configurations. Network size here means the number of nodes, and problem

configuration means a specific combination of the fraction of nodes infected and fraction of links removed. Reader should assume Random network [9] as the underlying network type throughout this paper unless some other network type is explicitly mentioned. The other network type used is the Scale-free network [2]. Reader also should not confuse Random networks with randomly generated networks. All the networks (both Random and Scale-free) used in this paper are randomly generated and have an average node degree of 4. Thus, the number of links in any of the networks is double of the number of nodes with very minor variation. However, two networks with the same network size might have different number of links between infected nodes. Therefore, the number of links might have minor variation even for the same network size when they are input into the optimization models because of the removal of links between infected nodes before inputting into the models. All the experiments were carried out on a computer with an Intel core i7 2.90GHz processor and 8GB RAM.

At first, we report the computation times of direct solutions of the MINCONNECT and the MINATRISK models by CPLEX and the computation times of the decomposition technique for the MINATRISK model. We do not report the computation times of the MINPATHS and the MINWPATHS models by CPLEX as they can be solved for only small networks. Then, we evaluate the performance of the greedy algorithms in terms of 1) the proportion of their solutions that are optimal and 2) the average optimality gap. We also report the computation times required by the greedy algorithms to solve problems up to 200 nodes.

5.1. MINCONNECT *Model*

We found that the computation time for solving this model using CPLEX varies a great deal even for problem instances with the same number of nodes and budget (b). This is infact true for both the MINCONNECT and the MINATRISK models. Thus, we conclude that problem instances become easy or difficult to solve depending on the relative positions of the infected and susceptible nodes in the network. Figure 2 shows the variation of the average computation time with respect to the network size for different problem configurations. For each parameter combination, the average computation time is taken over ten randomly-generated problem instances.

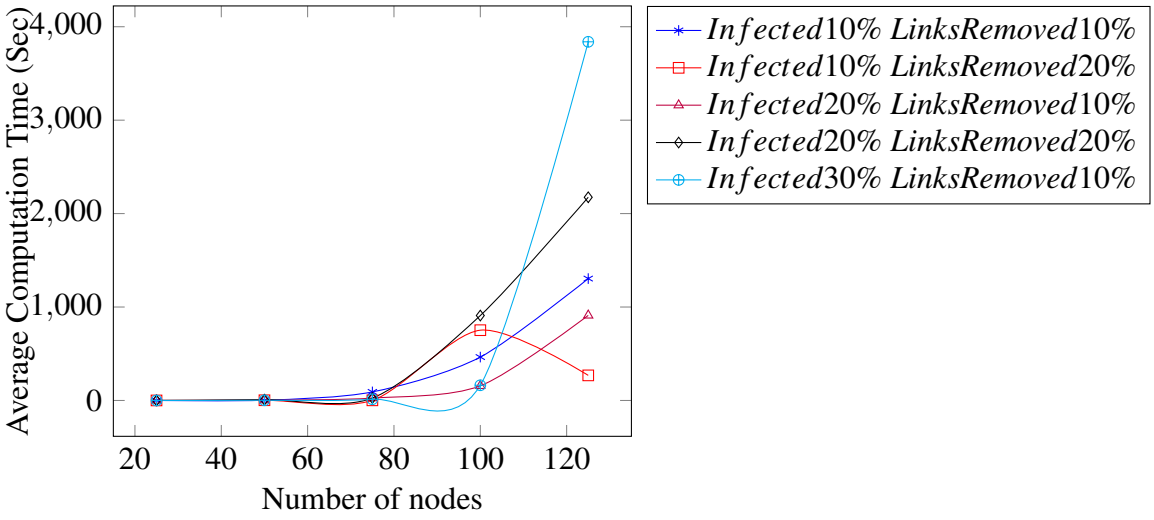


Figure 2: Variation of average computation time of the MINCONNECT model with respect to network size for different combinations of fraction of nodes infected and fraction of Links removed

5.2. MINATRISK *Model*

Recall that the MINATRISK model and the model in [8] have the same network interdiction objective. However, unlike the non-linear programming formulation in [8], we formulate the problem as a mixed-integer linear program which can be solved by a commercial solver such as CPLEX for relatively large problems. In [8], authors present an approximate algorithm that can solve problem instances with 200 nodes in about 2 hours. The equivalent MILP formulation proposed in this paper, especially the MINATRISK-Z formulation can be solved to optimality by CPLEX for problems with more than 150 nodes in less than 2 hours.

Figure 3 is constructed by plotting the average computation time of 60 problem instances of different configurations for each network size for both the MINATRISK-Z and the MINATRISK formulations. Figure 3 shows that average computation time of the MINATRISK-Z formulation is less than that of the MINATRISK formulations, and also the former increases slower than the latter as the number of nodes increases. In fact, computation times for the MINATRISK-Z formulation are always less than the corresponding computation times for the MINATRISK formulation.

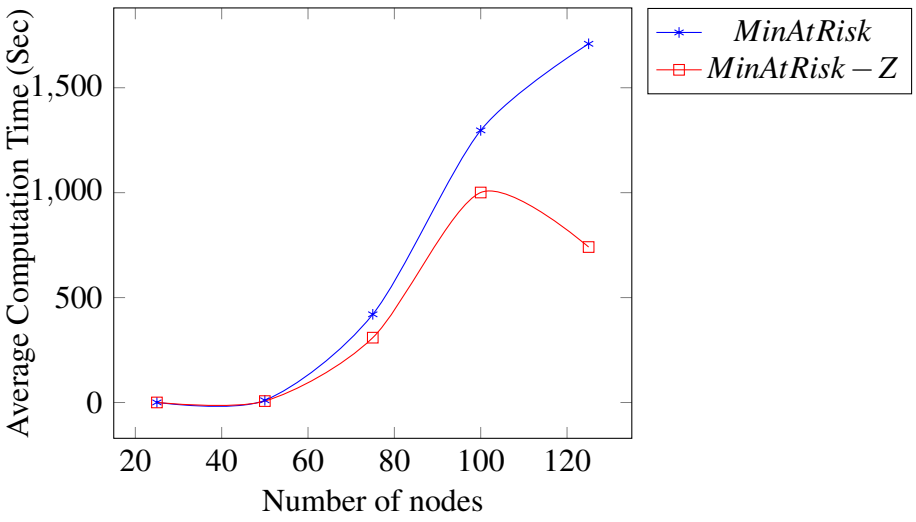


Figure 3: Variation of average computation time of the MINATRISK and the MINATRISK-Z formulations

Figure 4 demonstrates an interesting behavior of the computation times of the MINATRISK-Z formulation for different problem configurations. Apparently, the computation time varies significantly with the ratio between the fraction of nodes infected and the fraction of links removed. The average computation time decreases as the ratio becomes smaller or larger than one. Recall that there are approximately twice as many links as there are nodes in each of our experimental networks because of average node degree being 4. So, equal fraction of infected nodes and links to be removed means that on an average, the number of links that can be removed is half of the number of links connected to the infected nodes. This is likely to increase the number of feasible solutions, making the problem combinatorially more difficult.

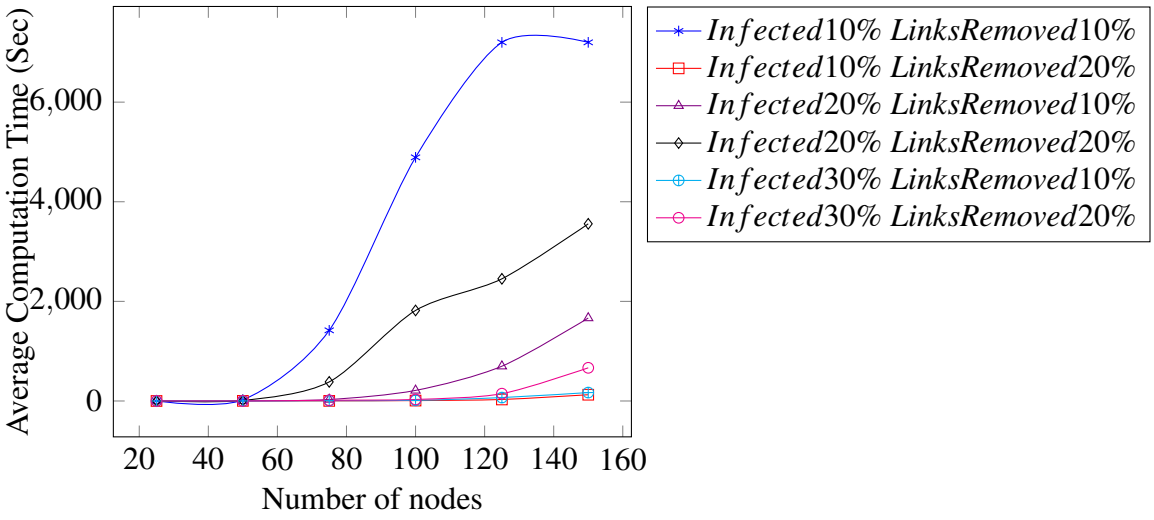


Figure 4: Variation of average computation time of the MINATRISK-Z model with respect to network size for different combinations of fraction of nodes infected and fraction of links removed

5.2.1. Benders Decomposition of the MINATRISK *Formulation*

Table 1 presents the computation times of 20 problem instances of 25 nodes. According to Table 1, computation times using Benders decomposition, $TotalTime = SubTime + MasterTime$ for the MINATRISK formulation are much greater than the corresponding computation times using CPLEX (MinAtRiskTime, MinAtRisk-ZTime). The problem is that even for relatively small problems such as those with 25 nodes, the algorithm requires a significantly large number of Benders optimality cuts (Cuts) to find the optimal solution. The consequence is that the algorithm solves the master problem to optimality many times. So, the proportion of the total time for solving the master problem is dominant in the total time for solving a problem using Benders decomposition. Formulation of the Benders decomposition master problem is given below. In the formulation of the master problem (5), $q(y)$ represents the sub-problem which calculates the number of susceptible nodes at risk of infection given a set of removed links. Refer to the supplemental material for a complete description of the formulation and algorithm.

$$
\begin{aligned}
MP = \quad & \min \quad q(y) && \text{(5a)}\\
& \text{s.t.} \quad \sum_{(i,j)\in A} y_{ij} \leq b && \text{(5b)}\\
& \qquad y_{ij} \in \{0,1\} \quad \forall (i,j) \in A && \text{(5c)}
\end{aligned}
$$

No.	Node	Arc	Infected	LinkRemove	Cuts	SubTime	MasterTime	TotalTime	MinAtRiskTime	MinAtRisk-ZTime
1	25	35	0.1	0.1	4	0	0	0	1	0
2	25	47	0.2	0.1	68	1	10	11	0	1
3	25	52	0.1	0.1	126	0	33	33	1	1
4	25	35	0.2	0.1	23	0	3	3	0	0
5	25	43	0.1	0.1	61	0	8	8	0	1
6	25	54	0.2	0.1	178	2	33	35	0	0
7	25	50	0.1	0.1	244	1	161	162	3	0
8	25	43	0.2	0.1	36	0	4	4	0	0
9	25	47	0.1	0.1	33	1	2	3	0	0
10	25	45	0.2	0.1	53	1	5	6	1	0
11	25	37	0.1	0.1	4	0	0	0	1	0
12	25	49	0.2	0.1	82	1	10	11	0	0
13	25	45	0.1	0.1	23	0	3	3	0	0
14	25	37	0.2	0.1	12	0	1	1	0	0
15	25	56	0.1	0.1	134	1	41	42	1	1
16	25	50	0.2	0.1	340	7	224	231	1	0
17	25	51	0.1	0.1	196	1	67	68	2	1
18	25	47	0.2	0.1	150	2	25	27	1	0
19	25	39	0.1	0.1	26	0	3	3	1	0
20	25	37	0.2	0.1	41	0	5	5	0	0

Table 1: Computation times (seconds) using benders decomposition (classical implementation) versus direct solution using CPLEX.

5.3. Greedy Algorithms

To evaluate the performance of the greedy algorithms, we randomly generated 100 Random networks, each with 12 nodes, for several problem configurations. Recall that the MINPATHS and MINWPATHS models can only be solved by CPLEX for small networks. Therefore, to keep the same basis of comparison for all the algorithms, we use a network of 12 nodes for performance evaluation. The values of M (number of replications) are set to 100 in all of the greedy algorithms.

The dash bordered columns in figure 5 show the fraction of times the algorithms produced optimal solutions, and the solid bordered columns show the average gap between the solutions generated by the greedy algorithms and the optimal solutions found using CPLEX. According to figure 5, more than 60% and 70% solutions for the MINCONNECT and the MINATRISK models, respectively, solutions were optimal. On the other hand, even though the MINPATHS and MINWPATHS algorithms found optimal solutions only about 15% and 23% of the times respectively, their average optimality gaps were both less than 5%.

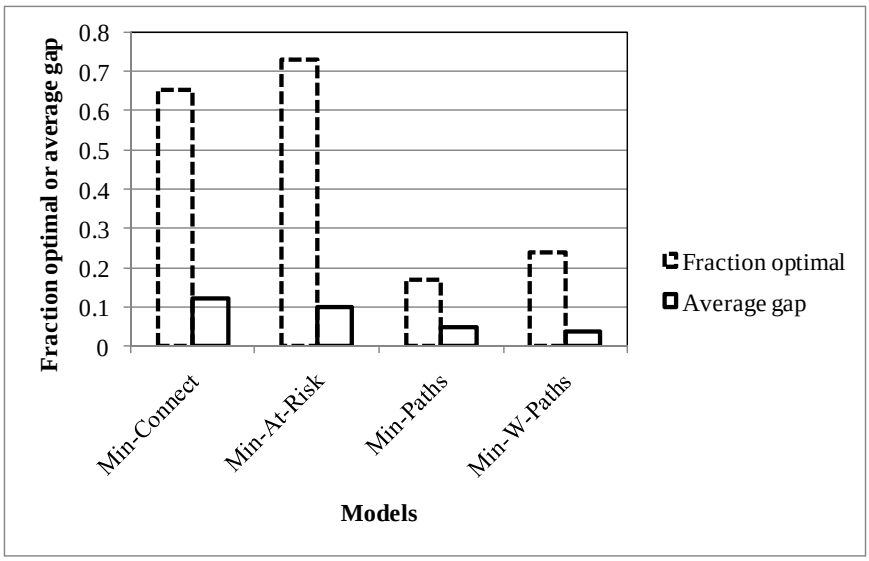


Figure 5: Fraction of optimal and average gap of the greedy algorithms with respect to CPLEX solutions

Figure 6 presents average run times of the greedy algorithms for different network sizes. To create this plot, we used 30 randomly generated random networks, 10 networks for each of the three network sizes: 100, 150, and 200 nodes. The fraction of nodes infected, fraction of links that can be removed, and transmission probability is equal to 0.2, 0.2, and 0.15 respectively for all the problems. Figure 6 shows that on an average, the greedy algorithm for the MINATRISK model takes about 600 seconds to solve problems with networks of 200 nodes, and this greedy algorithm takes the longest time. In fact, the greedy algorithms for the MINPATHS and the MINWPATHS models solve problems with networks of 200 nodes within a few seconds.

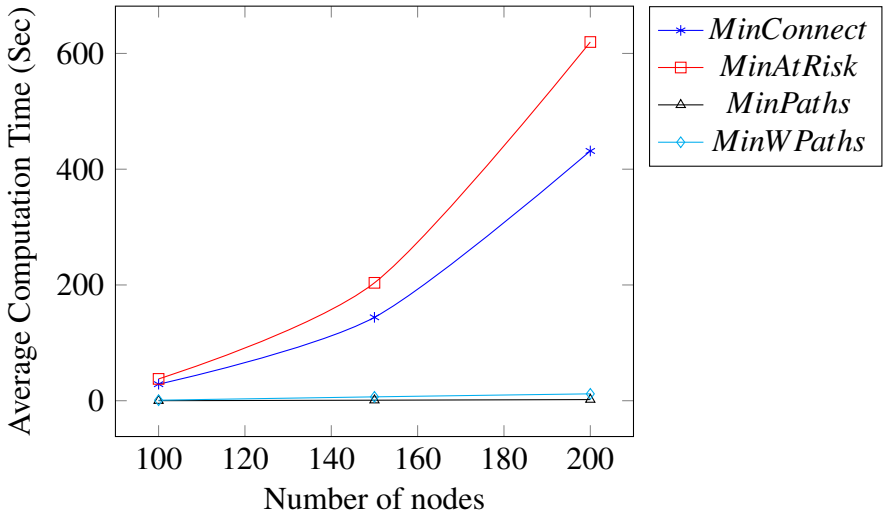


Figure 6: Variation of average computation times with respect to network size using the greedy algorithms for all the four models

6. Comparison of Link Removal Methods in Minimizing Spread

Recall that our original objective is to minimize or to slow down the spread, and the network interdiction models have connectivity-related objectives, which serve as proxies for reducing spread. Therefore, to evaluate the effectiveness of these models and their associated greedy algorithms as link removal methods in achieving the original objective, we compare them with three existing methods using simulation. Thus, the methods evaluated in this section are:

1. Optimal solution of MINCONNECT / Greedy algorithm solution of MINCONNECT.
2. Optimal solution of MINATRISK / Greedy algorithm solution of MINATRISK.
3. Optimal solution of MINPATHS / Greedy algorithm solution of MINPATHS.
4. Optimal solution of MINWPATHS / Greedy algorithm solution of MINWPATHS.
5. **RANDDEL.** In this link removal method, a random set, $L \subseteq A$ of the links is removed from the network, and a simulation is run on the residual network. The performance of this method is evaluated as the average simulated performance over M replications.
6. **GREEDYDEL** [16]. In this method, links are iteratively removed from the network using a metric called the minimum average contamination degree. A link to be evaluated in the remaining network is temporarily removed from the network. Next, one random number is generated for each of the links in the network, and the links having corresponding random numbers greater than the probability of transmission are temporarily removed from the network. Then, the contamination of the link under evaluation is calculated as the total number of susceptible nodes at risk of infection in the remaining network. The above two steps are carried out M times, and a total of M contamination degrees are estimated. Average contamination degree is then calculated as the average of the M contamination degrees. Average contamination degree is calculated for all the other links, and the link with the minimum average contamination degree is removed from the network. The algorithm proceeds to the next iteration and recalculates the average contamination degrees of the remaining links.
7. **BETWEENDEL.** Betweenness centrality score ($c(a)$) of a link for this greedy algorithm is calculated using the formula below. Although, Betweenness centrality score is a standard metric for evaluating the importance of a link, the following formula is adapted from the standard formula by Enns et al. [8]. The change is that the nodes pair (i, j) is composed of an infectious and a susceptible node rather than (i, j) being any pair of nodes.

$$c(a) = \sum_{(i,j) \in I \times S} \frac{\phi(i,j|a)}{\phi(i,j)} \tag{6}$$

Here, $\phi(i, j|a)$ is the number of shortest paths between infected node i and susceptible node j on which link a is one of the links. $\phi(i, j)$ is the total number of shortest paths between node i and node j. Then, the greedy algorithm works as follows. At each iteration, the link among the remaining links having the maximum centrality score is removed from the network. Then, similar to the GREEDYDEL algorithm, this algorithm also proceeds to the next iteration and recalculates the centrality score of all the remaining links.

To evaluate one of these seven methods on a particular problem instance, we do the following: 1) generate the network, 2) obtain a solution from the method, 3) remove the links from the network that are prescribed by the solution, and then 4) run a simulation of spread on the residual network.

We use discrete time stochastic SI and SIR simulations to compare the performances of all the link removal methods. After running a simulation, performance metrics are estimated: 1) Expected number of new infections ($E(I_{new})$) and 2) Expected time to infect half of the susceptible nodes ($E(T)$). $E(I_{new})$ is the average number of new infections, and it is estimated from the SIR simulation. $E(T)$ is the average time to infect half of the susceptible nodes at the beginning of each simulation run, and it is estimated from the SI simulation. $E(I_{new})$ and $E(T)$ resemble the amount and speed of the spread of infections, respectively. Average new infections is estimated from the SIR simulation because it is always possible to estimate average new infections from this simulation model whether the value is low or high, and the SIR simulation is also realistic enough to be applicable in most of the spreading scenarios. In contrast, there might be a few or no new infections if the nodes recover making it difficult to estimate the time to infect half of the susceptible nodes. Hence, this metric is esimated from the SI simulation.

In the SI simulation, at each iteration (tick), all the infected nodes infect their susceptible neighbors if the corresponding random numbers generated for the links are less than the probability of transmission. In this simulation, infected nodes do not recover from infection. Infection spreads in the same way in the SIR simulation also. However, there is a fixed probability of recovery for the infected nodes. If the random number generated corresponding to an infected node is less than the probability of recovery, that infected node recovers from infection and becomes immune from further infection.

6.1. Experimental Setup

A total 1000 simulation replications were carried out to estimate the performance metrics for each of the different combinations of the following factors: probability of transmission, fraction of nodes infected, and fraction of links that can be removed. Table 2 shows the different values of factors for which simulations were run:

Factor	Values
Transmission probability	0.05, 0.15, 0.25, 0.9
Fraction infected	0.1, 0.2, 0.3
Fraction of links removed	0.1, 0.2
Network size (nodes)	12, 50, 150

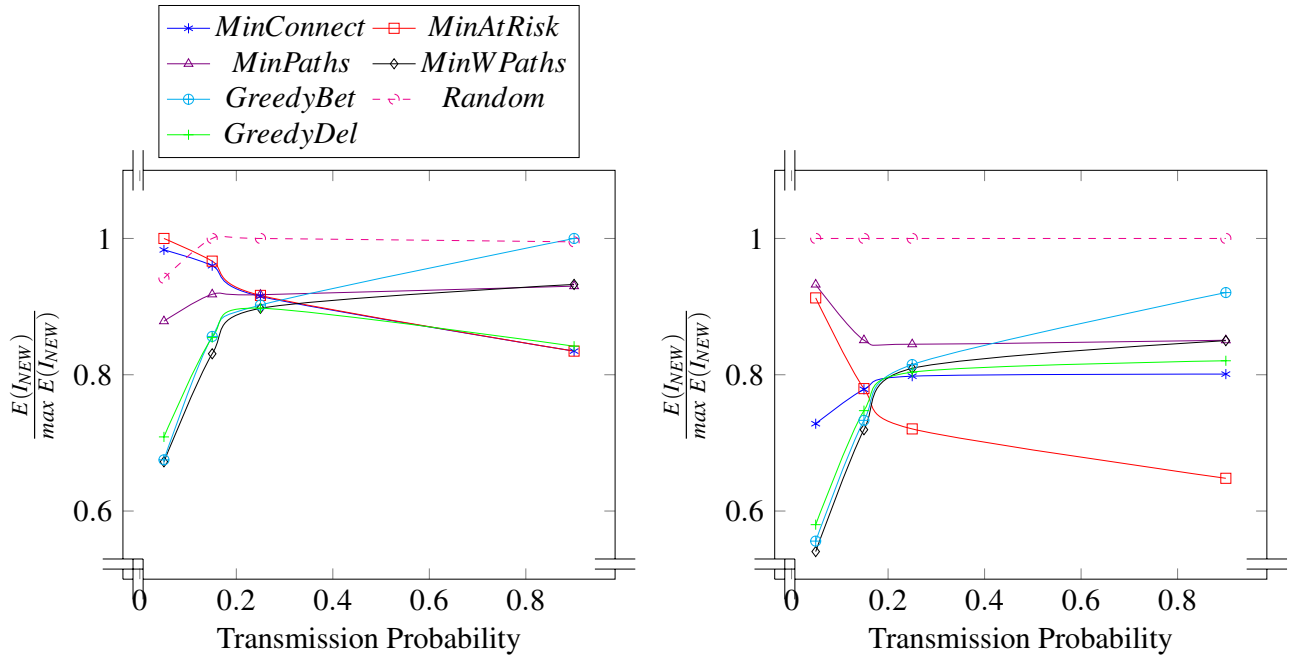
Table 2: Factors and their values in the experiments

We compare the link removal methods both based on the optimal solutions of our models and also based on the solutions using the greedy algorithms. We use networks of 12 nodes for the former set of comparisons, and networks of 50 and 150 nodes for the latter set of comparisons. Recall that the MinPaths and the MinWPaths can only be solved for small networks. Thus, networks of 12 nodes is used for the comparisons involving optimal solutions of the models. We performed the former set of comparisons (Figures 7 and 8) only on Random networks. But, we performed the latter set of comparisons (Figures 9, 10, 11, and 12) on Random and Scale-free networks. Networks of 150 nodes are used in the experiments involving Scale-free networks. Note that the legend above figure 7 is applicable for all the plots used in the comparisons.

6.2. Comparison of the methods

Figures 7, 8, 9, 10, 11, and 12 show how the relative effectiveness of the link removal methods vary with respect to transmission probability. This plots are created using normalized values of $E(I_{NEW})$ and $E(T)$. The maximum value in the denominator is taken over the seven methods for each combination of factor settings.

It is clear from figures 7, 9, and 11 that the MINATRISK model is much more effective than all the other methods in minimizing the spread of infections when the probability of transmission is not very low. When the probability of transmission is very low, MINWPATHS is the most effective in minimizing the spread of infections. It suggests that the infected nodes recover before spreading infections through paths of low transmission probability. Thus, only the paths of high transmission probability need to be removed. GREEDYBET and GREEDYDEL methods are comparable with the MINWPATHS model in most of the scenarios with low transmission probability. It is also clear from these figures that as the fraction of infected nodes increases relative to the fraction of links that can be removed, the transmission probability beyond which the MINATRISK model is more effective than other methods decreases. For example, in figure 11a, 0.1 is the transmission probability beyond which the MINATRISK model is the most effective, and in figure 11b that transmission probability is 0.05. Note that the ratio $\frac{Infected}{Links\,removed}$ is equal to $\frac{20\%}{20\%}$ in figure 11a and $\frac{30\%}{20\%}$ in figure 11b.



(a) Infected = 10%, Links removed = 10%

(b) Infected = 20%, Links removed = 20%

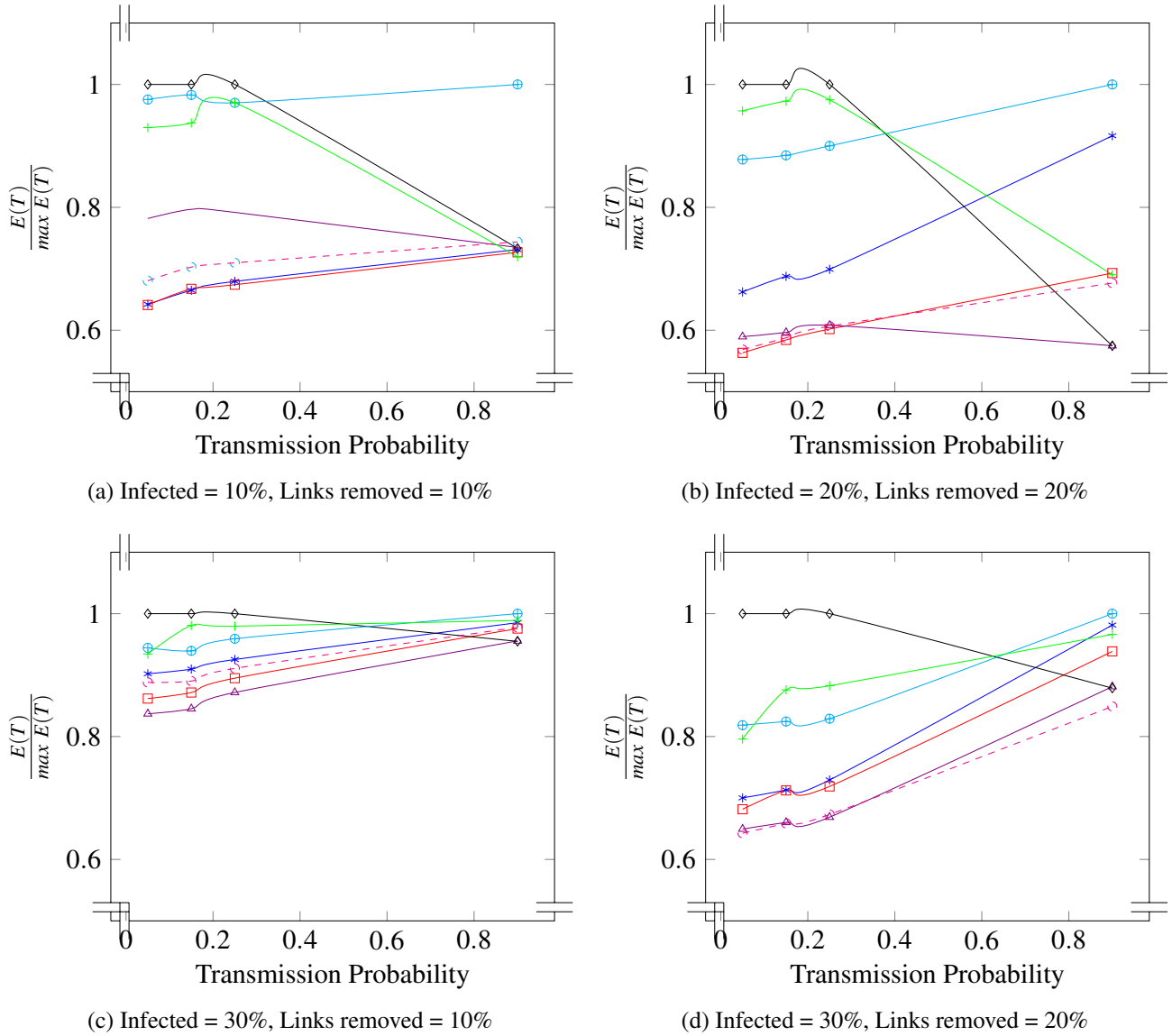
(c) Infected = 30%, Links removed = 10%

(d) Infected = 30%, Links removed = 20%

Figure 7: Relative effectiveness of the link removal methods to minimize the average new infections (models are solved to optimality using CPLEX)

Figures 8, 10, and 12 demonstrate that the MINWPATHS model is the most effective in slowing down the spread when the probability of transmission is not very high. Effectiveness of the GREEDYDEL and the GREEDYBET methods are the closest to the MINWPATHS model for low values of transmission probability. However, as the probability of transmission increases, effectiveness of the MINWPATHS model drops and becomes worse than the GREEDYBET and GREEDYDEL methods in most scenarios. Effectiveness of the GREEDYBET method is not particularly con-

sistent. In some scenarios, it is the second or the third most effective method at low transmission probabilities and the most effective method at high transmission probabilities (Figure 8, 10a, and 12a). But, in some other scenarios, effectiveness of this method is worse than most of the other methods (Figure 10b, and 12b). The MINATRISK model which is very effective in minimizing the spread of infections is inferior in slowing down the spread of infections in most scenarios. However, the MinAtRisk model performs quite well in figure 12b. This is an indication that when there are too many infected nodes and not enough links can be removed, complete isolation of the susceptible nodes is a good method also to slow down the spread.



(a) Infected = 10%, Links removed = 10%

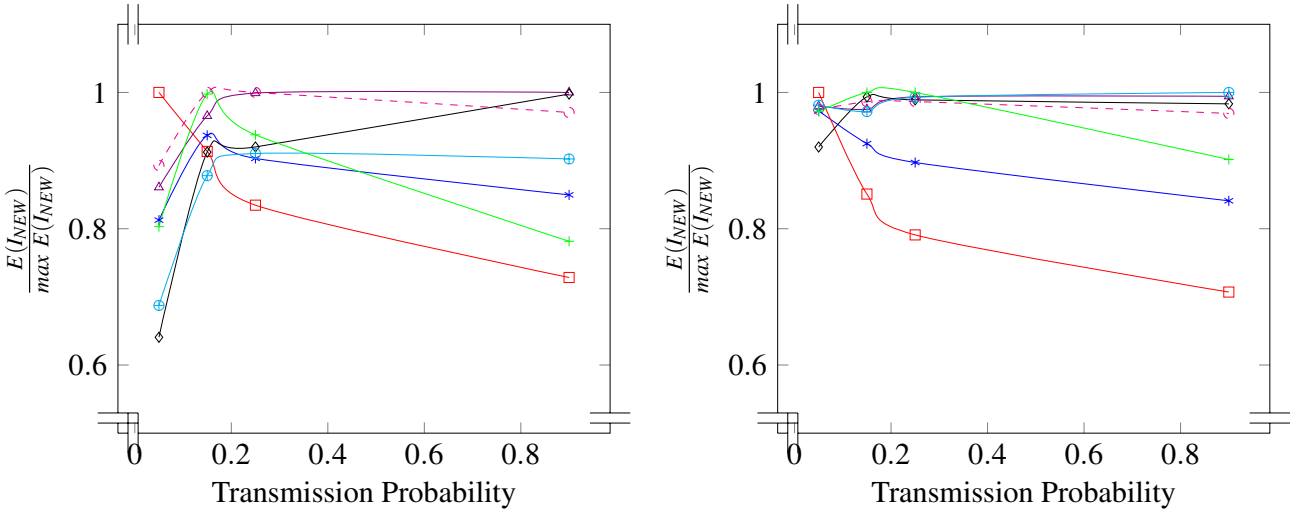
(b) Infected = 20%, Links removed = 20%

(c) Infected = 30%, Links removed = 10%

(d) Infected = 30%, Links removed = 20%

Figure 8: Relative effectiveness of the link removal methods to maximize the time to infect half of the susceptible nodes (models are solved to optimality using CPLEX)

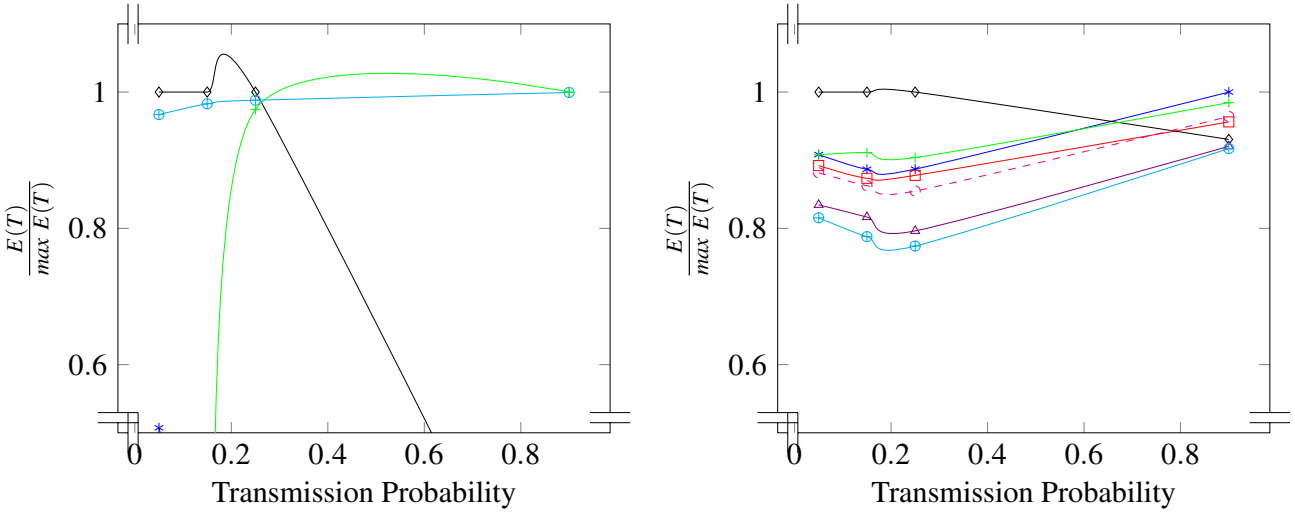
Patterns of the plots in figures 7 and 8 are similar to their counterparts among figures 9, 10, 11, and 12. Recall that the first set of figures are generated using the optimal solutions (by CPLEX) of our models, whereas, the second set is generated using the solutions by the greedy algorithms. The fact that the patterns are similar is another validation of the effectiveness of the greedy algorithms.



(a) Infected = 20%, Links removed = 20%

(b) Infected = 30%, Links removed = 20%

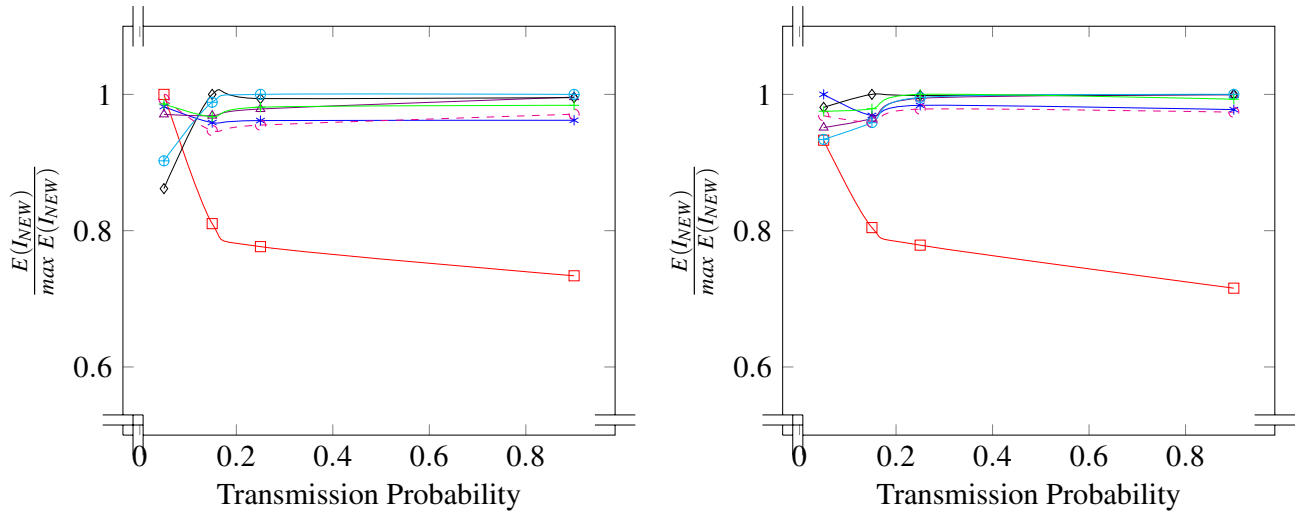
Figure 9: Relative effectiveness of the link removal methods to minimize the average new infections (models are solved by greedy algorithms), Random network, n = 50



(a) Infected = 20%, Links removed = 20%

(b) Infected = 30%, Links removed = 20%

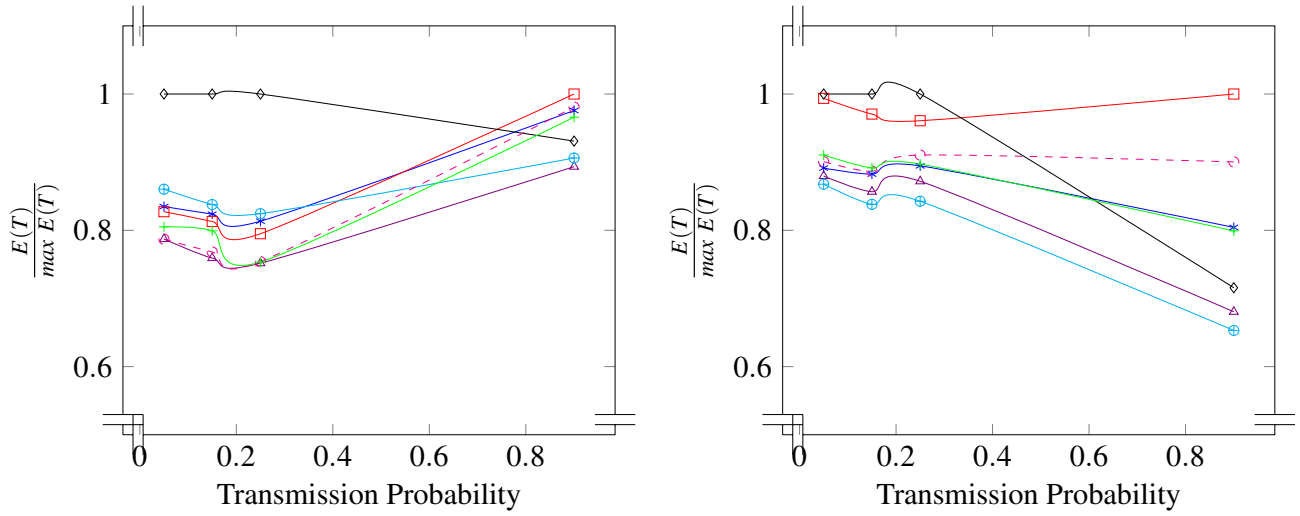
Figure 10: Relative effectiveness of the link removal methods to maximize the time to infect half of the susceptible nodes (models are solved by greedy algorithms), Random network, n = 50



(a) Infected = 20%, Links removed = 20%

(b) Infected = 30%, Links removed = 20%

Figure 11: Relative effectiveness of the link removal methods to minimize the average new infections (models are solved by greedy algorithms), Scale-free network, n = 150



(a) Infected = 20%, Links removed = 20%

(b) Infected = 30%, Links removed = 20%

Figure 12: Relative effectiveness of the link removal methods to maximize the time to infect half of the susceptible nodes (models are solved by greedy algorithms), Scale-free network, n = 150

7. Conclusion

This paper investigates the problem of removing a set of links from a network to minimize the spread of infections. For that purpose, we develop four network interdiction models and formulate them as mixed-integer programs. We analyze a decomposition technique for the models and explored their efficacy in solving the mixed-integer programming formulations. We also proposed

greedy algorithms for the models. Then, we compared our methods along with random link removal, the link removal method proposed by Kimura et al. [16], and a method based on modified betweenness centrality in minimizing the spread of infections.

Our major findings are as follows:

1. The MINCONNECT and the MINATRISK models for networks with more than 125 nodes can be solved to optimality within a reasonable time (2 hours). The MinAtRisk-Z formulation can be solved for networks with more than 150 nodes in less than 2 hours.

2. Our greedy algorithms can solve problem instances with a network size of 200 nodes in less than 10 minutes. On an average, more than 60% of the solutions for the MINCONNECT model are optimal, and 70% of the MINATRISK solutions are optimal. Average optimality gaps of the solutions for the MINPATHS and MINWPATHS algorithms are both less than 5%.

3. The probability of transmission is an important factor in determining the effectiveness of the link removal methods. When infected nodes do not recover from infection (SI simulation) and the transmission probability is low to moderate, the most effective strategy in slowing down the spread of infections is to remove links to reduce the weighted number of paths (MINWPATHS model). The effectiveness of this link removal method relative to other methods increases as the probability of transmission decreases. Therefore, intervention policies should be based on removing paths of high transmission probabilities to slow down spread for relatively less virulent infections.

4. When the probability of transmission is moderate to high and infected nodes can recover from infection (SIR simulation), the most effective strategy in minimizing the number of new infections is to remove links to minimize the number of susceptible nodes at risk of infection (MINATRISK model). The effectiveness of this method relative to other methods increases as the probability of transmission increases. So, for highly virulent infections, as many susceptible nodes as possible should be completely isolated from the rest of the nodes.

Potential future studies could include: a) developing approximation algorithms to solve the optimization models for large problems, b) extending the idea of this paper to combined node and link-based spread control, c) extend the idea to other applications such as finding optimal travel restrictions in transportation networks, optimal interdiction of terrorist networks, etc.

References

[1] Albert, R., Jeong, H., Barabasi, A.-L., 2000. Error and attack tolerance of complex networks. Nature 406 (6794), 378–382.
[2] Barabási, A.-L., Albert, R., Oct. 1999. Emergence of Scaling in Random Networks. Science 286 (5439), 509–512.

[3] Boginski, V., Commander, C. W., 2009. Identifying critical nodes in protein-protein interaction networks. Clustering challenges in biological networks, 153–167.
[4] Brown, G., Carlyle, M., Salmerón, J., Wood, K., Nov. 2006. Defending Critical Infrastructure. Interfaces 36 (6), 530–544.
[5] Commander, C. W., 2007. Optimization problems in telecommunications with military applications. Ph.D. thesis, University of Florida.
[6] Costa, A. M., Jun. 2005. A survey on benders decomposition applied to fixed-charge network design problems. Computers & Operations Research 32 (6), 1429–1450.
[7] Cplex, I. L. O. G., 2007. 11.0 User's Manual. ILOG SA, Gentilly, France.
[8] Enns, E. A., Mounzer, J. J., Brandeau, M. L., Feb. 2012. Optimal link removal for epidemic mitigation: A two-way partitioning approach. Mathematical Biosciences 235 (2), 138–147.
[9] Erdos, P., Renyi, A., 1960. On the evolution of random graphs. Publ. Math. Inst. Hungar. Acad. Sci 5, 17–61.
[10] Eubank, S., Guclu, H., Anil Kumar, V. S., Marathe, M. V., Srinivasan, A., Toroczkai, Z., Wang, N., May 2004. Modelling disease outbreaks in realistic urban social networks. Nature 429 (6988), 180–184.
[11] Guimerà, R., Mossa, S., Turtschi, A., Amaral, L. A. N., May 2005. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. Proceedings of the National Academy of Sciences 102 (22), 7794–7799.
[12] He, J., Liang, H., Yuan, H., 2011. Controlling infection by blocking nodes and links simultaneously. Vol. 7090 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 206–217.
[13] HHS, 2000. The great pandemic - The United States in 1918-1919. Tech. rep., United States Department of Health and Human Services.
[14] Holme, P., Kim, B. J., Yoon, C. N., Han, S. K., 2002. Attack vulnerability of complex networks. Physical Review E 65 (5), 056109.
[15] Kimura, M., Saito, K., Motoda, H., 2008. Minimizing the Spread of Contamination by Blocking Links in a Network. Vol. 8. AAAI, pp. 1175–1180.
[16] Kimura, M., Saito, K., Motoda, H., Apr. 2009. Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data (TKDD) 3 (2), 9.
[17] Latora, V., Marchiori, M., Apr. 2004. How the science of complex networks can help developing strategies against terrorism. Chaos, Solitons & Fractals 20 (1), 69–75.
[18] Marcelino, J., Kaiser, M., Aug. 2009. Reducing influenza spreading over the airline network. PLoS Currents 1, RRN1005+.
[19] Marcelino, J., Kaiser, M., 2012. Critical paths in a metapopulation model of H1N1: Efficiently delaying influenza spreading through flight cancellation. PLoS currents 4.
[20] Nandi, A. K., Medal, H. R., 2013. Optimization models to control infectious disease spread in networks. Proceedings of the 8th INFORMS Workshop on Data Mining and Health Informatics.
[21] Pastor-Satorras, R., Vespignani, A., Apr. 2001. Epidemic spreading in scale-free networks. Physical Review Letters 86 (14), 3200–3203.
[22] Prakash, Chakrabarti, D., Valler, N., Faloutsos, M., Faloutsos, C., Jul. 2012. Threshold conditions for arbitrary cascade models on arbitrary networks. Knowledge and Information Systems 33 (3), 549–575.
[23] Resende, M. G. C., Pardalos, P. M., 2008. Handbook of optimization in telecommunication networks. Springer.
[24] Sanger, D. E., Jun. 2012. Obama order sped up wave of cyberattacks against Iran. The Newyork Times.
[25] Satorras, R. P., Vespignani, A., Mar. 2002. Epidemic dynamics in finite size scale-free networks. Physical Review E 65 (3), 035108+.
[26] Shen, S., Smith, J. C., Goli, R., Aug. 2012. Exact interdiction models and algorithms for disconnecting networks via node deletions. Discrete Optimization 9 (3), 172–188.
[27] Taniguchi, C. M., Emanuelli, B., Kahn, C. R., Feb. 2006. Critical nodes in signalling pathways: insights into insulin action. Nat Rev Mol Cell Biol 7 (2), 85–96.
[28] Tong, H., Prakash, B. A., Rad, T. E., Faloutsos, M., Faloutsos, C., 2012. Gelling, and melting, large graphs by edge manipulation. In: Proceedings of the 21st ACM international conference on Information and knowledge management. CIKM '12. ACM, New York, NY, USA, pp. 245–254.
[29] Ventresca, M., Aleman, D., Sep. 2014. A Randomized Approximation Algorithm for the Critical Node Detection Problem. Computers & Operations Research 43, 261–270.
[30] Wang, Y., Chakrabarti, D., Wang, C., Faloutsos, C., Oct. 2003. Epidemic spreading in real networks: an eigenvalue viewpoint. In: Reliable Distributed Systems, 2003. Proceedings. 22nd International Symposium on. Vol. 0. IEEE, Los Alamitos, CA, USA, pp. 25–34.
[31] Watts, D. J., Strogatz, S. H., Jun. 1998. Collective dynamics of 'small-world' networks. Nature 393 (6684), 440–442.
[32] WHO, Beaglehole, R., Irwin, A., 2004. The World Health Report, 2004: Changing History. Tech. rep., World Health Organization.