Optimal Response to Epidemics and Cyber Attacks in Networks

Noam Goldberg * Sven Leyffer† Ilya Safro‡

May 13, 2015

Abstract

This paper introduces novel formulations for optimally responding to epidemics and cyber attacks in networks. In our models, at a given time period, network nodes (e.g., users or computing resources) are associated with probabilities of being infected, and each network edge is associated with some probability of propagating the infection. A decision maker would like to maximize the network's utility; keeping as many nodes open as possible, while satisfying given bounds on the probabilities of nodes being infected in the next time period. The model's relation to previous deterministic optimization models and to both probabilistic and deterministic asymptotic models is explored. Initially, maintaining the stochastic independence assumption of previous work, we formulate a nonlinear integer program with high order multilinear terms. We then propose a quadratic formulation that provides a lower bound and feasible solution to the original problem. Further motivation for the quadratic model is given by showing that it alleviates the assumption of stochastic independence. The quadratic formulation is then linearized in order to be solved by standard integer programming solvers. We develop valid inequalities for the resulting formulations.

Keywords: Nonlinear integer programming, cutting planes, network optimization, probability bounds, cybersecurity, epidemiology.

AMS-MSC2010: 90C10, 90C35, 90B10, 90B18, 90C90, 92D30.

1 Introduction and Background

Cyber terror and other cyber threats have been on the rise with many individual incidents that are estimated to cost billions of dollars [12, 23]. Mathematical models of biological epidemics and their extensions to cybersecurity have been extensively investigated during past few decades; see, for example, [10, 19, 20]. In contrast to previous work that analyzes the asymptotic growth of infections, we consider a decision problem that arises at a given time period in a network where a certain infection may spread. Each node is associated with the probability of being infected at a given time period, and some of the nodes may be shut down in order to maintain desired bounds

*Department of Management, Bar-Ilan University, Ramat Gan, Israel 52900, noam.goldberg@biu.ac.il.
†MCS Division, Argonne National Laboratory, Argonne, IL 60439, leyffer@mcs.anl.gov.
‡Computer Science Department, Clemson University, isafro@g.clemson.edu.

on the infection probabilities of the nodes. Similar decision problems involving a response to spread of an infection or fire in a network have been introduced by [1] and [16]; as far as we are aware, however, previous work on optimal response to network infections has considered only deterministic models.

Probabilistic network optimization models have been extensively studied and applied in the literature of reliable network design [7, 26]. In network reliability one is usually interested in designing a network that can withstand the possibility that some of the terminal nodes become disconnected, with a certain probability. In contrast in our application: We would like to remove rather than to install nodes. The probability of a node becoming infected depends on the probability of contracting the virus from a neighbor. A bound on the probability applies to each network node, rather than to the entire network or to pair of terminals. Finally, the objective is to maximize rather than minimize an increasing function of the open nodes. In the following subsections we first expand on mathematical models of epidemics and their extensions to cybersecurity. Next, we elaborate on previous work on the decision problems of determining which nodes to treat in the network in order to mitigate an infection.

1.1 Models for the Spread of Computer Viruses Motivated by Epidemiology

Traditional epidemiological models use a few parameters to estimate the growth of the infected population as a whole and, in particular, to predict whether an epidemic dies out at the limit or whether it ensues. A key input parameter of the epidemic in these models is the virus *birth rate*, β, the rate at which healthy individuals become infected with a virus when coming in contact with infected individuals. Similarly, a virus *death rate*, δ, is the rate at which infected individuals are cured. The Kephart-White model [20] applies epidemiology-based modeling to computer viruses. Let η denote the size of the infected population and k denote the average degree of the contact graph. The model is captured by the ordinary differential equation

$$\frac{d\eta}{dt} = \beta k \eta \left(1 - \frac{\eta}{n}\right) - \delta\eta.$$

The steady state solution of this model may provide an approximate solution of the growth of an epidemic in networks where the contact among individuals is sufficiently homogeneous (which is unlikely to hold for computer networks). However, this model does not model specific individuals, nor does it suggest which individuals should be treated with limited resources.

Some recent models address more details such as an underlying structure for the propagation of the infection. Typically, the structure can be described by a *contact graph*: a graph $G = (V, E)$ with n nodes corresponding to individuals, or groups, of the general population and edges corresponding to possible contacts among individuals. Let $N(i)$ be the set of neighbors of node i. Let $h_{i,t}$ be the probability that a node i is not infected from one of its neighbors at time period t, and let $\pi_{i,t}$ be the probability that a node i is infected at time t (with $\pi_{i,t} = 1 - h_{i,t}$). Chakrabarti et al. [10] propose a probabilistic model in which

$$h_{i,t} = \prod_{j \in N(i)} (1 - \rho\pi_{j,t-1}), \tag{1}$$

where ρ is the probability of a node to contract the virus from an infected neighbor. This model assumes independence of the events of a node being infected from each of its neighbors, an assumption that is justified when time steps are small enough so that the effect of the positive correlation of infection events is assumed to be negligible. Further, the assumption of stochastic independence significantly simplifies their analysis. Chakrabarti et al. [10] analyze an *epidemic threshold* for a dynamical system associated with (1); the magnitude of this threshold compared with ρ/δ determines whether an infection ensues or becomes endemic. They determine a threshold value that is inversely proportional to the largest eigenvalue of the adjacency matrix of the network. Note that this result is derived under a restrictive assumption that the probability of infection spreading along every edge equals the same probability ρ.

1.2 Deterministic Network Response Optimization Models

A well studied deterministic network response model is known as the firefighter problem; see [2] and [16] and references therein. The problem is to iteratively decide which nodes to defend (immunize) while a fire (infection) is spreading in the network. At each time step the network nodes are partitioned into three parts: inflamed (infected) nodes, defended (recovered) nodes, and susceptible nodes. The fire spreads out deterministically from each node to its neighbors. To prevent the fire from spreading, at each step only a limited number of nodes can be defended. Once a node is defended (or recovered), it cannot be compromised at succeeding iterations. The objective is to contain the fire while minimizing the total number of nodes lost. The problem is known to be $\mathcal{NP}$-hard even for tree networks with a maximum node degree of three [13]. An alternative formulation of the problem, also known to be $\mathcal{NP}$-hard, has the objective of minimizing the cost of saving all nodes from the fire [2].

The model of the firefighter problem is inherently sequential. However, the model is restrictive in assuming that all unprotected neighbors of an infected node become infected in the next time period. It also assumes that the fire may not spread beyond the neighbors of an infected node within a single time period. In practice there is a decision problem to be solved at a particular moment in time. Further, the time periods in which action can be taken to respond to the epidemic may not necessarily correspond to periods over which the epidemic may spread.

In contrast to the firefighter's problem, Altunay et al. [1] suggest an optimization problem for determining an optimal response to cyber attacks. The network is modeled by an undirected graph $G = (V, E)$ with vertex set V, consisting of $n = |V|$ nodes and edge set E, with $|E| = m$. The set of nodes correspond to sites, servers, or individual users, and the edges may correspond to communication links or connections. Further suppose $V_c \subseteq V$ is a set of sites known to be compromised or infected. Accordingly, $V_u = V \setminus V_c$ is the set of uncompromised or susceptible sites. The network operator would like to maximize the utility of the network resources that remain open while shutting down some of the nodes to maintain an acceptable level of threat.

Let $x \in \{0,1\}^n$ be a vector of decision variables; $x_i = 1$ means that node i remains open, and $x_i = 0$ corresponds to shutting it down. A network utility function measures the usability of the network as a function of the node configuration $x \in \{0,1\}^n$, for some $0 \leq W \in \mathbb{R}^{n \times n}$:

$$u(x) = \sum_{\{i,j\} \in E} W_{ij} x_i x_j.$$

This choice of a bilinear utility function is appropriate when the graph models the application layer of a communications network, for example when nodes and edges correspond to user groups or servers and, respectively, communication patterns between users and servers. Different choices of utility functions are common for performance modeling of networks; see for example [18]. For an edge $\{i,j\} \in E$, W_{ij} denotes its value as long as it remains open (an edge is open if both endpoint nodes are open). Altunay et al. [1] formulate an optimization problem for determining $x \in \{0,1\}^n$ that maximizes u while the threat is determined endogenously by a decision variable $t_i \in [0, T_i]$ for each node i, and its given bound $T_i \leq 1$. The details of the model are described, and the problem's computational complexity is established in Appendix A.

2 Preliminary Analysis and Models

We first introduce a new probabilistic model of network response. A preliminary optimization model is introduced under the assumption of stochastic independence of the events of the infection propagating along the edges of the network. We then propose an approximate model (in fact a restriction) that also alleviates the independence assumption, and that is solved more efficiently in our experiments.

2.1 Probabilistic Network Response

Motivation for our probabilistic model can be drawn from the application described in [1], though some of the modeling assumptions with respect to the spread of the infection are similar to those of the asymptotic analysis appearing in [10].

For $i \in V$, let $N(i) = \{j \in V \mid \{i,j\} \in E\}$ be the set of nodes that are connected to node i by an edge, and let

$$x_i = \begin{cases} 1 & \text{node } i \text{ is open} \\ 0 & \text{otherwise.} \end{cases}$$

In general, the probability of an open node i remaining healthy (i.e., uninfected) in a given time period is given by

$$h_i(x) = P\left(\bigcap_{j \in N(i)} \{i \text{ is not infected by node } j \mid x_j = 1\} \right),$$

where $P(A)$ is the probability of event A. As in [10], we may assume that the events of nodes being infected in the previous time period are independent of one another. Although it may be more realistic to assume that the events that neighboring nodes are infected are dependent on one another, an independence assumption simplifies the computation (i.e., allowing one to tractably evaluate of $h_i(x)$) and modeling (i.e., relieving one from having to specify the joint probabilities). Let p_{ij} be the probability of node i being infected by node j. Let π_i be the probability of node i being infected at a given time period. Then, under the independence assumption, the probability of node i not being infected in the following time period is then

$$h_i(x) = \prod_{j \in N(i)} (1 - p_{ij}\pi_j x_j). \tag{2}$$

The probability of a node i being infected equals the probability of at least one of its non-removed neighbors being infected times the probability that the neighbor infects i. Note that in the following only a single time period is considered, although one may experiment with the effectiveness of our models over several time periods, for example, by repeatedly applying them within a discrete event simulation.

We would like to choose a network configuration $x \in \{0,1\}^n$ that maximizes the utility $u(x)$ while bounding the infection probability of each node $i \in V$

$$1 - h_i(x) = P\left(\bigcup_{j \in N(i)} \{i \text{ is infected by node } j \mid x_j = 1\}\right)$$

by the given T_i. The resulting formulation is

$$\begin{aligned}
&\underset{x}{\text{maximize}} && u(x) && && \text{(3a)}\\
&\text{subject to} && x_i - h_i(x) \leq T_i && i \in V, && \text{(3b)}\\
& && x \in \{0,1\}^n. && && \text{(3c)}
\end{aligned}$$

The constraint (3b) for $i \in V$ ensures that either the node is shut down, that is $x_i = 0$, or if $x_i = 1$, then the probability of i being infected is at most T_i. The following proposition establishes the computational complexity of (3).

Proposition 1. *Problem* (3), *with $W_{ij} = 1$ for all $\{i,j\} \in E$, is $\mathcal{NP}$-hard.*

The proof of the proposition is deferred to Appendix B.

2.2 Bounding the Infection Probability

When assuming independence of the infection in the previous period, i.e., that each h_i is given by (2), then (3) is an integer nonlinear program (INLP) with high-order (but polynomially many) multilinear terms. If the infection events are not independent, then to express each h_i as a function of x may become intractable as it requires an exponential number of multilinear terms using the inclusion-exclusion formula. We now consider a restriction of the optimization problem using a probability bound that can be used to approximate (3): it provides a feasible solution and a corresponding lower bound on its optimal solution. Specifically, we bound the probability of the union of events of a node being infected from each of its neighbors. A known bound for the probability of the union of multiple events is Hunter's bound [17,28]: for a sample space Ω, given a finite set of events $\{A_i \subseteq \Omega \mid i \in N\}$,

$$P(\cup_{i\in N} A_i) \leq \sum_{i\in N} P(A_i) - \sum_{\{i,j\}\in\mathcal{T}} P(A_i \cap A_j), \tag{4}$$

where $\mathcal{T}$ is a maximum weighted spanning tree of the complete graph with vertex set N, having the probability $P(A_i \cap A_j)$ as the weight of each edge $\{i,j\} \in N \times N$. The union bound, which does not use the joint probabilities, corresponds only to the sum of the positive terms on the right-hand side of (4), and is not as tight as Hunter's bound. Although, even tighter upper bounds exist, these may require the computation of third- or higher-order terms; see for example [9].

For each $i \in V$, let $\mathcal{T}_i$ be the maximum weighted spanning tree of the complete graph with nodes $N(i)$, and each edge $\{j,k\}$ weighted by the joint probability of i being infected by both j and k. To simplify the notation, for each $\{i,j\} \in E$, let $q_{ij} = p_{ij}\pi_j$. For each $\{j,k\} \in \mathcal{T}_i$, let r_{ijk} denote the joint probability that i is infected by both j and k (and note that $r_{ijk} = r_{ikj}$). For $i \in V$, let M_i be a suitably large constant; for example, it suffices to set $M_i = \sum_{j\in N(i)} q_{ij}$. Then, for each $i \in V$, we replace the constraint (3b) by a more conservative (tighter) constraint

$$\begin{aligned} &\sum_{j\in N(i)} q_{ij}x_j - \sum_{\{j,k\}\in\mathcal{T}_i} r_{ijk}x_jx_k \leq T_i - M_i(x_i - 1), \\ \Leftrightarrow g_i(x) \equiv M_i x_i + &\sum_{j\in N(i)} q_{ij}x_j - \sum_{\{j,k\}\in\mathcal{T}_i} r_{ijk}x_jx_k \leq M_i + T_i. \end{aligned}$$

The resulting formulation is a INLP with a set of bilinear constraints:

$$\begin{aligned} &\underset{x}{\text{maximize}} && u(x) && && \text{(5a)} \\ &\text{subject to} && g_i(x) \leq M_i + T_i && \text{for } i \in V && \text{(5b)} \\ & && x \in \{0,1\}^n. && && \text{(5c)} \end{aligned}$$

Although the formulation (5) has bilinear rather than general multilinear terms, and is simpler compared with the nonlinear formulation (3), the next proposition establishes that it remains $\mathcal{NP}$-hard.

Proposition 2. *The problem* (5) *with $W_{ij} = 1$ for every $\{i,j\} \in E$ is $\mathcal{NP}$-hard.*

The proof of the proposition is given in Appendix C.

Hunter's bound, and accordingly (5), applies in the general case that the events corresponding to the probabilities q_{ij} are not independent for all $i \in V$ and $j \in N(i)$. In our current experiments, however, we focus on the simpler case of stochastic independence, which implies that $r_{ijk} = q_{ij}q_{ik}$ for each $i \in V$ and $j,k \in N(i)$. In Appendix D we compare the quality of optimal solutions of (5) when used to approximate the solution of (3) with $h_i(x)$ given by (2) (and hence it models the stochastic independent case). These experiments confirm that the formulation (5) can be solved faster than (3) while better approximating the optimal solution than a simple union bound based formulation. We would like to emphasize, however, that (5) applies more generally when $h_i(x)$

is not given by (2), and in which case it may not even be tractable to write (5) as a mathematical program when using the inclusion-exclusion formula to express $h_i(x)$.

In the next section we develop computational techniques to solve (5) more effectively as an integer program.

3 Computational Techniques

We now consider inequalities that are valid for (5) and that tighten the continuous relaxation upper bound in order to ultimately speed the computation of the integer program.

3.1 Valid Cover Inequalities

A cover inequality for $S \subseteq V$ is a valid linear inequality of the form

$$\sum_{j \in S} x_j \leq |S| - 1. \tag{6}$$

Inequalities of this form were first developed as valid inequalities for linear knapsack problems; see [4,5]. In particular, (6) must hold if S is the support of $x \in \{0,1\}^n$ for which a corresponding knapsack constraint is violated. Such inequalities and their extensions have been used for general mixed-integer programs (MIPs), and have also been specialized for specific applications; see for example [21] for a recent application to a network optimization problem. Further, cover inequalities have recently been extended for conic-quadratic nonlinear formulations [3]. However, we note that these recent extensions, as well as standard applications to knapsack constraints, all require that the left-hand side of the associated constraint is monotone in x; for example with linear knapsack constraints binary variables with negative coefficients need to be complemented (which in turn increases the constant in right-hand side).

In the case of formulation (5), note that for every $i \in V$, $g_i(x)$ has negative quadratic coefficients. For every $i \in V$ and $j \in N(i)$ (assuming that $r_{ijk} = q_{ij}q_{ik}$), although

$$\frac{\partial g_i}{\partial x_j} = q_{ij} - \sum_{\substack{k \in N(i): \\ \{k,j\} \in \mathcal{T}_i}} r_{ijk} x_k = q_{ij} \left(1 - \sum_{\substack{k \in N(i): \\ \{k,j\} \in \mathcal{T}_i}} q_{ik} x_k \right)$$

tends to be positive, it may be negative if $\sum_{\substack{k \in N(i): \\ \{k,j\} \in \mathcal{T}_i}} q_{ik} x_k > 1$. We note, however, that the negative coefficients of the quadratic terms tend to be small in our application; $r_{ijk} \leq \min\{q_{ij}, q_{ik}\}$ for each $i \in V$ and $\{j,k\} \in \mathcal{T}_i$ in general, and in particular in the case of stochastic independence when $r_{ijk} = q_{ij}q_{ik}$. Also, since $\mathcal{T}_i$ is a tree the total number of quadratic terms of g_i, is linear in the number of nodes and is bounded by $|N(i)| - 1$. Further, for $j \in N(i)$, the number of adjacencies in the tree $|\{k \in N(i) \mid \{k,j\} \in \mathcal{T}_i\}|$, which is the number of negative quadratic terms in $\partial g_i / \partial x_j$, tends to be small. Nevertheless, in general $g_i(x)$ is not monotone in x. To apply standard knapsack-cover inequalities to constraints such as (5b), one may consider linearizing and then complementing the negative variables. This approach, however, results in weak inequalities.

In the following let $\mathcal{S}(x) \equiv \{j \in V \mid x_j = 1\}$ and let e^j denote the unit vector with one at the jth component. Lemma 3 establishes a weaker condition than monotonicity that is useful for deriving our novel valid inequalities.

Lemma 3. *Suppose $x, y \in \{0,1\}^n$ and $i \in \mathcal{S}(x) \cap \mathcal{S}(y)$ that satisfy $\mathcal{S}(x) \cap N(i) \subseteq \mathcal{S}(y) \cap N(i)$. Further suppose that for all $\{j,k\} \in \mathcal{T}_i$, $r_{ijk} \leq \min\{q_{ij}, q_{ik}\}$. Then it follows that*

$$g_i(y) \geq g_i(x) + \sum_{j \in (\mathcal{S}(y) \setminus \mathcal{S}(x)) \cap N(i)} [g_i(x + e^j) - g_i(x) - q_{ij}].$$

Proof. Letting x and y be defined as in the claim of the lemma, since $(\mathcal{S}(x) \setminus \mathcal{S}(y)) \cap N(i) = \emptyset$ we have that

$$g_i(y) = g_i(x) + \sum_{j \in (\mathcal{S}(y) \setminus \mathcal{S}(x)) \cap N(i)} q_{ij} - \sum_{\substack{\{j,k\} \in \mathcal{T}_i: \\ j \in \mathcal{S}(y) \setminus \mathcal{S}(x), k \in \mathcal{S}(x)}} r_{ijk} - \sum_{\substack{\{j,k\} \in \mathcal{T}_i: \\ j \in \mathcal{S}(y) \setminus \mathcal{S}(x), k \in \mathcal{S}(y) \setminus \mathcal{S}(x)}} r_{ijk}. \tag{7}$$

By the hypothesis assuming that for all $\{j,k\} \in \mathcal{T}_i$, $r_{ijk} \leq \min\{q_{ij}, q_{ik}\}$ and by $\mathcal{T}_i$ being a tree that spans $j \in (N(i) \cap \mathcal{S}(y)) \setminus \mathcal{S}(x)$, it follows that

$$\sum_{\substack{\{j,k\} \in \mathcal{T}_i: \\ j \in \mathcal{S}(y) \setminus \mathcal{S}(x), k \in \mathcal{S}(y) \setminus \mathcal{S}(x)}} r_{ijk} \leq \sum_{\substack{\{j,k\} \in \mathcal{T}_i: \\ j \in \mathcal{S}(y) \setminus \mathcal{S}(x), k \in \mathcal{S}(y) \setminus \mathcal{S}(x)}} \min\{q_{ij}, q_{ik}\} \leq \sum_{j \in (\mathcal{S}(y) \setminus \mathcal{S}(x)) \cap N(i)} q_{ij},$$

implying that

$$g_i(y) - g_i(x) \geq - \sum_{\substack{\{j,k\} \in \mathcal{T}_i: \\ j \in \mathcal{S}(y) \setminus \mathcal{S}(x), k \in \mathcal{S}(x)}} r_{ijk} = \sum_{j \in (S(y) \setminus \mathcal{S}(x)) \cap N(i)} [g_i(x + e^j) - g_i(x) - q_{ij}]. \qquad \square$$

Lemma 3 identifies a lower bound on $g_i(y)$ for $y \in \{0,1\}^n$ in terms of a given $x \leq y$. In the following proposition, we use this bound to prove a sufficient condition for a given set $S \subseteq V$ to define a valid inequality (6). Further, this sufficient condition is testable; it is straightforward to determine whether this condition holds in polynomial (in fact linear) time.

Proposition 4. *Suppose that for some $i \in V$ and $x \in \{0,1\}^n$ with $S \equiv (\mathcal{S}(x) \cap N(i)) \cup \{i\}$,* (5b) *is violated and*

$$g_i(x + e^j) - g_i(x) - q_{ij} \quad \geq \quad -\frac{1}{|N(i)| - |S| + 1}(g_i(x) - T_i - M_i) \qquad \text{for all } j \in N(i) \setminus S. \tag{8}$$

Then (6) *is a valid inequality for* (5).

Proof. Assume the hypothesized conditions of the proposition hold, and that $y \in \{0,1\}^n$ is a feasible solution of (5) satisfying $\bar{S} \equiv \mathcal{S}(y) \cap N(i) \supseteq \underline{S} \equiv \mathcal{S}(x) \cap N(i) = S \setminus \{i\}$ and $y_i = 1$. Note that (5b) being violated by x implies that $x_i = 1$. Then, by Lemma 3, (8), and the fact that (5b) is violated by x, it follows that

$$\begin{aligned} g_i(y) \geq g_i(x) + \sum_{j \in \bar{S} \setminus \underline{S}} [g_i(x + e^j) - g_i(x) - q_{ij}] \quad &\geq \quad g_i(x) - \frac{|\bar{S}| - |\underline{S}|}{|N(i)| - |\underline{S}|}(g_i(x) - T_i - M_i) \\ = g_i(x) \frac{|N(i)| - |\bar{S}|}{|N(i)| - |\underline{S}|} + (T_i + M_i) \frac{|\bar{S}| - |\underline{S}|}{|N(i)| - |\underline{S}|} > T_i + M_i&, \end{aligned}$$

thereby establishing a contradiction to the feasibility of y. $\square$

The following example illustrates the use of inequality (6) for (5) and it will also be used to illustrate the differences from standard knapsack-cover inequalities.

Example 1. *Suppose* $V = \{1, 2, 3, 4\}$. $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}\}$. *Let* $q_{12} = q_{13} = q_{14} = \frac{1}{2}$. *Note that for* $i = 1$, $M_1 = \frac{3}{2}$ *and* $T_1 = \frac{1}{2}$ *the constraint* (5b)*:*

$$\frac{3}{2}x_1 + \frac{1}{2}(x_2 + x_3 + x_4) - \frac{1}{4}(x_2x_3 + x_3x_4) \leq 2$$

is violated with $x' = \begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix}^T$. *The inequality* (6) *with* $S = \{1, 2, 3\}$ *is given by*

$$x_1 + x_2 + x_3 \leq 2. \tag{9}$$

Evidently the condition (8) *is satisfied for* $4 \in N(1) \setminus S$*, for which* $x' + e^4 = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T$. *It follows by Proposition 4 that the inequality* (9) *is valid for* (5)*.*

Quadratic and higher-order INLPs can be linearized through standard techniques involving the introduction of additional auxiliary variables. For each $\{i, j\} \in E$, let $u_{ij} \in [0, 1]$ be a real variable used to linearize the corresponding bilinear term x_ix_j using the following additional constraints:

$$u_{ij} \leq x_i \qquad u_{ij} \leq x_j \qquad x_i + x_j - 1 \leq u_{ij} \qquad u_{ij} \geq 0. \tag{10}$$

Let $\bar{E} = \bigcup_{i \in V} \mathcal{T}_i \cup E$. We refer to (5) with each quadratic term x_ix_j for $\{i, j\} \in \bar{E}$ replaced by u_{ij}, and with the additional constraints (10), as the *linearization* of formulation (5). Also, we refer to the linearization of (5) with the integrality constraints (5c) replaced by $x \in [0, 1]^n$, as the *linear relaxation* of (5). We use the linearization in Section 4 in order to solve formulation (5) using a standard MIP solver. First, we consider an example that illustrates the difference of our cover inequalities from standard knapsack-cover inequalities that can be applied to the linearization of (5).

Example 2. *After linearizing and complementing negative variables of the constraint in Example 1, it results in the knapsack constraint*

$$\frac{3}{2}x_1 + \frac{1}{2}(x_2 + x_3 + x_4) + \frac{1}{4}(z_{23} + z_{34}) \leq \frac{5}{2},$$

where $z_{ij} = 1 - u_{ij}$ *for* $\{i, j\} \in E$*.*

Note that the inequality $x_1 + x_2 + x_3 \leq 2$ of Example 1 is not a valid knapsack-cover inequality for the constraint in Example 2. Further, note that although Generalized Upper Bound (GUB) knapsack cover inequalities [29] are designed to account for additional GUB constraints in conjunction with a knapsack constraint, the inequality considered in Example 1 cannot be derived as a GUB knapsack cover inequality either: The associated linearization constraints (10) do not follow the form of (nonoverlapping) GUB constraints. Therefore, Example 2 illustrates that the cover inequalities under consideration are essentially different and cannot be generated as knapsack-cover inequalities of the linearized formulation. Finally, valid inequalities that are stronger than (6), and which are analogous to extended knapsack-cover inequalities are considered in Section 3.2.

3.2 Extended Cover Inequalities

For knapsack constraints, extended covers provide stronger inequalities that can also be generated and separated efficiently in practice [4, 14, 15]. We now develop analogous extended cover inequalities for (5).

Definition 1. *Suppose* (6) *is a valid inequality for* (5) *for some* $S \subseteq V$. *Then* $X_{S,i} \subseteq V \setminus S$ *is an* extension of a covering set $S \subseteq V$, *for some* $i \in V$, *if*

$$q_{ij} - \sum_{\substack{k \in X_{S,i} \cup S \\ \{j,k\} \in \mathcal{T}_i}} r_{ijk} \;\geq\; \max_{\ell \in S} q_{i\ell}, \qquad \textit{for all } j \in X_{S,i}. \tag{11}$$

We refer to $E_{S,i} \equiv S \cup X_{S,i}$ as an *extended covering set*. For such a set the following proposition suggests valid inequalities that are stronger than (6).

Proposition 5. *Suppose* $x' \in \{0,1\}^n$ *and* $i \in V$ *so that, with* $S = (\mathcal{S}(x') \cap N(i)) \cup \{i\}$ *and* x *replaced by* x', (5b) *is violated and* (8) *is satisfied. If* $X_{S,i} \subseteq N(i) \setminus S$ *satisfies* (11), *then*

$$(|E_{S,i} \setminus S| + 1)\, x_i + \sum_{j \in E_{S,i} \setminus \{i\}} x_j \leq |E_{S,i}| - 1 \tag{12}$$

is valid for (5).

Proof. Suppose $\widehat{x}$ is feasible for (5). First, considering the case that $\widehat{x}_i = 0$, then

$$\sum_{j \in E_{S,i} \setminus \{i\}} \widehat{x}_j \leq |E_{S,i}| - 1 = |E_{S,i}| - 1 - (|E_{S,i} \setminus S| + 1)\widehat{x}_i.$$

Otherwise, consider the case that $\widehat{x}_i = 1$. Assume for the sake of deriving a contradiction that

$$\sum_{j \in E_{S,i} \setminus \{i\}} \widehat{x}_j = \sum_{j \in S \setminus \{i\}} \widehat{x}_j + \sum_{j \in E_{S,i} \setminus S} \widehat{x}_j > |E_{S,i}| - 1 - (|E_{S,i} \setminus S| + 1)\widehat{x}_i = |S| - 2.$$

Note that the integrality of $\widehat{x}$ implies that $\sum_{j \in E_{S,i} \setminus \{i\}} \widehat{x}_j \geq |S| - 1$. Hence $|\mathcal{S}(\widehat{x}) \cap E_{S,i}| \geq |S|$. Then, by the definition of $E_{S,i}$, and (11), it follows that

$$\begin{aligned} g_i(\widehat{x}) - M_i = {} & \sum_{j \in S \setminus \{i\}} q_{ij}\widehat{x}_j - \sum_{\substack{\{j,k\} \in \mathcal{T}_i: \\ j,k \in S}} r_{ijk}\widehat{x}_j\widehat{x}_k + \\ & \sum_{j \in X_{S,i}} q_{ij}\widehat{x}_j - \sum_{\substack{\{j,k\} \in \mathcal{T}_i \\ j \in X_{S,i} \text{ or } k \in X_{S,i}}} r_{ijk}\widehat{x}_j\widehat{x}_k \geq \sum_{j \in S \setminus \{i\}} q_{ij}\widehat{x}_j - \sum_{\substack{\{j,k\} \in \mathcal{T}_i: \\ j,k \in S}} r_{ijk}\widehat{x}_j\widehat{x}_k + \max_{j \in S}\{q_{ij}\} \sum_{k \in E_{S,i} \setminus S} \widehat{x}_k. \end{aligned}$$

Then, since $|\mathcal{S}(\widehat{x}) \cap E_{S,i}| \geq |S|$, by a pigeonhole argument and since (5b) is violated by x'

$$g_i(\widehat{x}) - M_i \geq \sum_{j \in S \setminus \{i\}} q_{ij}x'_j - \sum_{\{j,k\} \in \mathcal{T}_i} r_{ijk}x'_jx'_k > T_i,$$

which contradicts the feasibility of $\widehat{x}$. $\square$

3.3 Separation of Valid Inequalities

To apply Proposition 5 we first need to determine a subset S that violates (5b) for some $i \in V$. Note that, for a given candidate $S \subseteq V$, it is straightforward to verify condition (8). Then, using S that satisfies (8) we extend it to a set $E_{S,i} \supseteq S$.

First, we motivate a greedy heuristic for determining $S \subseteq V$ starting from $S = \{i\}$ for a given $i \in V$. If one presupposes that the condition (8) is satisfied for i then the separation problem for determining S, given a linear relaxation solution x^*, can be simplified and formulated as a quadratic knapsack problem

$$\begin{aligned}
&\underset{z}{\text{maximize}} && g_i(z) && && \text{(13a)}\\
&\text{subject to} && \sum_{j \in N(i)} (1 - x_j^*) z_j \leq 1 - \epsilon && && \text{(13b)}\\
& && z_j = 0 && j \in V \setminus (N(i) \cup \{i\}) && \text{(13c)}\\
& && z \in \{0,1\}^n, && && \text{(13d)}
\end{aligned}$$

where $\epsilon \in \left(0, \min_{j \in N(i)} \left\{ x_j^* \;\middle|\; x_j^* > 0 \right\}\right)$. Note that constraint (13b) ensures that x^* violates the cover inequality (6) with S that is defined as the support set of z. We note that the function $g_i : \{0,1\}^n \to \mathbb{R}$, the objective (13a), is a submodular quadratic function of binary variables. This observation motivates a greedy heuristic for the separation problem. For the special case with monotone $g_i(z)$ (for example, if $q_{ij} = p \leq \frac{1}{|N(i)|}$ for all $j \in N(i)$, as in the proof of Proposition 2), and equal coefficients $(1 - x_j^*)$ for all $j \in N(i)$, a greedy algorithm is known to have an $(e-1)/e$ approximation factor guarantee, where e is the base of the natural logarithm [25]. For a nonmonotone submodular objective function, and with general knapsack constraints, there are more computationally intensive extensions that yield constant factor approximations; see, respectively, [27] and [22]. However, due to the computational advantage and ease of implementation, here we only apply a simple greedy algorithm.

Let $x(S) \in \{0,1\}^{|V|}$ denote the characteristic vector of a set $S \subseteq V$. For a given function $\phi : 2^V \times V \to \mathbb{R}$, and $i \in V$, we consider Algorithm 1 as a greedy procedure for computing an extended covering set. These sets are then used to generate valid inequalities for the linearization of (5). Algorithm 1 is run for $i \in V$ whose corresponding constraint (5b) is tight in the linear relaxation solution x^*; in Section 4 we invoke this algorithm for $i \in V$ with the largest dual multiplier value for constraint (5b). We consider two different variants of Algorithm 1, each corresponding to a different definition of the function ϕ in line 7 of the algorithm. In particular, we consider either

$$\phi(\cdot, j) = x_j^*, \qquad \text{or} \tag{14a}$$

$$\phi(S, j) = \frac{g_i(x(S \cup \{j\})) - g_i(x(S))}{1 - x_i^*}; \tag{14b}$$

these are defined analogously to the two common approaches for greedily constructing knapsack covers.

In the context of the knapsack problem, extended cover inequalities are effectively generated in practice by greedily constructing S to consist of $j \in V$ for which x_j^* is largest (where x^* is an

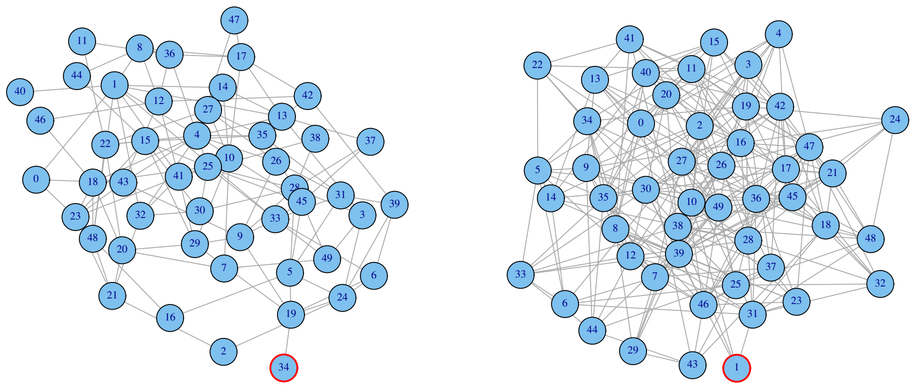
Algorithm 1 An algorithm for computing an extended covering set $E_{S,i} \subseteq V$.

```
1: Input: an optimal solution x* of the relaxation of (5), i ∈ V
2: Q_1 ← {i}
3: for k = 1, ..., n do
4:    if (5b) is violated with x replaced by x(Q_k), and (8) holds then
5:       break
6:    end if
7:    Pick some j* ∈ argmax_{i∈N(i)} φ(Q_k, j)
8:    Q_{k+1} ← Q_k ∪ {j*}
9: end for
10: S ← Q_k
11: X_{S,i} = ∅
12: while Φ ≡ argmax_{j∈N(i)\(X_{S,i}∪S)} {g_i(x(S ∪ {j})) | (11)} ≠ ∅ do
13:    Pick some j* ∈ Φ
14:    X_{S,i} ← X_{S,i} ∪ {j*}
15: end while
16: Output: E_{S,i} = S ∪ X_{S,i}
```

optimal solution of the relaxation); see for example [15]. This approach is similar in to our choice of ϕ defined by (14a) in Algorithm 1. The choice of ϕ that is defined by (14b) within Algorithm 1 corresponds to a greedy heuristic for solving (13), and in the special case that g_i is linear it is similar to a heuristic proposed for separating knapsack cover inequalities in [6]. Given the cover S that is computed by Algorithm 1 in steps 2 – 10, it is then iteratively extended; starting with $E_{S,i} = S$, at each iteration we insert into $E_{S,i}$ an element $j^* \in N(i)$ that maximizes $g_i(x(E_{S,i} \cup \{j\}))$ over all $j \in N(i) \setminus E_{S,i}$ satisfying (11). Note that upon termination of Algorithm 1, assuming that the continuous relaxation solution x^* is non-integer, we are guaranteed to get a valid cover inequality; however, the resulting cover inequality does not necessarily cut off x^*. Thus, in the experiments of Section 4 it is verified upon termination that the resulting cover inequalities are violated before appending them to (5).

4 Computational Results

In this section we show computational results for the linearization of (5) using the methods described in Section 3. We experiment with networks that are generated randomly by using the Erdős-Rényi random graph model (for details the reader may refer to [8]), with different graph density values: $0.1, 0.2$, and 0.25. Examples of graphs generated with 50 nodes are shown in Figure 1. For each network we randomly generated ten different sets of input parameter values for W, π, p, and T. A set of compromised nodes $V_c \subseteq V$ is chosen at random so that $|V_c| = \lceil 0.1\,|V| \rceil$, for which $\pi_i = 1$ for each $i \in V_c$. Otherwise, for $i \in V \setminus V_c$, π_i is generated uniformly at random from $(0, 0.8)$. For each $\{i, j\} \in E$, W_{ij} is generated uniformly at random from $(0, 1)$. For each



(a) An example network with 0.1 density.

(b) An example network with 0.2 density.

Figure 1: Randomly generated Erdős-Renýi graphs with 50 nodes that are used in the computational study of Section 4.

$\{i, j\} \in E$, the probability of the infection propagating from node j to i is given by

$$p_{ij} = \frac{W_{ij}}{\sum_{k \in N(j):(k,j) \in E} W_{kj}},$$

similar to the choice of parameter values in the experiments of Altunay et al. [1].

We run the linearized formulation of (5) using the open-source CBC [24] solver (trunk version, revision #2008). Also, our cut generation routines use the open-source network algorithm LEMON [11] library. The experiments are run in serial on a machine with Intel Xeon 2.66 GHz CPUs with a cache size of 6 MB, and a 64-bit Ubuntu 12.04 Linux operating system. CBC was used in its default setting[1] unless otherwise indicated. In its default setting CBC invokes a variety of MIP cuts and heuristics based on whether they are found effective at reducing the optimality gap.

Let u_{relax} denote the optimal objective value of the relaxation of (5) upper bound, let u_{cuts} be the optimal objective value of (5) with cuts (inequalities (6) or other cuts as indicated otherwise), and let $u_{\text{lb,relax}}$, and $u_{\text{lb,cuts}}$, be the lower bounds computed by a feasible solution heuristic in the runs without and with cuts, respectively. To evaluate the cuts, we consider the performance indicators of the optimality gap closed and bound reduction,

$$1 - \frac{u_{\text{cuts}} - u_{\text{lb,cuts}}}{u_{\text{relax}} - u_{\text{lb,relax}}} \qquad \text{and} \qquad 1 - \frac{u_{\text{cuts}}}{u_{\text{relax}}},$$

respectively. Tables 1-3 show the optimality gap closed as well as bound reduction at the root node for each setting of cuts that is indicated. The data of each row in the table is based on 100 runs with ten different graphs and ten different sets of input parameter values for each graph. The data of the

[1]Due to implementation issues and missing library support for mapping variables following their elimination by preprocessing we had CBC preprocessing switched off. Note that the random instances that we consider seldom have redundancies, and accordingly preprocessing did not seem to significantly reduce the size of these problems (and their running times).

table suggests that generating a small number (a few tens) of cuts (12) can tighten the formulation and reduce the relaxation upper bound by as much as 11%. The greatest tightening of the root relaxation is achieved by cuts (12) together with the CBC cuts, in which case the initial relaxation upper bound is reduced by up to 50%. The cuts generated by greedily maximizing (14b) appear to be more effective in tightening the formulation than those generated by greedily maximizing (14a). While in denser graphs the tightening of the LP relaxation becomes more modest, this applies also to other MIP cuts that are generated. Also note that the number of cuts that are generated at the root (as well as branch-and-bound nodes) in practice may depend on internal solver policies that are implemented in the CBC solver. An effective generation of a diverse set of cuts (12) and careful fine-tuning for a particular solver may warrant further computational studies.

Tables 4–6 show experimental results for the entire branch-and-cut algorithm with Algorithm 1 and ϕ defined by (14b). Experiments with cuts generated using ϕ defined by (14a) are omitted since they appear to be dominated by cuts generated with ϕ defined by (14b) in the experiments of Tables 1-3 (while not requiring a substantial increase in the running time of evaluating ϕ). The cuts and cover methods indicated in the tables are run in addition to the CBC defaults. Each row of the table corresponds to 50 runs – ten different graphs and five different sets of input parameters for each graph. Also, we set CBC to emphasize generating knapsack cover inequalities (in all runs). Otherwise, the generation of our cuts at the root node seemed to be causing CBC to disable the generation of standard knapsack cover inequalities later, which, in some cases, seemed to degrade the overall performance. We set the time limit in all runs to two hours. "LIMIT" is used to indicate that the data is not displayed due to the time limit being reached. Branch-and-bound node and CPU time averages are calculated over all of the runs. We initialized CBC to invoke our cover and cut generation routine (Algorithm 1) once in every 50 nodes, with automatic adjustment of the CBC solver based on the effectiveness of the cuts that are generated. The algorithm is invoked for $i \in V$ whose constraint (5b) is among the 60 with largest dual multiplier values at the root node, and among the two largest dual multiplier values at other branch-and-bound nodes.

Tables 4 – 6 show that in most cases the valid inequalities (12), whose covering set is computed using (14b), improved the performance of the CBC solver. In Table 4 the improvement was also evident in fewer runs that exceeded the time limit (of two hours).

5 Conclusion

We propose a novel probabilistic formulation for responding to cyber and biological threats in networks. Optimization problems with a similar objective were considered, for example by [1], but with a different set of constraints that lack the motivation of a probabilistic framework that is considered herein. In contrast with the probabilistic analytic framework considered in [10] for the spread of cyber threats in networks, the proposed model does not rely on a restrictive assumption of independence of the infection events. To solve the optimization problem, we consider a quadratic formulation that bounds the infection probabilities using Hunter's bound and thus determines a more conservative solution. The advantage of this approach is twofold. First, the quadratic formulation can be easily linearized and solved by standard MIP solvers. Second, the

Table 1: Computational results showing the optimality gap closed and bound reduction at the root node with random graphs of 0.1 density. Here + indicates CBC defaults w/o preprocessing, and - signifies a run without invoking any CBC cuts; $(\cdot)$ indicates the use of the corresponding feature and equation number, as defined in this paper. Each row displays the statistics of 100 runs with ten different graphs and ten using each graph but with different values for the input parameters W, π, P, and T.

Nodes	**Edges**	**Cuts/Cover**	**Gap Closed**			**Bound Reduction**			**Cut Number**		
	Avg		Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
		+	0.374	0.096	0.866	0.491	0.286	0.640	5591.9	3266	6327
		-,(14a)	0.028	-0.037	0.200	0.045	0.003	0.093	16.5	4	27
50	121.3	+,(14a)	0.372	0.096	0.841	0.494	0.282	0.640	5571.6	4284	6266
		-,(14b)	0.063	-0.014	0.291	0.111	0.070	0.170	31.6	24	42
		+,(14b)	**0.388**	0.118	0.821	**0.500**	0.305	0.636	5490.9	3296	6228
		+	0.241	0.069	0.545	0.442	0.283	0.611	6800.1	5961	7404
		-,(14a)	0.011	-0.011	0.107	0.030	0.000	0.096	14.6	4	27
60	177.6	+,(14a)	0.241	0.065	0.477	0.446	0.279	0.629	6776.9	3620	7652
		-,(14b)	0.046	0.000	0.212	0.097	0.044	0.151	36.9	26	51
		+,(14b)	**0.249**	0.055	0.568	**0.450**	0.297	0.620	6736.4	5672	7254
		+	0.163	0.065	0.363	0.423	0.268	0.532	5930.6	3774	7909
		-,(14a)	0.008	-0.023	0.059	0.023	0.000	0.070	11.6	0	26
70	244.0	+,(14a)	0.166	0.064	0.400	0.425	0.272	0.543	5887.5	3224	7953
		-,(14b)	0.033	0.000	0.105	0.085	0.052	0.119	41.1	30	53
		+,(14b)	**0.173**	0.082	0.475	**0.430**	0.272	0.557	5913.1	3050	8053
		+	0.122	0.043	0.276	0.378	0.283	0.494	5739.8	3284	7599
		-,(14a)	0.004	0.000	0.033	0.016	0.000	0.048	10.0	0	21
80	315.9	+,(14a)	0.126	0.043	0.308	0.380	0.285	0.503	5814.4	3063	7730
		-,(14b)	0.027	0.000	0.102	0.078	0.048	0.113	45.7	32	59
		+,(14b)	**0.129**	0.052	0.295	**0.385**	0.297	0.480	5600.0	3224	7506
		+	**0.101**	0.043	0.237	0.365	0.283	0.451	5882.1	3316	7715
		-,(14a)	0.003	0.000	0.033	0.014	0.000	0.049	8.3	0	24
90	394.2	+,(14a)	0.097	0.043	0.233	0.366	0.292	0.447	5816.6	3428	7364
		-,(14b)	0.019	0.000	0.090	0.071	0.054	0.095	49.7	39	61
		+,(14b)	**0.101**	0.043	0.247	**0.372**	0.292	0.458	5938.7	4043	7848

Table 2: Computational results showing the optimality gap closed and bound reduction at the root node with random graphs of 0.2 density. Here + indicates CBC defaults w/o preprocessing, and - signifies a run without invoking any CBC cuts; $(\cdot)$ indicates the use of the corresponding feature and equation number, as defined in this paper. Each row displays the statistics of 100 runs with ten different graphs and ten using each graph but with different values for the input parameters W, π, P, and T.

Nodes	**Edges**	**Cuts/Cover**	**Gap Closed**			**Bound Reduction**			**Cut Number**		
	Avg		Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
		+	0.093	0.021	0.218	0.365	0.260	0.489	5177.6	2543	6494
		-,(14a)	0.002	-0.011	0.034	0.008	0.000	0.037	3.6	0	12
50	249.7	+,(14a)	0.094	0.021	0.218	0.367	0.264	0.497	5097.7	2552	6403
		-,(14b)	0.018	0.000	0.069	0.062	0.035	0.092	27.4	15	41
		+,(14b)	**0.095**	0.032	0.207	**0.372**	0.249	0.506	4934.7	2141	6495
		+	0.065	0.010	0.196	0.334	0.242	0.466	5502.4	2880	7052
		-,(14a)	0.002	0.000	0.023	0.008	0.000	0.037	3.9	0	21
60	350.7	+,(14a)	0.065	0.021	0.196	0.332	0.240	0.446	5284.7	2882	6702
		-,(14b)	0.014	0.000	0.120	0.056	0.028	0.084	32.3	21	47
		+,(14b)	**0.069**	0.021	0.185	**0.339**	0.249	0.480	5467.9	2804	6694
		+	0.055	0.021	0.121	0.315	0.262	0.387	5944.3	3779	7426
		-,(14a)	0.001	0.000	0.031	0.005	0.000	0.023	2.9	0	11
70	480.9	+,(14a)	0.055	0.021	0.120	0.313	0.262	0.380	5845.4	3779	7427
		-,(14b)	0.009	0.000	0.055	0.047	0.030	0.065	37.0	25	56
		+,(14b)	**0.056**	0.021	0.135	**0.317**	0.247	0.385	5816.3	2851	7250
		+	0.041	0.010	0.088	0.292	0.215	0.378	6359.2	3616	8833
		-,(14a)	0.000	0.000	0.011	0.004	0.000	0.024	2.7	0	24
80	619.5	+,(14a)	0.041	0.010	0.130	0.290	0.215	0.378	6282.5	3286	8461
		-,(14b)	0.006	0.000	0.033	0.043	0.023	0.060	42.3	28	56
		+,(14b)	**0.043**	0.010	0.109	**0.295**	0.211	0.362	6219.3	3900	8830
		+	0.034	0.010	0.076	0.260	0.200	0.312	6464.2	3709	9281
		-,(14a)	0.001	0.000	0.011	0.003	0.000	0.017	2.7	0	19
90	796.6	+,(14a)	0.033	0.010	0.076	0.258	0.207	0.313	6370.3	3231	9764
		-,(14b)	0.005	0.000	0.033	0.036	0.022	0.056	46.0	32	61
		+,(14b)	**0.035**	0.010	0.087	**0.263**	0.207	0.328	6413.8	4152	9017

Table 3: Computational results showing the optimality gap closed and bound reduction at the root node with random graphs of 0.25 density. Here + indicates CBC defaults w/o preprocessing, and - signifies a run without invoking any CBC cuts; $(\cdot)$ indicates the use of the corresponding feature and equation number, as defined in this paper. Each row displays the statistics of 100 runs with ten different graphs and ten using each graph but with different values for the input parameters W, π, P, and T.

Nodes	**Edges**	**Cuts/Cover**	**Gap Closed**			**Bound Reduction**			**Cut Number**		
			Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
50	305.4	+	0.068	0.011	0.222	0.337	0.250	0.427	5076.7	2593	6291
		-,(14a)	0.001	0.000	0.021	0.006	0.000	0.035	2.7	0	18
		+,(14a)	0.069	0.011	0.188	0.337	0.231	0.418	5103.6	2533	6431
		-,(14b)	0.014	0.000	0.106	0.051	0.028	0.076	26.6	19	37
		+,(14b)	**0.075**	0.010	0.267	**0.342**	0.247	0.431	5027.3	2460	6350
60	454.9	+	**0.049**	0.010	0.120	0.304	0.235	0.445	5559.1	2608	6831
		-,(14a)	0.000	0.000	0.012	0.003	0.000	0.025	1.8	0	14
		+,(14a)	0.048	0.010	0.125	0.302	0.221	0.441	5389.0	3101	6852
		-,(14b)	0.007	-0.010	0.034	0.043	0.022	0.066	31.1	20	43
		+,(14b)	**0.049**	0.010	0.135	**0.306**	0.237	0.446	5424.3	2416	7322
70	594.9	+	0.038	0.010	0.097	0.285	0.216	0.365	5919.6	2986	7929
		-,(14a)	0.001	0.000	0.011	0.003	0.000	0.020	2.1	0	12
		+,(14a)	0.036	0.010	0.108	0.282	0.207	0.351	5825.7	2986	7933
		-,(14b)	0.007	0.000	0.076	0.039	0.017	0.055	36.5	23	50
		+,(14b)	**0.041**	0.010	0.128	**0.290**	0.219	0.370	5855.2	2634	7458
80	800.9	+	0.027	0.010	0.065	**0.255**	0.206	0.303	6216.7	3202	8784
		-,(14a)	0.000	0.000	0.011	0.002	0.000	0.016	2.0	0	12
		+,(14a)	0.027	0.010	0.076	0.250	0.195	0.302	6096.1	3511	8650
		-,(14b)	0.003	0.000	0.022	0.033	0.021	0.046	41.6	28	55
		+,(14b)	**0.028**	0.010	0.059	**0.255**	0.204	0.306	6014.1	3029	9203
90	994.6	+	0.025	0.010	0.077	**0.245**	0.204	0.301	6678.9	3813	9668
		-,(14a)	0.001	0.000	0.011	0.002	0.000	0.014	2.7	0	19
		+,(14a)	0.024	0.010	0.076	0.238	0.190	0.297	6424.5	3118	9647
		-,(14b)	0.003	0.000	0.033	0.028	0.018	0.037	48.4	34	66
		+,(14b)	**0.026**	0.010	0.114	0.244	0.187	0.299	6538.8	2425	9791

Table 4: Computational results with random graphs of 0.10 density. Here + indicates CBC defaults w/o preprocessing; $(\cdot)$ indicates the use of the corresponding feature and equation number, as defined in this paper. Each row displays the statistics of 100 runs with ten different graphs and ten using each graph but with different values for the input parameters W, π, P, and T.

Nodes	Edges	Cuts/Cover	B&B Nodes			CPU sec			# over
			Avg	Min	Max	Avg	Min	Max	Limit
50	123.4	+	110.6	10	381	11.0	4.1	24.4	0
		+,(14b)	**92.1**	10	604	**10.5**	4.4	32.4	0
60	180.3	+	**823.9**	68	4902	45.3	16.8	169.2	0
		+,(14b)	848.4	54	3895	**45.0**	15.3	145.3	0
70	236.1	+	4311.3	170	22266	185.4	23.4	716.9	0
		+,(14b)	**3225.8**	206	15203	**148.4**	31.7	510.1	0
80	310.8	+	**14830.6**	684	52501	883.8	83.7	3638.9	0
		+,(14b)	15664.0	728	77422	**875.3**	75.8	3890.9	0
90	396.2	+	72787.6	24719	135896	5799.3	1863.8	LIMIT	26
		+,(14b)	**68326.7**	15284	124864	**5572.2**	1395.9	LIMIT	25

Table 5: Computational results with random graphs of 0.20 density. Here + indicates CBC defaults w/o preprocessing; $(\cdot)$ indicates the use of the corresponding feature and equation number, as defined in this paper. Each row displays the statistics of 100 runs with ten different graphs and ten using each graph but with different values for the input parameters W, π, P and T.

Nodes	Edges	Cuts/Cover	Nodes			CPU sec			# over
			Avg	Min	Max	Avg	Min	Max	Limit
50	256	+	3227.2	319	11302	129.9	25.4	439.4	0
		+,(14b)	3082.7	466	13638	**122.2**	29.3	354.2	0
60	328	+	**17284.0**	4594	47602	**890.8**	295.4	2097.3	0
		+,(14b)	17351.5	4169	62974	903.5	238.8	3257.5	0
70	471	+	65210.7	17338	112893	5399.8	1662.6	LIMIT	14
		+,(14b)	**62406.2**	19742	114868	**5072.4**	1669.3	LIMIT	14

Table 6: Computational results with random graphs of 0.25 density. Here, + indicates CBC defaults w/o preprocessing; $(\cdot)$ indicates the use of the corresponding feature and equation number as defined in this paper. Each row displays the statistics of 100 runs with ten different graphs and ten using each graph but with different values for the input parameters W, π, P and T.

Nodes	**Edges**	**Cuts/Cover**	**Nodes**			**CPU sec**			**# over**
			Avg	Min	Max	Avg	Min	Max	limit
50	301.9	+	5967.1	790	16168	266.4	68.8	696.7	0
		+,(14b)	**5741.7**	694	14280	**247.3**	53.9	693.3	0
60	447.3	+	30775.0	7668	101299	**1845.4**	389.6	7123.1	0
		+,(14b)	**29767.2**	5096	116839	1855.2	346.3	6634.9	0

application of the quadratic probability bound, and thus the resulting formulation, need not require stochastic independence.

To improve on the solution time of standard solvers, we develop novel cover inequalities as cutting planes for the quadratic formulation and its linearized counterpart. We also show stronger extended-cover inequalities to be valid for our problem. While knapsack-cover inequalities are derived for constraints whose left-hand side is monotone and convex, these properties do not hold in our case (i.e., convexity does not hold before our constraints are linearized). We prove the validity of our inequalities upon verifying a simple, testable condition. This condition is verified in runtime before appending the inequality as a cutting plane in the course of the branch-and-cut algorithm. In nearly all computational experiments, generating a small number of the proposed cuts results in modest and in some cases substantial improvements in terms of tightening the root relaxation bound, overall branch-and-bound CPU time, and branch-and-bound nodes. In future work we will further investigate the tightening and generation of a diverse set of our extended-cover inequalities. Further, we will consider extensions and simulation experiments with this work in a dynamic setting when a network response problem is solved repeatedly as an epidemic is spreading through the network. We will consider multilevel methods to scale the solution approach for solving larger instances.

A Analysis of the Deterministic Model

For $i \in V$, let $T_i^0 \in [0,1)$ be the level of threat of node i if it is isolated (i.e., if none of its neighbors are open). A nonlinear model based on [1] for network response is

$$\begin{aligned}
&\underset{x,t}{\text{maximize}} && u(x) && && \text{(15a)}\\
&\text{subject to} && t_i = T_i^0 x_i + \sum_{j \in N(i)} p_{ij} t_j x_i x_j && i \in V_u && \text{(15b)}\\
& && 0 \le t_i \le T_i && i \in V_u && \text{(15c)}\\
& && t_i = T_i = 1 && i \in V_c && \text{(15d)}\\
& && x \in \{0,1\}^n. && && \text{(15e)}
\end{aligned}$$

The formulation (15) relaxes a set of additional constraints that are present in the site model of Altunay et al. [1] in order to discourage shutting down uncompromised sites. Instead we may add a term $-C \sum_{i \in V_c} x_i$ in (15a); here, $C \ge 0$ is a penalty for compromised sites that remain open. For small values of C the parameter serves as a tiebreaker among different optimal solutions, favoring those that shut down more compromised sites.

Altunay et al. [1] formulate their problem as a mixed-integer nonlinear program (MINLP) similar to (15) and show that it can be solved as a MIP through a particular linearization scheme. This solution approach is well justified for problems that are recognized to be computationally hard (otherwise, more efficient and scalable solution methods may exist). We now show that their network response problem (more precisely, its relaxation (15)), is $\mathcal{NP}$-hard.

Proposition 6. *The problem* (15) *with* $W_{ij} = 1$ *for every* $\{i,j\} \in E$ *is* $\mathcal{NP}$*-hard.*

Proof. The proof follows by reduction of the maximum independent set problem. Let $\bar{G} = (\bar{V}, \bar{E})$ with $\bar{V} = \{1, \ldots, \ell\}$ denote an input graph of the independent set problem. Let $V = \{1, \ldots, 2\ell\}$ and $E = \bar{E} \cup \{\{i, \ell + i\} \mid i = 1, \ldots, \ell\}$. For an (arbitrary) constant $\epsilon > 0$ set $T_i^0 = T_i = \epsilon$ for all $i \in V$, and let

$$p_{ij} = \begin{cases} 0 & \{i,j\} \in E \setminus \bar{E}, \\ \epsilon & \text{otherwise}, \end{cases}$$

and let $W_{ij} = 1$ for all $\{i,j\} \in E$. Evidently, since $T_i^0 = T_i > 0$ and $p_{ij} > 0$ for all $\{i,j\} \in \bar{E}$, every solution (x,t) that is feasible for (15) satisfies $x_i x_j = 0$ for all $\{i,j\} \in \bar{E}$ and thus corresponds to an independent set in the graph $\bar{G}$. Conversely, given an independent set $S \subseteq \bar{V}$, its (extended) characteristic vector $x \in \{0,1\}^n$ with $x_k = 0$ for all $k \in \bar{V} \setminus S$, must satisfy $x_i x_j = 0$ for all $\{i,j\} \in \bar{E}$, and hence $(x, \epsilon x)$ is feasible for (15). For a solution (x^*, t^*) that is optimal to (15), it follows that

$$\sum_{\{i,j\} \in E} W_{ij} x_i^* x_j^* = \sum_{\substack{\{i,j\} \in E: \\ i \in V \setminus \bar{V}, j \in \bar{V}}} x_i^* x_j^* = \sum_{i \in \bar{V}} x_i^* x_{i+\ell}^* = \sum_{i \in \bar{V}} x_i^*. \tag{16}$$

The last equality followed from optimality; $p_{i,i+\ell} = 0$, for all $i \in \bar{V}$, implies the feasibility of

$(x,t) \in \{0,1\}^n \times [0,1]^n$ defined by

$$x_j = \begin{cases} x_j^* & j \in \bar{V}, \\ 1 & j \in V \setminus \bar{V}, \end{cases} \qquad \text{and} \qquad t_j = \begin{cases} x_j^* \epsilon & j \in \bar{V} \\ 1 & j \in V \setminus \bar{V}, \end{cases}$$

for (15). By (16) and the correspondence of feasible solutions of (15) and independent sets, it follows that the optimal solution of (15) in G yields a maximum independent set in $\bar{G}$. □

B Proof of the Proposition 1

The proof follows by a similar reduction of the maximum independent set problem in the input graph $\bar{G} = (\bar{V}, \bar{E})$, as in the proof of Proposition 6; specifically, let G and W be defined as in that proof. For arbitrary constants (independent of n) $\epsilon, \delta \in (0,1)$ with $0 < \epsilon < \delta < 1$, let $T_i = \epsilon$ for $i \in \bar{V}$, and let

$$p_{ij} = \begin{cases} \delta & \{i,j\} \in \bar{E} \\ 0 & \{i,j\} \in E \setminus \bar{E} \end{cases}.$$

It follows that every solution x that is feasible for (3) has $x_i x_j = 0$, for every $\{i,j\} \in E$; otherwise, by the fact that the left-hand side of (3b) is increasing in x,

$$x_i - \prod_{k \in N(i)} (1 - \delta x_k) \geq 1 - (1 - \delta) = \delta > \epsilon = T_i,$$

a contradiction to the feasibility of x. Conversely, for an independent set $S \subset \bar{V}$, if $x_i = 0$ for every $i \in \bar{V} \setminus S$, then x is feasible for (3). Further, the fact that $q_{ij} = 0$ for $\{i,j\} \in E \setminus \bar{E}$ implies that (16) holds. Therefore, an optimal solution of (3) yields a solution that is optimal to the maximum independent set problem. □

C Proof of Proposition 2

The proof follows by a similar reduction of a maximum independent set problem in the input graph $\bar{G}$ as in the proof of Proposition 1. Let $G = (V, E)$, W, q, and T be as defined as in that proof, but choosing $\delta, \epsilon > 0$ so that $\delta = 1/n > \epsilon$. Evidently δ and also the ϵ can be selected so that both are represented using at most an $O(\log n)$ number of bits. Further, for all $i \in V$ and $j, k \in N(i)$ let $r_{ijk} = q_{ij} q_{ik}$. It follows that for all $x \in \{0,1\}^n$ with $x_j = 0$, for some $j \in V$,

$$g_i(x + e^j) - g_i(x) = q_{ij} \left(1 - \sum_{\substack{k \in N(i): \\ \{j,k\} \in \mathcal{T}_i}} q_{ik} x_k \right) \geq \frac{1}{n} - N(i) \left(\frac{1}{n^2} \right) \geq \frac{1}{n} - \frac{n-1}{n^2} > 0.$$

Now, assume x' is feasible for (3). Then, $x'_i x'_j = 1$ for some $\{i,j\} \in \bar{E}$ implies that

$$g_i(x') \geq M_i + q_{ij} = M_i + \delta > \epsilon + M_i = T_i + M_i,$$

a contradiction. Therefore $x'_i x'_j = 0$ for all $\{i, j\} \in \bar{E}$. So, applying the converse argument as in the proof of Proposition 1, it follows that a solution x' is feasible for (5) if an only if $\left\{ i \in \bar{V} \mid x'_i = 1 \right\}$ is an independent set. Then, it follows from (16) that each solution x^* that is optimal for (5) yields a maximum independent set in $\bar{G}$. $\square$

D Comparison of probability bounds under stochastic independence

In the following we empirically compare (3) with two formulations: the first-order (union bound) formulation, with all bilinear terms in (5) satisfying $r_{ijk} = 0$, and formulation (5) with $r_{ijk} = q_{ij} = q_{ik}$. To evaluate each, we consider the relative error, defined as

$$\frac{u(x^{(3)}) - u(x^{(5)})}{u(x^{(3)})},$$

where $x^{(\cdot)}$ is an optimal solution of the corresponding formulation $(\cdot)$. The computational experiments are run using the state-of-the art CPLEX solver, version 12.3, to compute (the linearized version of) (5) with $r_{ijk} = 0$ and (5) with $r_{ijk} = q_{ij} q_{ik}$ for all $i \in V$, and $\{j, k\} \in \mathcal{T}_i$. To compute (3), we use the open-source state-of-the-art MINLP solver BARON, Version 9.3.1.

The results of the experiments are shown in Table 7. Each row displays an average of 10 runs, each with a randomly generated subset of $V_c \subseteq V$ of the indicated cardinality having $\pi_j = 1$ for all $j \in V_c$. For each $i \in V \setminus V_c$, π_i is generated uniformly at random from $(0, 0.8)$.

Table 7: Error rates and running times over ten runs with the aggregated Open Science Grid example network, OSG-1 appearing in [1]; each run corresponds to a random subset V_c of the indicated cardinality, and which contains all nodes i with $\pi_i = 1$.

$\lvert\mathbf{V_c}\rvert$	Runs of (5) **with** $r = \mathbf{0}$ **(union bound)**				**Runs of** (5) (Hunter's bound)				**Runs of** (3)	
	Avg Err	Max Err	Avg CPU	Max CPU	Avg Err	Max Err	Avg CPU	Max CPU	Avg CPU	Max CPU
1	0.10	0.15	0.21	0.31	0.07	0.13	0.37	0.43	0.76	1.29
2	0.08	0.14	0.21	0.27	0.02	0.07	0.45	0.59	0.66	1.55
3	0.06	0.14	0.20	0.28	0.01	0.06	0.51	0.62	1.12	4.10
4	0.09	0.16	0.22	0.29	0.02	0.06	0.52	0.81	0.72	2.96
5	0.12	0.16	0.28	0.49	0.01	0.05	0.58	0.69	0.80	2.32
6	0.12	0.17	0.26	0.35	0.03	0.07	0.77	2.14	0.84	2.90
7	0.13	0.20	0.26	0.41	0.02	0.06	0.86	2.38	1.24	4.76
8	0.14	0.17	0.28	0.50	0.01	0.09	0.65	0.97	0.74	2.19
9	0.15	0.19	0.38	0.64	0.01	0.05	0.85	1.43	1.61	5.49
10	0.15	0.21	0.33	0.50	0.02	0.10	0.92	1.75	1.09	1.55

Table 7 shows that the relative error of the formulation (5) with $g_i(x)$ having bilinear terms is substantially smaller than the relative error of the formulation involving only linear terms in $g_i(x)$. Further, the error does not significantly grow as $|V_c|$ increases in contrast to the formulation involving only linear terms. The running times of the exact (assuming the independence of infection probabilities) nonlinear formulation (3) are about twice as much as that of the quadratic approximation (5). However, the full product formulation (3) becomes increasingly difficult to solve: the running times of (5) with $|V| = 60$ and $|E| = 328$ are on the order of a few minutes, while the full product form (3) could not be solved to optimality within two hours. Thus, we can use the quadratic formulation (5) as a suitable surrogate for the nonconvex multilinear formulation (3), which is much harder to solve. Moreover, the quality of the quadratic approximation significantly improves on the quality of the linear approximation.

References

[1] M. Altunay, S. Leyffer, J.T. Linderoth, and Z. Xie. Optimal security response to attacks on open science grids. *Computer Networks*, 55(1):61–73, 2011.

[2] E. Anshelevich, D. Chakrabarty, A. Hate, and C. Swamy. Approximability of the firefighter problem. *Algorithmica*, 62(1-2):520–536, 2012.

[3] A. Atamtürk, L.F. Muller, and D. Pisinger. Separation and extension of cover inequalities for second-order conic knapsack constraints with GUBs. *INFORMS Journal on Computing*, 25(3):420–431, 2013.

[4] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8(1):146–164, 1975.

[5] E. Balas and E. Zemel. Facets of the knapsack polytope from minimial covers. *SIAM Journal on Applied Mathematics*, 34(1):119–148, 1978.

[6] T. Bektas and O. Oğuz. On separating cover inequalities for the multidimensional knapsack problem. *Computers & Operations Research*, 34:1771–1776, 2007.

[7] F. Boesch, A. Satyanarayana, and C. Suffel. On residual connectedness network reliability. In F. Roberts, F. Hwang, and C. Monma, editors, *Reliability of Computer and Communication Networks*. DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, 1991.

[8] B. Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[9] J. Bukszár and A. Prékopa. Probability bounds with cherry trees. *Mathematics of Operations Research*, 26:174–192, 2001.

[10] D. Chakabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Securitys*, 10, 2008.

[11] B. Dezsö, A. Jüttner, and P. Kovács. LEMON an open source C++ graph template library. In *Workshop on Generative Technologies (WGT)*, 2010.

[12] Ernst and Young. Under cyber attack: EY's global information security survey, October 2013.

[13] S. Finbow, A.D. King, G. MacGillivray, and R. Rizzi. The firefighter problem for graphs of maximum degree three. *Discrete Mathematics*, 307(16):2094–2105, 2007.

[14] V. Gabrel and M. Minoux. A scheme for exact separation of extended cover inequalities and applicaiton to multidimensional knapsack problems. *Operations Research Letters*, 30:252–264, 2002.

[15] Z. Gu, G.L. Nemhauser, and M.W.P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10(10):427–437, 1998.

[16] B. Hartnell. Firefighter! an application of domination. Presented at the 25th Manitoba Conference on Combinatorial Mathematics and Computing, 1995.

[17] D. Hunter. An upper bound for the probability of a union. *Journal of Applied Probability*, 13:597–603, 1976.

[18] F.P. Kelly, A.K. Maulloo, and D.K.H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, pages 237–252, 1998.

[19] J.O. Kephart, G.B. Sorkin, W.C. Arnold, D.M. Chess, G. Tesauro, and S.R. White. Biologically inspired defenses against computer viruses. In *IJCAI*, pages 985–996, 1995.

[20] J.O. Kephart and S.R. White. Measuring and modeling computer virus prevalence. In *Proceedings of the 1993 IEEE Symposium on Security and Privacy*, pages 2–15, Washington, DC, USA, 1993.

[21] O. Klopfenstein and D. Nace. Cover inequalities for robust knapsack sets – application to the robust bandwidth packing problem. *Networks*, 59(1):59–72, 2012.

[22] J. Lee, V.S. Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing nonmontone submodular functions under matroid and knapsack constraints. *SIAM Journal on Discrete Math*, 23:2053–2078, 2010.

[23] J.A. Lewis. Assessing the risks of cyber terrorism cyber war and other cyber threats, 2002.

[24] R. Lougee-Heimer, F. Barahona, B.L. Dietrich, J. Fasano, J. J. Forrest, R. Harder, L. Ladanyi, T. Pfender, T. Ralphs, M. Saltzman, and K. Scheinberg. The COIN-OR initiative: accelerating operations research progress through open-source software. *ORMS Today*, 28(5), 2001.

[25] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[26] A. Prékopa, E. Boros, and K-W Lih. The use of binomial moments for bounding network reliability. In F. Roberts, F. Hwang, and C. Monma, editors, *Reliability of Computer and Communication Networks*. DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, 1991.

[27] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32:41–43, 2004.

[28] P. Veneziani. Optimality conditions for Hunter's bounds. *Discrete Mathematics*, 308:6009–6014, 2008.

[29] L.A. Wolsey. Valid inequalities for 0-1 knapsacks and MIPs with generalised upper bound constraints. *Discrete Applied Mathematics*, 29:251–261, 1990.