Risk-Averse Bi-Level Stochastic Network Interdiction Model for Cyber-Security Risk Management

May 30, 2019

Abstract

Security of cyber networks is crucial; recent severe cyber-attacks have had a devastating effect on many large organizations. The attack graph, which maps the potential attack paths of a cyber network, is a popular tool for analyzing cyber system vulnerability. In this study, we propose a bi-level stochastic network interdiction model on an attack graph to enable a risk-averse, resource constrained cyber network defender to optimally deploy security countermeasures that protect against attackers with an uncertain budget. This risk-averse conditional-value-at-risk (CVaR) model minimizes a weighted sum of the expected maximum loss over all scenarios and the expected maximum loss from the most damaging attack scenarios. We develop a customized constraint and column generation algorithm to solve our model as well as several acceleration techniques to improve the computational efficiency. Numerical experiments demonstrate that the acceleration techniques enable the solution of relatively large problems within a reasonable amount of time: applying all the acceleration techniques also reduces the average computation time of the basic algorithm by 71% for 100-node graphs. Using metrics called mean-risk value of stochastic solution and value of risk-aversion, computational results suggest that our stochastic risk-averse model significantly outperforms deterministic and risk-neutral models when 1) the distribution of attacker budget is heavy-right-tailed and 2) the defender is highly risk-averse.

Keywords: Attack graph, Stackelberg game, mixed-integer programming, conditional-value-at-risk, cyber-security

1 Introduction

Cyber network security has become a crucial issue for many organizations. In recent history, major cyber-attacks resulted in catastrophic business losses for many large-scale organizations. For example the 2017 Equifax data breach compromised the social security numbers and driver's license numbers of 143 million consumers (Bernard et al., 2017), the 2016 Uber data breach disclosed personal information of 20 million Uber users (Shields and Newcomer, 2018), and the WannaCry cyber-attack in 2017 infected more than 200,000 computers in nearly 150 countries causing an estimated loss of 4 billion U.S. dollars (Berr, 2017). The severity of these cyber-attacks suggests that protection of cyber networks is crucial for organizations to minimize loss of data due to security breaches while still taking advantage of the benefit of increased connectivity of internet networks. Therefore, research studies aim to develop methodologies to provide decision support to the organizations in mitigating the risks of cyber-attacks.

Vulnerability assessment of the cyber networks is a key input in developing robust cyber-security strategies to deploy security countermeasures (e.g., firewalls, intrusion detection/prevention systems) to cyber-attacks. Attack graph is one of the tools to analyze the vulnerability of cyber networks and develop strategies to deploy security countermeasures (Nandi et al., 2016). Attack graphs can be used to map an organization's network topology to potential attack paths.

Research studies have used attack graphs to analyze vulnerability of cyber systems; however, most of these have employed logic-based models that are ad-hoc in nature (Dewri et al., 2007). Unlike logic-based models, a rigorous mathematical model that captures the interaction between a defender and an attacker can provide more robust network interdiction (network hardening by placing security countermeasures on potential attack paths) decisions. However, very few studies have proposed rigorous mathematical models to provide decision support for cyber network interdiction. Moreover, in reality, systems are vulnerable to multiple attackers, each with different skills, resources, etc. A network interdiction decision derived by considering uncertainty in attacker capabilities is more likely to be robust compared to an interdiction decision that results from modeling a single attacker with a known capability. Furthermore, in cyber-security, it is crucial to account for minimization of the risk of most damaging attacks, since, a sudden severe attack can drive an organization out of business. Risk-neutral models may perform well for problems dealing with repetitive decision making subject to similar conditions, but perform poorly in the presence of high variability and non-repetitive decision making under uncertainty (Noyan, 2012). Since cyber-attacker capabilities vary widely, and since large cyber-attacks can render severe consequences to an organization, a risk-averse network interdiction decision would be more robust than a risk-neutral one. Therefore, a new interdiction model is needed that models a risk-averse cyber network defender under uncertainty in attacker capabilities.

In this regard, we model a risk-averse defender-attacker stochastic Stackelberg game based on an attack graph. A stochastic network interdiction model with an uncertain attacker budget can to some extent capture multiple attackers with different capabilities, thus better representing the real-life scenario than the deterministic models considering a single attacker with a fixed budget. However, traditional risk-neutral stochastic programming typically seeking to minimize expected loss does not consider the most damaging attack (loss) scenarios. In contrast, a risk-averse stochastic programming model seeks to minimize expected loss as well as the risk of most damaging cyber-attacks under uncertain attacker budget. The goal of this research is to (1) model a risk-averse cyber network defender seeking to minimize the mean-risk expected maximum loss from cyber-attacks, (2) measure the robustness of the optimal interdiction policy resulting from a risk-averse model as opposed to the optimal interdiction policy from a risk-neutral one, (3) investigate the benefits of modeling multiple attackers when computing the optimal interdiction policy, and (4) provide experimental results and managerial insights to help network owners in optimally deploying security countermeasures to minimize the risk of cyber-attacks.

1.1 Related Literature

Research studies have widely used different variations of attack graphs as network analysis tools (Phillips and Swiler, 1998; Bistarelli et al., 2006; Roy et al., 2010; Serra et al., 2015). Network

interdiction based on attack graphs involves the removal (interdiction) of a set of arcs or nodes from the attack graph, known as cut-sets, to isolate a set of goal nodes (critical assets) and thereby protect them from potential attacks. Research studies enhanced network security by generating cut-sets (Dewri et al., 2007; Noel and Jajodia, 2008; Alhomidi and Reed, 2013) in the attack graph network interdiction literature.

Past research has also modeled the interaction between an attacker and a defender as a two player Stackelberg game. Dewri et al. (2012) modeled the interaction between a network defender and an attacker by a multi-objective optimization model. Their model provides an optimal plan for placing security countermeasures on a network to maximize the return on investment for the network defender. Several uncertainties exist in these attack-graph-based network hardening problems, such as the defender's imperfect information about the attacker's exploits and the attacker's imperfect information about network topology etc. Some studies considered these uncertainties in network interdiction models based on attack graphs. Zonouz et al. (2014) proposed a response and recovery engine (RRE) to model the attacker-defender interaction as a two-player stochastic Stackelberg game. The RRE utilized attack response trees (ARTs) to analyze undesired system-level security events and to consider the uncertainties in intrusion detection alert notifications. Durkota et al. (2015b) introduced a game theoretic model to capture the interaction between defender and attacker over a dependency attack graph: the network defender attempts to reduce the risk of attacks by optimally placing "honeypots" (fake hosts) with a limited budget. Their research was further extended by Durkota et al. (2015a): the authors assumed that the attacker has incomplete information about the location of the defender-installed honeypots. Nguyen et al. (2018) proposed another game-theoretic model based on a Bayesian attack graph that models multistage interaction between a network defender and an attacker. The authors proposed heuristic strategies to solve both attacker's and defender's problems, and employed a simulation approach to analyze game models over heuristic strategies. Zhang et al. (2018) presented a Monte Carlo graph search algorithm that can capture the interaction between cyber network defender and attackers over a wide range of graph structures.

In summary, there is a lack of rigorous mathematical models on cyber network hardening using attack graphs because most of the models are logic-based and ad-hoc models. Also, according to Nandi et al. (2016), the existing algorithms to solve the two-player games over attack graphs are mostly based on either simulation or heuristics. A bi-level network interdiction model over an attack graph was proposed by Nandi et al. (2016), where the outer level represents the defender's objective of minimizing the maximum loss from attacks and the inner level models the attacker's objective of maximizing the breach loss to the network defender. The authors formulated the model as a mixed-integer programming problem and solve the resulting model by developing a constraint and column generation algorithm. Although Nandi et al. (2016) provided the most rigorous mathematical models and algorithms on network security over an attack graph to date, the authors assumed that an attacker can breach a goal node through an arc with certainty if no countermeasure is deployed on that particular arc. However, in reality, the success of an attack through an arc is uncertain even if no countermeasure is deployed on that arc. Poolsappasit et al. (2012) considered probability of success of attack through arcs.

However, the probabilities are exogenous to the model and are pre-calculated. Addressing the need for a stochastic model, a two-stage stochastic programming model based on attack graphs was formulated by Bhuiyan et al. (2016), where the authors assumed the probability of success of attack through an arc is uncertain. The results presented in Bhuiyan et al. (2016) showed that the mean value problem performed well when the mean probabilities of success of attacks through arcs are significantly different. Also, Bhuiyan et al. (2016) considered a single attacker in computing the optimal interdiction policy, which may not be robust against different attackers having different capabilities to attack.

Besides mathematical modeling based on attack graphs, some studies proposed generic mathematical models to mitigate the risk of cyber-attacks, such as allocating limited mitigation resources to reduce the vulnerability of information technology supply chain infrastructure from cyber-attacks (Zheng et al., 2019; Zheng and Albert, 2019), incentivizing the implementation of countermeasures to mitigate the risk of cyber-attacks (Zhang et al., 2017), and disconnecting the phasor measurement units from the resulting network to mitigate the risk of cyber-attack propagation in a power grid network (Mousavian et al., 2015).

Our work contributes to the general network interdiction literature as well. In this literature, research studies modeled two-player Stackelberg game in several application areas, such as interdicting a terrorist's nuclear weapons project where the interdictor's goal is to maximize the minimum completion time of the project (Brown et al., 2009; Reed, 1994), allocating security resources to maximize the resilience of a water distribution network against terrorist attacks (Qiao et al., 2007; Jiang and Liu, 2018), interdicting or mitigating the disruptions to large-scale electrical power grids that can be caused by a terrorist's attack (Salmeron et al., 2004, 2009), and hedging against worst-case facility losses to maximize coverage (O´ Hanley and Church, 2011).

Some studies also incorporated uncertainties into their bi-level network interdiction models including protection of facilities against uncertain attacks to minimize the worst-case damage (Liberatore et al., 2011), interdiction of arcs in a network to maximize the length of the shortest path (Israeli and Wood, 2002) and penetration time (Xiao et al., 2018) of the attacker in the network, and interdiction of nuclear smuggling networks (Allain, 2016; Morton et al., 2007; Nehme, 2009; Pan and Morton, 2008; Sullivan et al., 2014). There is some similarity in the modeling perspective between our work and the studies on interdiction of nuclear smuggling networks, especially with Pan and Morton (2008). Similar to our work, the nuclear smuggling interdiction models seek to install sensors on the arcs of a network to minimize the maximum probability of an evader to traverse undetected. However, the above mentioned nuclear smuggling interdiction models assumed that in a specific realization the attacker can select only a single origin-destination path. Also, those models are risk-neutral.

Few research studies on bi-level stochastic network interdiction literature considered a risk-averse objective. Song and Shen (2016) developed a risk-averse chance constrained model for a stochastic shortest path interdiction problem. The goal of the risk-averse interdictor (leader) is to maintain a high probability that the follower has to travel a longer distance than a given threshold. The authors sought to optimize Value-at-Risk (VaR) and proposed a branch-and-cut algorithm to solve the risk-averse chance constrained model. The authors reformulated their

bi-level model into a single-level model using duality. Collado et al. (2017) proposed a risk-averse solution approach to a stochastic path detection problem, where the protector allocates security resources on a network to detect an invader's path with high probability. The authors employed a mean-upper semideviation risk measure for the risk-aversion approach. The risk-averse problem was reformulated to a single-level linear mean-semideviation model. Lei et al. (2018) modeled a maximum flow interdiction problem as a stochastic bi-level and tri-level model for different risk preference combinations of a leader and a follower. Due to the continuous nature of the follower's problem (inner level), the original bi-/tri-level models were reformulated into a single-level mixed-integer linear program using duality of the inner level model. Pay et al. (2019) developed a stochastic shortest path network interdiction model where the defender (leader) seeks to maximize travel cost of the attackers' (follower) shortest path. Unlike Song and Shen (2016), the authors assumed that the risk-preference is ambiguous to the defender and is modeled using an utility function, where the defender has incomplete information about the utility function. The authors reformulated the bi-level model into a single-level and implemented a branch-and-cut algorithm to solve it.

1.2 Research Gap

We see from the literature review on network interdiction over an attack graph that there is a scarcity of rigorous mathematical analysis on attack graph network interdiction. Moreover, most of the studies in the existing literature modeled the defender-attacker interaction using one defender and one attacker on an attack graph (e.g., Nandi et al., 2016; Bhuiyan et al., 2016). However, a network is usually attacked by many attackers having different capabilities and resources. Therefore, to have a robust interdiction decision, the network defense model should consider multiple attackers with varying capabilities and resources, which can be modeled by considering attackers with different budgets.

None of the existing research considered risk-aversion in attack-graph-based network interdiction models. That said, a risk-neutral optimal solution may not be robust in a non-repetitive decision making that occurs in cyber-security, in which large attacks can cripple an organization, as cyber-attacks cause not only a financial loss, but a loss of reputation. It is impossible to recover from such consequences in a short time; if recovery is possible at all. Therefore, risk minimization for the most damaging attacks is crucial in addition to minimizing the expected loss. In this case, an interdiction policy from a risk-averse approach that considers the variability of the random parameters is more robust than the interdiction policy resulting from a risk-neutral approach.

Furthermore, we see from the generic network interdiction literature that few studies modeled a risk-averse network defender in a bi-level stochastic network interdiction problem. Though few studies dealt with the risk-averse bi-level problems, the continuous nature of the inner level problem allows the conversion into a single-level using duality. However, the presence of a discrete inner level problem makes resolving the problem computationally more difficult. This requires dealing with additional computational challenges to solve a risk-averse bi-level stochastic network interdiction problem.

The research gap discussed above suggests further research on network interdiction over

attack graphs to answer the following research questions: (1) how much is the benefit of modeling multiple attackers with different budget, (2) how much more robust is the optimal interdiction policy of a risk-averse network defender when compared to a risk-neutral one, and (3) how sensitive is the optimal interdiction policy to changes in the risk preferences of the cyber network defender. There is also a need for further research on the general risk-averse stochastic network interdiction problem to develop new models and solution approaches.

1.3 Contributions

To address the aforementioned research gaps, this paper presents a modeling framework for attack graph interdiction under uncertainty. Our model allows for multiple attackers and models risk-aversion in the defender's objective. This research is also the first to model a risk-averse cyber network defender on network hardening over an attack graph. This research extends the general stochastic network interdiction literature by introducing a new mathematical model and algorithm.

Specifically, our research makes the following contributions in this paper: (1) formulates a new bi-level risk-averse mixed-integer stochastic programming model over an attack graph incorporating the conditional-value-at-risk measure; (2) propose a customized constraint and column generation algorithm to solve the risk-averse stochastic programming model; (3) presents a novel acceleration technique that speeds up the basic algorithm significantly to solve relatively large-sized problems; and (4) provides experimental results to demonstrate the benefit of modeling multiple attackers and, yields insights into the effect of risk-aversion on the optimal interdiction policy, and the significance of solving a risk-averse stochastic network interdiction problem rather than a risk-neutral and a deterministic one.

2 Problem Description

This paper studies the interaction of a risk-averse cyber network defender with attackers in a stochastic Stackelberg game over an attack graph. In this game, the defender acts first and places security countermeasures on potential attack paths through which the attackers can breach the critical assets of the organization. The defender pays a security cost to place a security countermeasure on an arc, and the total cost of countermeasure deployment cannot exceed the budget of the defender. Realizing the defender's action, each attacker generates an attack plan to maximize the loss to the defender. The attacker must also incur an attack cost for using an arc, and the total attack cost should be within the individual attacker's budget. The budgets of the attackers are known to the defender only in distribution. The defender's objective is to minimize the expected maximum loss from all the attackers as well as the expected maximum loss from the most damaging attacks.

2.1 Attack Graphs

Attack graphs represent the potential attack paths in a network. In an attack graph, a node can represent a security condition, the state of an attack, or a vulnerability. An arc (edge) between nodes stands for an attacker's action or exploit. The tail node and the head node

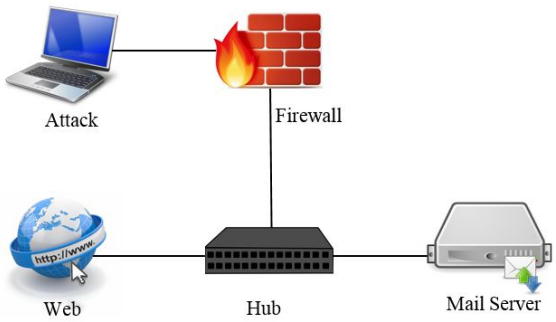


Figure 1: A simple network (adopted from Noel and Jajodia (2008))

of an arc are the pre-condition and the post-condition, respectively. To see how a network can be represented by an attack graph, consider a simple example (shown in Figure 1) first described in Noel and Jajodia (2008) in which a firewall is installed to block outside attack. However, there could be multiple paths that enable outside attackers to compromise the mail server. These potential attack paths are demonstrated in the attack graph of this network in Figure 2. The red ovals and the green rectangles represent the initial network conditions and attacker exploits, respectively. For example, the Nessus vulnerability 10671 of the web server is represented by the initial condition `nessus.10671.` The `iis_decode_bug(attack, web)` is an attacker exploit that requires two preconditions, `nessus.10671` and `execute (attack)`, to be satisfied. Completion of this exploit results in the post-condition `execute (web)`. The attacker can continue to make other exploits successful by satisfying the conditions and eventually meeting the overall attack goals, which are shown as blue hexagons.

2.2 Optimization Problem

The defender-attacker stochastic Stackelberg game over an attack graph can be formulated as a risk-averse bi-level stochastic optimization problem. The outer level represents the defender's problem, and the inner level is the attackers' problem. Both defender and attackers have limited budgets. We model multiple attackers by considering a single attacker with an uncertain budget. The defender of the network estimates the budget of the attackers from a probability distribution with known parameter values. In this stochastic optimization problem, we consider a finite number of scenarios, each representing a particular attacker. With a limited budget, the defender interdicts the best subset of arcs to minimize the expected maximum loss over all scenarios and to minimize the expected maximum loss from the most damaging attack scenarios. The expected maximum loss is the probability-weighted maximum losses from all the attack scenarios. The decision of interdicting a set of arcs is referred to as an interdiction plan. We refer to the summation of the expected maximum loss over all scenarios and the expected maximum loss from the most damaging attack scenarios as the mean-risk expected maximum loss. It is assumed that if an arc is interdicted, no attack is possible through that arc. Given a set of interdicted arcs, an attacker chooses an optimal attack plan within the limited budget to maximize the loss to the defender. An attacker starts the attack from an initially vulnerable node (initial security condition) and continues penetrating the network through the transition nodes until breaching

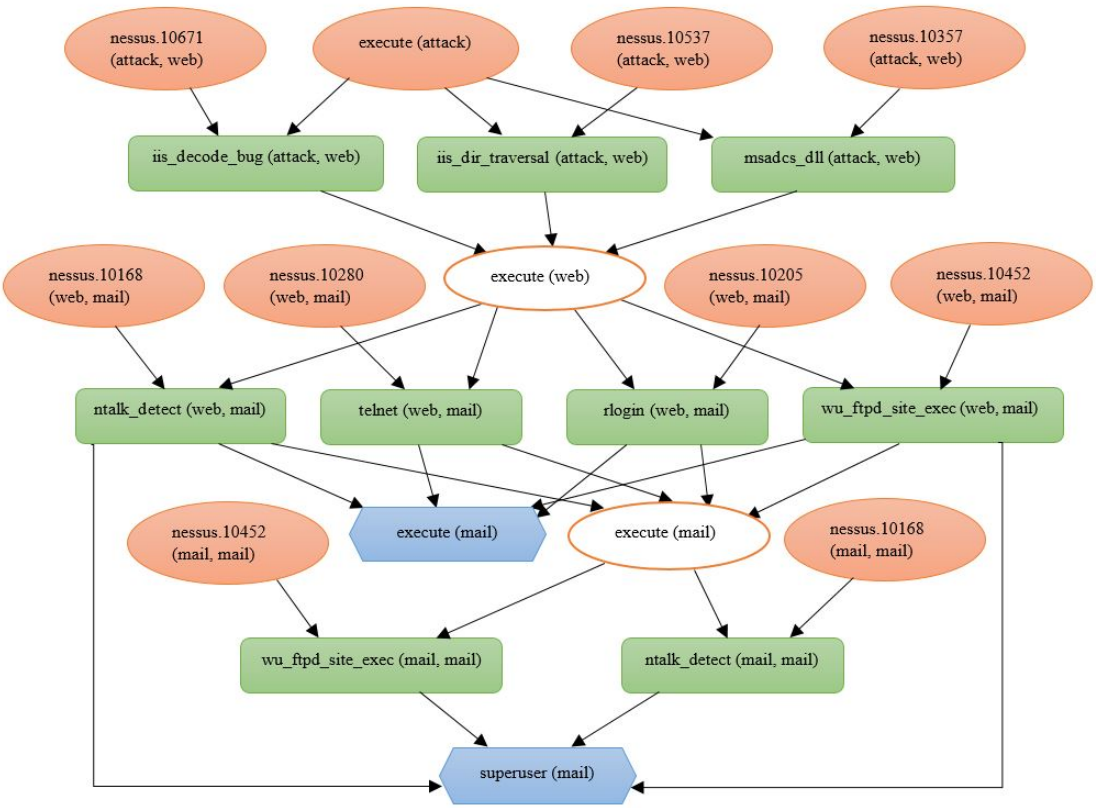


Figure 2: Attack graph of the small network (adopted from Noel and Jajodia (2008))

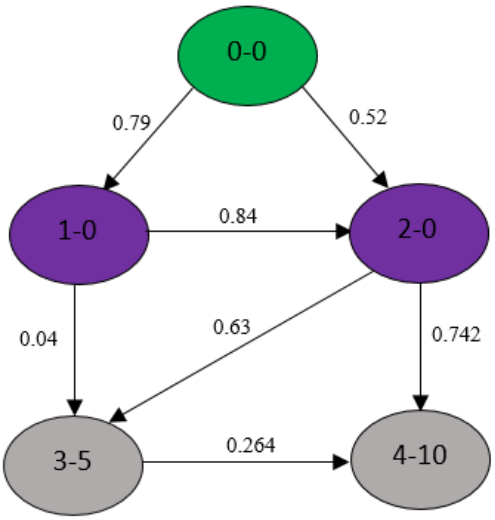


Figure 3: Attack graph example of level = 3

a goal node (critical asset). An attack path consists of the arcs from an initially vulnerable node to a goal node. A set of one or more attack paths constitutes an attack plan. It is assumed that an attacker has to incur attack cost for an arc only once even if the arc is used to breach multiple goal nodes in an attack plan, making the attacker's problem a discrete optimization problem. Also, we assume that a goal node is damaged completely if it is breached once.

2.3 Example

Figure 3 shows an example of the attack graphs used in this research. Each node is labeled as "l-b", where "l" is the node index and "b" is the loss incurred if that node is breached. The green nodes are the initially vulnerable nodes (initial security condition), the purple nodes are the transition nodes, and the grey nodes are the goal nodes (critical assets). The network defender incurs a loss if an attacker can breach one of the goal nodes. The arc weights represent the probability of success of an attack through that arc.

To understand the interaction between the defender and the attacker on the stochastic programming framework, consider a small example optimization problem. We assume two attackers with attack budgets of 2 and 3 units, respectively; using an arc at least once costs an attacker 1.0 units. We consider the two attack scenarios to occur with equal probability.

If we assume the defender has a budget of 0 units, no interdiction is possible. Now, with no interdiction, the optimal attack plan of attacker 1 is to breach goal node 4 through the attack path 0-2-4, which results in a maximum loss of 3.86 (=0.52×0.742×10) to the defender at a cost of 2.0 to the attacker. The optimal attack plan of attacker 2 is to use the attack paths 0-2-3 and 0-2-4 that result in a maximum loss of 5.496 (=0.52×0.742×10 + 0.52×0.63×5) at a cost of 3.0 to the attacker. Thus, the expected maximum loss to the defender is 4.68 (=0.50×3.86 + 0.50×5.496).

Now, if we assume that the defender has a budget of 1 unit, and the defender interdicts arc (0, 2), the optimal attack plan of attacker 1 is to use path 0-1-3 resulting in a maximum loss of 0.16. Attacker 2 chooses the attack path 0-1-2-4 that leads to a maximum loss of 4.92. The resulting expected maximum loss from the two scenarios becomes 2.54.

This example optimization problem, which seeks to minimize the expected maximum loss, raises an important question: if the defender was also concerned with minimizing the risk of the most damaging scenarios, would the defender's solution change? To address this question, we incorporate CVaR into our stochastic programming framework. This risk-averse approach is described in detail in the next section.

3 Mathematical Formulation

In this section, we formulate the risk-averse bi-level problem as a two-stage stochastic mixed-integer programming model. The first-stage model represents the outer level, which is referred as MinMaxExpLoss. The second-stage model stands for the inner level and is referred as MaxLoss. The parameters and variables for the mathematical formulation are listed in Table 1.

Table 1: Notation.

(a) Sets

Sets	Description
$\mathcal{N}$	Set of nodes
$\mathcal{N}_{\mathcal{I}}$	Set of initially vulnerable nodes
$\mathcal{N}_{\mathcal{T}}$	Set of goal nodes
$\mathcal{A}$	Set of arcs
$\mathcal{A}_l(i)$	Set of arcs leaving node i
$\mathcal{A}_e(i)$	Set of arcs entering node i
$\mathcal{S}$	Set of scenarios
$\mathcal{P}_k^s$	Set of paths in attack scenario s at iteration k
$\mathcal{A}_p$	Set of arcs in path p

(b) Parameters

Parameters	Description
l_t	Loss resulting from breaching a goal node $t \in \mathcal{N}_{\mathcal{T}}$
p_{ij}	Probability of success of attack through arc (i,j)
p^s	Probability of scenario s
c_{ij}^d	Cost of deploying countermeasures on arc (i,j)
c_{ij}^a	Cost of attack through arc (i,j)
λ	Risk coefficient
α	Level of confidence
b_d	Defender's budget
b_a	Attacker's budget
l_p^b	Loss resulting from breaching a goal node through path p
P_p^s	Probability of path p in scenario s

(c) Variables

Variables	Description
x_{ij}	1 if countermeasures are deployed on arc (i,j), 0 otherwise
f_{ij}	1 if arc (i,j) is used for one or more attacks, 0 otherwise
z_i	Probability of node i being breached
y_{ij}	Product of z_i and f_{ij}
η	1st stage variable (represents the value-at-risk, V_aR)
v^s	Excess loss variable for scenario $s \in \mathcal{S}$
r_p	1 if path p is interdicted, 0 otherwise

As we consider risk-aversion in the two-stage stochastic programming model, the risk-averse framework is described in the following sub-section.

3.1 CVaR Measure in Stochastic Programming

In this paper, the attacker budget is a random parameter which is estimated by the defender from a probability distribution. Therefore, the maximum loss to the defender in different scenarios is also a random variable, which in turn makes the expected maximum loss a random variable as well. In a risk-neutral approach, we would compare only the expected values in deciding the optimal solution to the stochastic programming problems. However, it is crucial to

consider the effect of variability and take risk measures as preference criteria in comparison of the random variables (Noyan, 2012). Mean-risk models are developed in stochastic programming to incorporate the risk measures that provides a robust solution in the presence of variability. The mean-risk function of a stochastic programming problem involves a risk-measure component in addition to the traditional expected value component:

$$min\ \mathbb{E}(g(\mathbf{x},\, \xi)) + \lambda\phi\,(g(\mathbf{x},\, \xi)) \tag{1}$$

where $g(\mathbf{x},\, \xi)$ is the second-stage problem for a particular realization of the random parameter ξ, $\mathbf{x}$ is the first-stage decision vector, ϕ is a specific risk measure, and λ is the risk coefficient. The value of the risk coefficient depends on the degree of risk preference of the decision maker. The risk coefficient represents the exchange rate of mean cost for risk. In the existing literature, several different risk measures are presented, for example, Ahmed (2006) presented several computationally tractable mean-risk models. In our risk-averse approach, we incorporate a conditional-value-at-risk measure, which is a well known downside risk measure. Artzner et al. (1999) presented some axiomatic properties required for risk measures to be coherent, and the authors showed that the CVaR is a coherent risk measure.

In this paper, we incorporate the CVaR approach similar to Noyan (2012) and Schultz and Tiedemann (2006). Both in our paper and in Schultz and Tiedemann (2006), the recourse function contains integer variables. Due to the mixed-integer recourse, convexity of the objective function in a mean-risk model does not hold. The risk measures should have some properties that make the corresponding mean-risk mixed-integer stochastic programs structurally sound and computationally tractable to make them applicable to real-life problems (Schultz and Tiedemann, 2006). Schultz and Tiedemann (2006) demonstrated that the CVaR measure possesses these requirements.

The conditional-value-at-risk quantifies the expected value of the $\alpha-$tail distribution of $g(\mathbf{x},\, \xi)$. Now, the mean-risk function (1) is as follows:

$$min\ \mathbb{E}(g(\mathbf{x},\, \xi) + \lambda CVaR_{\alpha}\,(g(\mathbf{x},\, \xi)) \tag{2}$$

where $CVaR_{\alpha}$ stands for the conditional-value-at-risk at the level of confidence α. In the context of our work, the $CVaR_{\alpha}$ computes the expected value of the excess losses that exceeds the value-at-risk at confidence level α. Value-at-risk (VaR_{α}) is also a risk measure that provides an upper bound on the loss that is exceeded only with a probability of $1-\alpha$. The value-at-risk can be mathematically expressed as:

$$VaR_{\alpha}(g) = inf\{\eta\,:\,\varphi(x,\eta) \geq \alpha\} \tag{3}$$

where $\varphi(x,\eta)$ is the distribution function of $g(\mathbf{x},\, \xi)$, and level of confidence $\alpha \in (0,1)$

The relation between the $CVaR_{\alpha}$ and VaR_{α} corresponding to the random variable g can be expressed as:

$$CVaR_\alpha(g) = \mathbb{E}(g \mid g \geq VaR_\alpha(g)) \tag{4}$$

The $CVaR_\alpha$ is also referred to as tail VaR at confidence level α. We can compute the conditional-value-at-risk at confidence level α from the following expression:

$$CVaR_\alpha(g) = inf_{\eta \in \mathbb{R}}\{\eta + \frac{1}{1-\alpha}\mathbb{E}(max(g-\eta), 0\} \tag{5}$$

According to Schultz and Tiedemann (2006), for a finite number of scenarios $\xi^1, \xi^2, ..., \xi^{|\mathcal{S}|}$ and the corresponding probabilities $p^1, p^2, ..., p^{|\mathcal{S}|}$, minimization of the conditional-value-at-risk

$$min\ CVaR_\alpha\, g(\mathbf{x}, \xi) \tag{6}$$

can be reformulated as

$$min\ \{\eta + \frac{1}{1-\alpha}\sum_{s\in\mathcal{S}} p^s v^s : Wy^s = \mathbf{h}^s - T^s\mathbf{x}\} \tag{7}$$

$$\text{where, } v^s \geq (\mathbf{q}^s)^T\mathbf{y}^s - \eta, \qquad \forall s \in \mathcal{S}$$

$$y^s \geq 0, \qquad \forall s \in \mathcal{S}$$

$$\mathbf{x} \in X$$

$$\eta \in \mathbb{R}$$

$$v^s \geq 0, \qquad \forall s \in \mathcal{S}$$

where v^s represents the excess loss in scenario $s \in \mathcal{S}$ and is considered an additional second-stage variable. The variable η acts as an additional first-stage variable. The CVaR measure is incorporated to the bi-level two-stage stochastic programming model and the resulting mean-risk model is presented in the following subsection.

3.2 Mean-Risk Two-Stage Stochastic Programming Model

The first-stage model represents the defender's objective, which is to minimize the expected maximum loss from all the scenarios as well as to minimize the expected maximum loss from the most damaging attack scenarios. The first-stage problem is formulated as follows (MinMaxExpLoss) :

$$min \quad \sum_{s\in\mathcal{S}} p^s Q^s(\hat{x}) + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{s\in\mathcal{S}} p^s v^s\right) \tag{8a}$$

$$\text{s.t.} \quad c^d_{ij}x_{ij} \leq b_d \tag{8b}$$

$$v^s \geq Q^s(\hat{x}) - \eta \qquad \forall s \in \mathcal{S} \tag{8c}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A} \tag{8d}$$

$$\eta \in \mathbb{R} \tag{8e}$$

$$v^s \geq 0 \qquad \forall s \in \mathcal{S} \tag{8f}$$

The objective function (8a) has two components, where the first component computes the expected maximum loss over all the scenarios and the second component models the CVaR measure. Constraint (8b) ensures that the total cost of deploying countermeasures on a subset of arcs should be within the defender's budget. Constraints (8c) compute the excess loss for all the attack scenarios.

The second-stage model (sub-problem) stands for a particular realization of the random attacker budget. The sub-problem from each scenario represents an individual attacker (inner level) whose objective is to maximize the total loss to the defender for a given interdiction plan of the defender $\hat{\mathbf{x}}$ determined by the first-stage (outer level) model. We refer to the following formulation of the inner problem as MaxLossNLP.

$$
\begin{aligned}
Q^s(\hat{x}) = \quad & max \sum_{t \in \mathcal{N}_\mathcal{T}} l_t z_t && \text{(9a)}\\
\text{s.t.} \quad & \sum_{(i,j) \in \mathcal{A}} c^a_{ij} f_{ij} \leq b_a && \text{(9b)}\\
& f_{ij} \leq 1 - \hat{x}_{ij} \qquad \forall (i,j) \in \mathcal{A} && \text{(9c)}\\
& z_j \leq \sum_{(i,j) \in \mathcal{A}_e(j)} p_{ij} z_i f_{ij} \qquad \forall j \in \mathcal{N} \diagdown \mathcal{N}_\mathcal{I} && \text{(9d)}\\
& \sum_{(i,j) \in \mathcal{A}_e(j)} f_{ij} \leq 1 \qquad \forall j \in \mathcal{N} \diagdown \mathcal{N}_\mathcal{I} && \text{(9e)}\\
& f_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A} && \text{(9f)}\\
& 0 \leq z_j \leq 1 \qquad \forall j \in \mathcal{N} && \text{(9g)}
\end{aligned}
$$

The objective function (9a) calculates the maximum total loss to the defender or the maximum total reward acquired by the attacker given an interdiction plan of the defender. Constraint (9b) enforces that the total attack cost cannot exceed the attacker's budget. Each of the constraints (9c) ensures that no attack is possible through an arc if that arc is already interdicted by the defender in the first-stage model. Our model ensures that the attacker chooses an arc with maximum attack success probability to breach a node, which is enforced by constraints (9d). Constraints (9e) ensure our assumption that the attack cost is incurred once for an arc even if the arc is used in multiple attack paths.

In the MaxLossNLP(9) formulation, the constraint (9d) contains product of the two variables f_{ij} and z_i that makes MaxLossNLP a nonlinear formulation. However, this type of non-linearity can be linearized by employing standard procedures. The linearization technique introduces an additional continuous variable y_{ij}, which replaces the product term $z_i f_{ij}$. The linearization also introduces some additional linearization constraints. We refer to the linearized second-stage model as MaxLossMIP, which is presented as follows:

$$
\begin{aligned}
Q^s(\hat{x}) = \quad & max \sum_{t \in \mathcal{N}_\mathcal{T}} l_t z_t && \text{(10a)}\\
\text{s.t} \quad & \sum_{(i,j)\in\mathcal{A}} c^a_{ij} f_{ij} \leq b_a && \text{(10b)}\\
& f_{ij} \leq 1 - \hat{x}_{ij} \qquad \forall (i,j) \in \mathcal{A} && \text{(10c)}\\
& z_j \leq \sum_{(i,j)\in\mathcal{A}_e(j)} p_{ij} y_{ij} \qquad \forall j \in \mathcal{N} \diagdown \mathcal{N}_\mathcal{I} && \text{(10d)}\\
& y_{ij} \geq z_i - (1 - f_{ij}) \qquad \forall (i,j) \in \mathcal{A} && \text{(10e)}\\
& y_{ij} \leq f_{ij} \qquad \forall (i,j) \in \mathcal{A} && \text{(10f)}\\
& y_{ij} \leq z_i \qquad \forall (i,j) \in \mathcal{A} && \text{(10g)}\\
& \sum_{(i,j)\in\mathcal{A}_e(j)} f_{ij} \leq 1 \qquad \forall j \in \mathcal{N} \diagdown \mathcal{N}_\mathcal{I} && \text{(10h)}\\
& f_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A} && \text{(10i)}\\
& 0 \leq z_j \leq 1 \qquad \forall j \in \mathcal{N} && \text{(10j)}\\
& 0 \leq y_{ij} \leq 1 \qquad \forall (i,j) \in \mathcal{A} && \text{(10k)}
\end{aligned}
$$

The MaxLossMIP is a mixed-integer linear programming formulation of the second-stage problem. Constraints (10d)-(10g) are introduced to the model due to the linearization of the nonlinear constraint (9d) of MaxLossNLP (9). The objective function and the other constraints of MaxLossMIP are the same as those of MaxLossNLP.

4 Solution Approach

This section details the solution methodology for solving our risk-averse bi-level stochastic programming model. Most of the research in the existing literature contains binary variables only in the outer level, a situation that is computationally advantageous because a bi-level model can be reformulated as a single-level model by taking the dual of the inner level model (Wood, 1993; Israeli and Wood, 2002). However, some studies have dealt with the solution of bi-level problems that involve binary variables in both levels (Scaparra and Church, 2008; Brown et al., 2009; Nandi et al., 2016). Moore and Bard (1990) discussed the difficulties faced in the solution of bi-level mixed-integer programming.

In our paper, the binary variables exist in both inner and outer levels. Therefore, we cannot utilize duality of the inner problem to make the whole problem as a nested min-min problem. Additionally, we are solving a risk-averse stochastic mixed-integer problem, which also makes our problem computationally challenging. Therefore, to solve the formulated model, we develop a customized constraint and column generation (CCG) algorithm based on the algorithm proposed by Nandi et al. (2016). Algorithms using a similar framework are also proposed by Brown et al. (2009) and Alderson et al. (2014). However, each of these previous studies are on deterministic bi-level problems. We extend their framework to incorporate the solution strategies for stochastic mixed-integer programming and conditional-value-at-risk framework. The solution of the outer

level problem (i.e., the master problem) provides the lower bound of the algorithm. The upper bound of the algorithm is obtained by solving the scenario sub-problems (MaxLossMIP (10)) for a given solution from the outer level problem.

4.1 Upper Bound

A feasible solution of the outer level problem (MinMaxExpLoss (8)) provides a feasible interdiction plan of the network defender. For a given feasible interdiction plan ($\hat{x}^k$) at an iteration k, the attackers try to maximize their gain. We solve each scenario sub-problem (MaxLossMIP (10)) at iteration k for a given interdiction plan. To compute upper bound at an iteration k, we calculate the expected value of the optimal objective values of the scenario sub-problems as well as the CVaR at confidence level α. Therefore, $Q(\hat{x}^k) = \sum_{s\in\mathcal{S}} p^s Q^s(\hat{x}) + \lambda(\eta + \frac{1}{1-\alpha}\sum_{s\in\mathcal{S}} p^s v^s)$ is the upper bound of our algorithm at iteration k. As our problem is a minimization problem, the upper bound (ub) up to iteration k is the minimum of $Q(\hat{x}^k)$ found through iteration k, i.e, $ub \leq Q(\hat{x}^{k'})$ for $k' = 1, \ldots, k$.

4.2 Lower Bound

To compute the lower bound of our algorithm, we use a technique similar to the MinAtRisk model of Nandi and Medal (2016) that minimizes the number of nodes at risk of infection from other already-infected nodes. In this research, we formulate an algorithm for solving our specific defender's problem. We refer to our developed algorithm for calculating the lower bound as MinMeanRisk($\bar{\mathcal{A}}^k$), where $\bar{\mathcal{A}}^k$ is the set of arcs used by the attackers up to iteration k. We obtain this set by solving the MaxLossMIP(10) problem at iteration k of the algorithm. Let $f(\bar{\mathcal{A}}^k)$ be the optimal objective value and $\hat{x}^k$ be the optimal solution of the MinMeanRisk($\bar{\mathcal{A}}^k$) model, then the current lower bound (lb) at iteration k is the maximum of $f(\bar{\mathcal{A}}^k)$ found through iteration k, i.e., $lb \geq f(\bar{\mathcal{A}}^{k'})$ for $k' = 1, \ldots, k$.

The model providing the lower bound is formulated as follows (MinMeanRisk($\bar{\mathcal{A}}^k$)):

$$f(\bar{\mathcal{A}}^k) = \quad min\, \theta \tag{11a}$$

$$\text{s.t.} \quad \theta \geq \sum_{s\in\mathcal{S}} p^s \left(\sum_{t\in\mathcal{N}_{\mathcal{T}}^{ks}} l_t z_t^{ks} \right) + \lambda \left(\eta^k + \frac{1}{1-\alpha} \sum_{s\in\mathcal{S}} p^s v^{ks} \right) \; \forall k \in \mathcal{K} \tag{11b}$$

$$z_j^{ks} \geq p_{ij}^s z_i^{ks} - x_{ij} \qquad \forall (i,j) \in \mathcal{A}^{ks},\, k \in \mathcal{K},\, s \in \mathcal{S} \tag{11c}$$

$$z_i^{ks} = 1 \qquad \forall i \in \mathcal{N}_{\mathcal{I}}^{ks},\, k \in \mathcal{K},\, s \in \mathcal{S} \tag{11d}$$

$$\sum_{(i,j)\in\mathcal{A}} c_{ij}^d x_{ij} \leq b_d \tag{11e}$$

$$v^{ks} \geq \sum_{t\in\mathcal{N}_t^{ks}} l_t z_t^{ks} - \eta^k \qquad \forall k \in \mathcal{K},\, s \in \mathcal{S} \tag{11f}$$

$$x_{ij} \in \{0,\, 1\} \qquad \forall (i,j) \in \mathcal{A} \tag{11g}$$

$$\eta^k \in \mathbb{R} \qquad \forall k \in \mathcal{K} \tag{11h}$$

$$v^{ks} \geq 0 \qquad \forall k \in \mathcal{K},\, s \in \mathcal{S} \tag{11i}$$

In this formulation, the objective function (11a) and the constraint (11b) together ensure that the objective of this model is to minimize the maximum of all the mean-risk expected maximum losses through iteration k. Each constraint (11c) ensures that if a node j is at risk of breach through arc $(i,\, j)$ in an attack scenario s at iteration k, this arc must be interdicted to protect the node j from being breached by an attacker. Therefore, to protect a node j, all the incoming arcs need to be interdicted if the tail node of the arc is at risk of breach. The initially vulnerable nodes used in an attack scenario s at iteration k are always at the risk of breach. This condition is satisfied by the constraints (11d). Constraint (11e) represents the defender's budget constraint. Each of the constraints (11f) computes the excess loss corresponding to the attack scenario s at iteration k.

In this formulation, we generate a new set of variables for the nodes and arcs used in the attack scenarios through iteration k. Also, we generate the associated constraints (11b), (11c), (11d), and (11f) to represent the connectivity of the nodes. To distinguish among different attack scenarios, a new variable z_j^{ks} is generated for node j if the node is used in attack scenario s at iteration k. At each iteration of the MinMeanRisk($\bar{\mathcal{A}}^k$) model, the constraints (11b) force the lower bound closer to optimal solution, which resembles an optimality cut in a L-shaped algorithm. As the algorithm proceeds, the new variables and constraints associated with the attack scenarios are added to the model, and the solution of the master problem moves towards the optimal solution.

The proof of theorem 1 explains the theoretical justification that the master problem provides a valid lower bound to the algorithm.

Theorem 1. *The master problem (*MinMeanRisk*) provides a valid lower bound.*

Proof. According to Nandi et al. (2016), an attack is a tree in the attack graph. In our MinMeanRisk (11) formulation, the set of constraints (11c), (11d), and (11f) add $\mid \mathcal{S} \mid$ attack

trees to the attack graph used by the master problem at each iteration of the algorithm. Here, $s \in \mathcal{S}$ stands for the scenario index. The fact that each distinct attack tree corresponding to a scenario in a set is due to the distinct attacker budgets in the scenarios. The master problem formulation adds k distinct sets of $\mid \mathcal{S} \mid$ attack trees through iteration k. Since a subset of all the possible alternative sets of attack trees are added to the MinMeanRisk(11) formulation through iteration k, the objective value of the master problem provides a lower bound to the minimization problem. □

4.3 CCG Algorithm

The complete pseudocode of the customized constraint and column generation algorithm, so called because new variables and constraints are added at each iteration, is demonstrated in Algorithm 1.

Algorithm 1 Constraint and Column Generation Algorithm

function ConstraintandColumnGeneration
  initialize. Set $ub \leftarrow \infty$, $lb \leftarrow 0$, $k \leftarrow 1$; $\epsilon \leftarrow$a small number, and $\mathbf{x}^* \leftarrow \mathbf{0}$
  while $ub - lb > \epsilon$ **do**
    Solve master problem MinMeanRisk(11), returning $f(\bar{\mathcal{A}}^k)$ and $\hat{\mathbf{x}}^k$.
    if $f(\bar{\mathcal{A}}^k) > lb$ **then** $lb \leftarrow f(\bar{\mathcal{A}}^k)$ and $\mathbf{x}^* \leftarrow \hat{\mathbf{x}}^k$.
    Solve subproblem MaxLossMIP($\hat{\mathbf{x}}^k$) $\forall s \in \mathcal{S}$, returning optimal attack plans $\hat{\mathbf{f}}^{ks}$ and set of nodes $\mathcal{N}^{ks}$ and arcs $\bar{\mathcal{A}}^{ks}$ used in attacks.
    Compute $Q(\hat{\mathbf{x}}^k) := \sum_{s\in\mathcal{S}} p^s Q^s(\hat{x}) + \lambda \left(\eta + \frac{1}{1-\alpha} \sum_{s\in\mathcal{S}} p^s v^s\right)$.
    if $Q(\hat{\mathbf{x}}^k) < ub$ **then** $ub \leftarrow Q(\hat{\mathbf{x}}^k)$ and $\mathbf{x}^* \leftarrow \hat{\mathbf{x}}^k$.
    if $ub - lb \leq \epsilon$ **then break.** Otherwise, go to next step.
    Create variables z_i^{ks} for the nodes used in attacks, and add constraints (11b), (11c),(11d), and (11f) to the master problem corresponding to $\mathcal{N}^{ks}$ and $\bar{\mathcal{A}}^{ks}$.
    $k \leftarrow k + 1$.
  return Optimal interdiction plan, $\mathbf{x}^*$

To provide a better explanation of the CCG algorithm, a few steps of the algorithm are demonstrated with a simple numerical example in Appendix A.

The proofs of Lemma 1 and Theorem 2 in Appendix B explain the theoretical foundation of the convergence of our algorithm.

4.4 Acceleration Techniques

In the MinMeanRisk algorithm, we are adding variables and constraints for each node and arc used in a new attack at each iteration. Due to adding the large number of variables and constraints in each iteration, the master problem computation time increases exponentially in the problem size. To improve the computational efficiency, we implement the following enhancements to the algorithm.

4.4.1 Path-Based Formulation

Nandi et al. (2016) showed that an attacker solution can be represented by a distinct set of paths. We run a search algorithm on the attacker solutions of each scenario at each iteration to find the set of paths used by the attackers and compute the probability of success of an attack through a path. Once a new path is found in an iteration, we add a path variable (r_p) and the associated constraints to the master problem. If a path is found that was already used in a previous attack scenario, we also add the path to the associated constraints. The path-based formulation (MinMeanRiskPath) of the master problem is as follows:

$$
\begin{aligned}
f(\bar{\mathcal{A}}^k) = \quad & min\, \theta && && (12a)\\
s.t. \quad & \theta \geq \sum_{s\in\mathcal{S}} p^s \left(\sum_{p\in\mathcal{P}_k^s} l_p^b P_p^s (1-r_p) \right) && && \\
& \qquad +\lambda \left(\eta^k + \frac{1}{1-\alpha} \sum_{s\in\mathcal{S}} p^s v^{ks} \right) && \forall k \in \mathcal{K} && (12b)\\
& r_p \leq \sum_{(i,j)\in\mathcal{A}_p} x_{ij} && \forall p \in \cup_{k\in\mathcal{K}, s\in\mathcal{S}} \mathcal{P}_k^s && (12c)\\
& r_p \leq 1 && \forall p \in \cup_{k\in\mathcal{K}, s\in\mathcal{S}} \mathcal{P}_k^s && (12d)\\
& \sum_{(i,j)\in\mathcal{A}} c_{ij}^d x_{ij} \leq b_d && && (12e)\\
& v^{ks} \geq \sum_{p\in\mathcal{P}_k^s} l_p^b P_p^s (1-r_p) - \eta^k && \forall k \in \mathcal{K}, s \in \mathcal{S} && (12f)\\
& x_{ij} \in \{0,1\} && \forall (i,j) \in \mathcal{A} && (12g)\\
& \eta^k \in \mathbb{R} && \forall k \in \mathcal{K} && (12h)\\
& v^{ks} \geq 0 && \forall k \in \mathcal{K}, s \in \mathcal{S} && (12i)\\
& r_p \geq 0 && \forall p \in \cup_{k\in\mathcal{K}, s\in\mathcal{S}} \mathcal{P}_k^s && (12j)
\end{aligned}
$$

Each constraint (12b) computes the mean-risk expected maximum loss at iteration k. The objective (12a) and constraints (12b) together ensure that the algorithm minimizes the maximum of all the mean-risk expected maximum losses over the iterations up to $|\mathcal{K}|$. For each path p, each pair of the constraints (12c) and (12d) ensures that the path is interdicted if only one arc in that path is interdicted. Constraint (12e) represents the budget limitation of the defender. Constraints (12f) compute the excess loss over all the scenarios in each iteration. The binary restrictions of the interdiction variables are imposed by constraints (12g). Constraints (12h), (12i), and (12j) stand for the sign restrictions of the associated variables. This MinMeanRiskPath formulation adds much fewer variables and constraints to the master problem at each iteration than the previous MinMeanRisk(11) formulation, thus reducing the master problem computation time.

4.4.2 Multiple Sub-Problem Solutions (Ms)

In our model, each scenario represents an attacker problem or the sub-problem. If we add only one solution from each scenario sub-problem to the master problem at each iteration, the algorithm runs through a large number of iterations until enough attacker solutions are added for convergence. To reduce the number of iterations and thus ensure quick convergence, we add multiple optimal and sub-optimal solutions of a scenario sub-problem to the master problem. (The Gurobi optimizer (Gurobi Optimization Inc., 2017) provides multiple optimal and sub-optimal solutions to the attacker problems). Adding these multiple sub-problem solutions to the master problem reduces the number of iterations and the average computation time. Computational experiments demonstrate that adding the best 30% of solutions provides the best results.

4.4.3 Trust Region Constraints (Tr)

At the initial iterations of the MinMeanRiskPath(12) formulation, the model produces very divergent solutions, resulting in a long time to convergence. To stabilize the master problem solution at the initial iterations, we add a trust region cut to the master problem in the early iterations of the algorithm. If $\hat{\mathbf{x}}^k$ is the master problem solution for iteration k and $\hat{\mathcal{X}}_1^k = \{(i,\, j) : \hat{x}_{ij}^k = 1\}$ then the trust region cut added to the master problem at iteration $k+1$ is as follows:

$$\sum_{(i,j)\notin\hat{\mathcal{X}}_1^k} x_{ij} + \sum_{(i,j)\in\hat{\mathcal{X}}_1^k} (1 - x_{ij}) \leq 0.40 \times 2\times \mid \hat{\mathcal{X}}_1^k \mid \tag{13}$$

This constraint (13) ensures that the maximum change in the arc interdiction between iterations k and $k+1$ is for 40% of the arcs. The left side of the constraint (13) calculates the Hamming distance (Hamming, 1950) between the interdiction plans of iteration k and iteration $k+1$. The right side of the constraint (13) forces that among the interdicted arcs at iteration k, a maximum of 40% can be replaced at iteration $k+1$. Experiments show that the trust region cut reduces the master problem solution time and the algorithm runtime.

4.4.4 Heuristic Solution to Master Problem (Hf)

To obtain a better solution of the master problem quickly, we use a heuristic to generate a solution used as a warm start to the master problem solution. Our heuristic is similar to the greedy heuristic proposed by Toyoda (1975) and Nandi et al. (2016). The heuristic method is shown in Algorithm 2.

Algorithm 2 Greedy Heuristic Generating Initial Solution of the Master Problem

function GREEDYHEURISTIC
  initialize. Set of arcs to be interdicted, $\mathbf{x} \leftarrow \emptyset$, and total cost of interdiction, $TCost \leftarrow 0$.
  while $TCost + c^d_{ij} \leq b_d$ **do**
    Compute $Score_{ij} = \frac{LossReduction_{ij}}{c^d_{ij}}$ for each arc not interdicted.
    $LossReduction_{ij}$ = Current MREXPLOSS - MREXPLOSS after interdicting arc (i, j)
    Interdict arc having maximum $Score_{ij}$, and $TCost \leftarrow TCost + c^d_{ij}$.
    if $TCost > b_d$ **then break.**
  return Set of interdicted arcs, $\mathbf{x}$

4.4.5 Separation of Sub-problem Solutions (Ss)

At each iteration of the algorithm, we add an attacker solution for each scenario sub-problem to the master problem. To add more different attacker solutions to the master problem, we add a constraint to each scenario sub-problems at the first few iterations of the algorithm that forces the sub-problems to generate different attack plans between two consecutive iterations. These sub-problem separation constraints ensure that enough attacker solutions are explored more quickly by the master problem and thus significantly reduce both number of iterations to converge and average computation time.

Assume that $\hat{\mathbf{f}}^{ks}$ is the solution of the scenario sub-problem s at iteration k and $\hat{\mathcal{F}}^{ks}_1 = \{(i, j) : \hat{f}^{ks}_{ij} = 1\}$. Then, we add the following constraint to the scenario sub-problem s at iteration $k + 1$.

$$\sum_{(i,j) \notin \hat{\mathcal{F}}^{ks}_1} f^{ks}_{ij} + \sum_{(i,j) \in \hat{\mathcal{F}}^{ks}_1} (1 - f^{ks}_{ij}) \geq 0.10 \times 2 \times \mid \hat{\mathcal{F}}^{ks}_1 \mid \tag{14}$$

The left side of the constraint (14) computes the Hamming distance between the optimal attack plans of scenario s at iterations k and $k + 1$. The right side of the constraint (14) forces that at least 10% of the arcs chosen by the optimal attacker solution of scenario s at iteration k should not be selected by the optimal attacker solution of scenario s at iteration $k + 1$. We add these constraints at the few initial iterations of the algorithm. This is a novel acceleration technique, and the experiments show that this reduces the average master problem computation time by approximately 18%.

5 Numerical Experiments

We performed computational experiments to evaluate the performance of our risk-averse bi-level stochastic programming model in minimizing the mean-risk expected maximum loss from cyber-attacks. We analyzed the effects of the model parameters and the topological attributes of the attack graphs on the computation time of the algorithm. We conducted all the experiments on a laptop with an Intel core i7 2.80GHz processor and 8GB RAM. The algorithms were implemented in Python 2.7 with the Gurobi optimizer (Gurobi Optimization Inc., 2017).

5.1 Parameter Set-up

We performed experiments on synthetic attack graphs generated using the same approach as in Nandi et al. (2016). We refer the readers to Nandi et al. (2016) for more detail on the attack graph generation procedure. We conducted numerical experiments using four different sizes of attack graphs each having five levels to demonstrate the effects of network topology on computation time. The following graph parameters were generated using a uniform distribution: breach loss of the goal nodes, cost of attack through each arc, and cost of deploying countermeasure on each arc. The mean probabilities of success of attack through arcs were also generated using a uniform distribution.

We modeled the random attack budget using normal and Weibull distributions. To construct a right-tailed asymmetric attacker budget distribution, we used Weibull distribution. We discretized the continuous normal and Weibull distribution to obtain discrete attacker budget distribution. To demonstrate the effects of the risk parameters, we performed experiments with two different levels of confidence, α and several levels of risk coefficient, λ. Table 2 shows the different parameters along with their values used in our experiments. The base configurations of the parameters are shown in boldface type in Table 2. In the numerical experiments, all parameters are set at their base values unless otherwise specified.

Table 2: Parameters and their values used in the experiments.

Parameters	**Values**
Network size (nodes, $\mid \mathcal{N} \mid$)	50, **100**, 150, 200
Arcs, $\mid \mathcal{A} \mid$	$\approx$**2.15**$\times \mid \mathcal{N} \mid$
Breach loss of the goal nodes	$\sim$**uniform(500, 1500)**, $\sim$uniform(200, 2000)
Cost of attacks through arcs, c_{ij}^a	$\sim$**uniform(10, 30)**
Cost of countermeasures deployment on arcs, c_{ij}^d	$\sim$**uniform(10, 30)**
Defender Budget, b_d	100, **150**, 200, 255
Attacker Budget, b_a	$\sim$normal(300, 60), $\sim$**Weibull(100, 200),** $\sim$Weibull(50,500)
Probability of attack success, p_{ij}	$\sim$uniform(0, 1)
Number of scenarios, $\mid \mathcal{S} \mid$	40, **100**, 150
Level of confidence, α	0.7, **0.9**
Risk coefficient, λ	**(0–100)**

5.2 Runtime and Solution Quality of Solution Procedures

This section presents the results on the computation time of the path-based CCG algorithm, effects of the acceleration techniques on the computational speed and solution quality of the exact algorithm, effects of the network size and the number of scenarios on the algorithm runtime.

We performed all computational experiments with the path-based formulation of our CCG algorithm (MinMeanRiskPath(12)), as it was shown in Nandi et al. (2016) that the path-based formulation outperforms the node-based formulation on computational speed. In this subsection, we set the parameter values as follows: $b_d = 100$, $ba \sim w(80, 180)$ for 50-node networks;

$b_d = 150$, $ba \sim w(100, 200)$ for 100-node and 150-node networks; $b_d = 200$, $ba \sim w(150, 250)$ for 200-node networks; and $\lambda = 0.1$ for all problem instances.

The total runtime, master problem computation time, and solution quality of the basic path-based algorithm (EA) and the acceleration techniques are presented in Table 3, Table 4, and Figure 4. The notations used in Table 3, Table 4, and Figure 4 represent the following: EA - Basic path-based algorithm (MinMeanRiskPath(12)) with no acceleration technique; All_ACC- Path-based algorithm with all acceleration techniques, MMs - Path-based algorithm with multiple attacker solutions (Ms) for each attack scenario at an iteration; MTr - Path-based algorithm with trust region cut constraints (Tr) added to the master problem; MHf - Path-based algorithm with a heuristic to provide a warm start to the master problem solution; MSs - Path-based algorithm with sub-problem separation constraints added to the master problem; MNl - Path-based algorithm with a limit to the number of nodes that the master problem solver can explore in the master problem branch-and-bound tree; MTrSsNl - Path-based algorithm with trust region cut constraints, sub-problem separation constraints, and node limit method added to the master problem.

The total runtime and the master problem computation time of the basic MinMeanRiskPath algorithm (EA) and all the acceleration techniques (All_ACC) for four different sizes of attack graphs are provided in Table 3. It is seen that the computation time increases at a high rate as the size of the network increases. The computation time of the EA is especially sensitive to the network size. The basic path-based algorithm cannot solve most of the 150-node and 200- node networks within 2 hours as shown in Table 3. However, implementing the acceleration methods enables the algorithm to solve 150-node and 200-node networks within a reasonable time; therefore, with the acceleration techniques, the path-based CCG algorithm can solve large network problems within a reasonable time.

Experiments show that in the initial iterations of the algorithm, the solution of scenario sub-problems consumes more time than the master problem solution time. However, as the number of iterations increases, more variables and constraints are added to the master problem resulting an increase in the size of the master problem and thus a sharp increase in the master problem computation time. As a result, the contribution of the master problem solution time to the total runtime of the EA is larger than the contribution of the master problem solution time to the total runtime of the All_ACC method. This is likely because the acceleration methods focus on reducing only the master problem solution time. This is evident from the comparison of the Mtime columns of the exact algorithm (EA) and the accelerated algorithm (All_ACC) in Table 3.

From Table 3 we observe that the solution times of the networks of the same size varies significantly. Because the attack graphs are generated randomly which makes the topologies of the graphs to be different. These different topologies cause the solution times of the same size graphs to be different. Moreover, the difference in graph topology also affects the performance of acceleration techniques. The randomly generated probabilities of attack success on arcs also causes the computation times of the same size networks to be different. As a result, it is observed from Table 3 that in some cases the solution time of a 150-node network is more than the solution time of a 200-node networks in All_ACC method.

Table 3: Growth of the computation time (clock seconds) of Exact algorithm (EA) and All_ACC (MMsTrHfSsNl) method with graph size. Four random instances of graphs are used for each graph size. All graphs are of level = 5. Asterisk (*) denotes those graphs that cannot be solved in 2 hours. Runtime and Mtime denote total runtime, and master problem computation time, respectively, for problems with 100 scenarios.

Graph Size		All_ACC		EA	
Nodes	**Arcs**	**Runtime**	**Mtime**	**Runtime**	**Mtime**
50	118	193.07	3.72	428.89	35.61
50	126	215.58	2.77	592.70	10.66
50	119	182.72	3.22	286.72	32.13
50	116	130.24	2.07	180.59	3.66
100	237	419.17	10.77	711.66	87.13
100	235	949.60	17.17	3490.75	1306.01
100	230	1143.75	17.29	3096.35	1023.50
100	248	808.70	5.94	4074.57	676.07
150	374	1861.49	20.53	*	*
150	374	1999.04	18.64	*	*
150	345	869.49	8.62	2843.56	619.10
150	352	2089.01	16.91	*	*
200	431	3246.30	20.67	*	*
200	437	2050.04	30.72	*	*
200	425	2910.97	28.29	*	*
200	424	1783.49	27.10	*	*

The effects of different acceleration techniques to the average computation time of the basic path-based algorithm are presented in Figure 4, which demonstrates that most of the acceleration techniques contribute to reduce the average computation time of our path-based algorithm. Especially, the node limit method (MNl) reduces the average computation time of the EA by a larger percentage compared to all other acceleration techniques when applied individually. The trust region cut (MTr) reduces the EA computation time in some instances, and sometimes the computation time is higher, which slightly increases the average computation time. The sub-problem separation method (MSs) provides better reduction in the average computation time than do the master problem heuristic (MHf) and trust region cut (MTr) methods. This novel acceleration technique provides an average reduction of 18% to the master problem computation time of the EA for the 100-node networks. The average computation time of the master problem heuristic (MHf) method for the 100-node networks is larger than the EA. The average reductions in computation time with MNl, MTrSsNl, and All_ACC method are almost the same.

From the experiments, we see that the application of MMs method reduces the average computation time of the EA in solving the 50-node network. However, in solving the 100-node networks with MMs method, the computation time becomes higher than the EA time. The objective of MMs method is to add more attacker solutions to the master problem in each iteration so that the algorithm converges within a small number of iterations. However, if the size of the network is large, then a large number of variables and constraints are added to the master problem in each iteration, which increases the solution time of each iteration. In a 50-node network, the problem size is not as large as in a 100-node network, and each iteration is

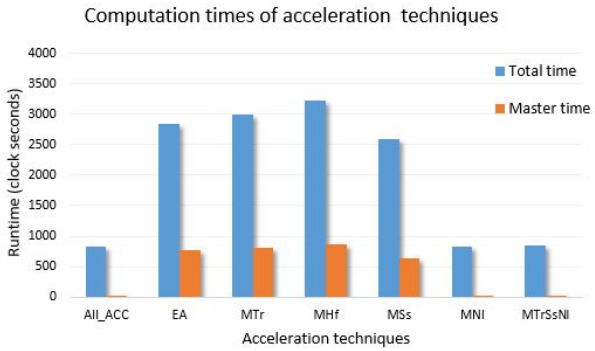


Figure 4: Average computation times (clock seconds) of the exact algorithm and the acceleration techniques for four 100-node attack graphs of level 5.

not taking longer time. Therefore, it is possible in a 50-node network to reduce the computation time by reducing the total number of iterations. As our problem is stochastic, in a 100-node network, the problem size increases significantly due to the addition of multiple solutions for each attack scenario in each iteration. Therefore, despite reducing the total number of iterations, the solution time of each iteration increases by a large amount. This causes the MMs method to take longer time than EA in solving larger networks.

Table 4 demonstrates the mean-risk expected maximum loss (MRExpLoss), master problem computation time, and the solution quality of the acceleration techniques compared to the exact algorithm. We see from Table 4 that the master problem computation times of the 100-node networks from the All_ACC method are higher than the master problem computation times from the MNl and MTrSsNl method. This is because the computation time of the MMs and MHf methods for 100-node networks are higher than the computation time of EA. Therefore, removing MMs and MHf from the All_ACC method reduces the master problem computation time of the 100-node networks, which is evident from the master problem computation times of the MTrSsNl method. However, removing multiple sub-problem solution (Ms) and the master problem heuristic (Hf) from the All_ACC method impairs the solution quality. The MMs and MHf tend to improve the poor solution resulting from the node limit method. The MHf method increases the solution quality as it provides a better start-up to the master problem solver. Therefore, although the average master problem computation time of the 100-node networks from the All_ACC method is slightly higher compared to the MNl and MTrSsNl method, the All_ACC method provides better solution quality than MNl and MTrSsNl as shown in Table 4. We also see from Table 4 that the solutions provided by the MNl, MTrSsNl, and All_ACC methods are optimal for all the 50-node network instances.

Table 5 shows the variation of computation times with the number of scenarios in the stochastic programming problem. The computation time of the algorithm increases as the number of scenarios in the problem increases.

Experiments also show that the computation time increases as the defender's budget increases. With more budget, the defender can interdict more attack paths, which requires exploring larger combination of possible attack paths and therefore increases the computation time.

Table 4: Solution quality and computation time (clock seconds) of the acceleration techniques (MNl, All_ACC, and MTrSsNl) compared to the exact algorithm (EA). Master model node limit = 1. MRExpLoss, Mtime, and %Δ denote mean-risk expected maximum loss, master problem computation time, and the percentage by which the MRExpLoss from the acceleration methods is larger than the MRExpLoss from EA, respectively.

Graph Size	**EA**		**MNl**		**All_ACC**		**MTrSsNl**	
Nodes	**MRExpLoss**	**Mtime**	**Mtime**	%Δ	**Mtime**	%Δ	**Mtime**	%Δ
50	175.87	35.61	4.39	0	3.72	0	5.67	0
50	258.88	10.65	5.66	0	2.77	0	7.05	0
50	251.63	32.13	4.94	0	3.22	0	4.55	0
50	225.82	3.66	1.68	0	2.07	0	1.02	0
100	315.16	87.13	7.12	12.49	10.77	11.47	4.68	12.47
100	426.49	1306.01	10.02	3.11	17.17	3.11	4.22	3.11
100	521.92	1023.5	5.69	12.12	17.29	1.00	6.59	10.56
100	555.65	676.07	4.51	3.17	5.94	2.65	5.31	3.17

Table 5: Increase in computation time (clock seconds) with number of scenarios. Two random instances of graphs are used for each graph size of levels = 5. Runtime represents the total runtime of the All_ACC method for 100 and 150 scenarios.

Graph Size		**Runtime**	
Nodes	**Arcs**	$\mid \mathcal{S} \mid = 100$	$\mid \mathcal{S} \mid = 150$
50	118	193.07	397.07
50	126	215.58	359.57
100	237	419.17	636.99
100	235	949.60	1285.61
150	374	1861.49	2680.61
150	374	1999.04	5477.53
200	431	3246.30	7095.72
200	437	2050.04	3498.20

5.3 Effect of Defender's Budget on Mean-Risk Expected Maximum Loss

The variation of mean-risk expected maximum loss with defender's budget for four 100-node graphs and the average mean-risk expected maximum loss of the four graphs are also shown in Figure 5. For a specific budget, the average loss is computed by simply averaging the losses of the four graphs. As the defender's budget increases, she can interdict more attack paths resulting in a lower mean-risk expected maximum loss. However, with budget increment, the loss does not decrease at a constant rate. This is because, at the beginning, the defender can overcome the relatively easy security challenges with a small budget increase after which the same budget increase is not enough to protect the next critical assets. The network defender can use Figure 5 to choose the amount she has to invest in network interdiction depending on the amount of mean-risk expected maximum loss she is willing to compromise. For example, for the attack graph 4, a network defender may choose to invest 400 units because at this point the curve levels off, resulting in a small marginal decrease in mean-risk expected maximum loss.

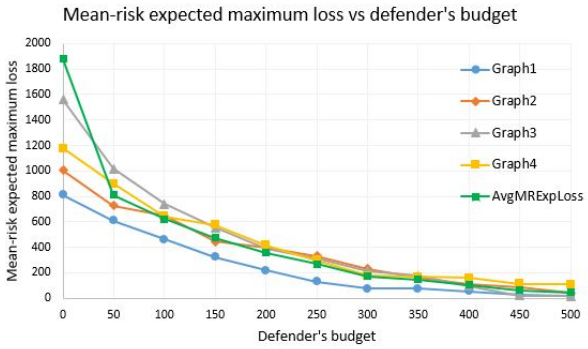


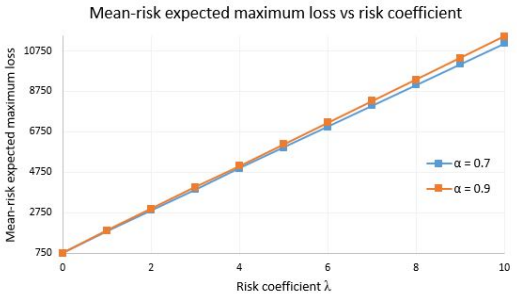
Figure 5: Variation of mean-risk expected maximum loss with defender's budget for four different graphs each with 100 -nodes.

5.4 Effects of Risk Parameters on Interdiction Policy, CVaR, and Mean-Risk Expected Maximum Loss

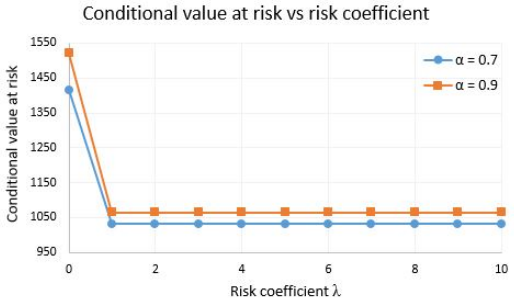
In our risk-averse stochastic network interdiction framework, we have two risk parameters: level of confidence, α, and the risk coefficient, λ. The experimental results in this section show the effects of incorporating the risk measure in minimizing the loss from cyber-attacks with a random attacker budget. Figure 6 shows the variation of mean-risk expected maximum loss, expected maximum loss, and conditional-value-at-risk with the risk coefficient (λ) for two different levels of α. Conditional-value-at-risk at a given level α quantifies the expected value of the worst $(1-\alpha)\%$ of the losses. Figure 6 demonstrates that the mean-risk expected maximum loss and the conditional-value-at-risk become larger for larger α levels. A higher level of α means that the decision maker is concerned about the realizations corresponding to larger losses. Therefore, as the α level increases, the corresponding value-at-risk increases and the $CVaR_\alpha$ increases. Thus, the larger value of α results in more conservative policies and provides more concern to the scenarios corresponding to larger losses.

The risk coefficient (λ) represents the relative importance between the expected maximum loss and the CVaR component. As the value of λ increases, the relative importance of the CVaR component increases and thus leads to more risk-averse policies. It is evident from Figure 6b that the CVaR component decreases as the value of λ increases. As the value of λ increases, relatively less importance is given to the minimization of the expectation criteria, and thus the expected maximum loss increases which is demonstrated in Figure 6c. With more risk-averse policies, the mean-risk expected maximum loss also increases which is demonstrated in Figure 6a. It is seen that as the value of λ or α increases, the mean-risk expected maximum loss increases.

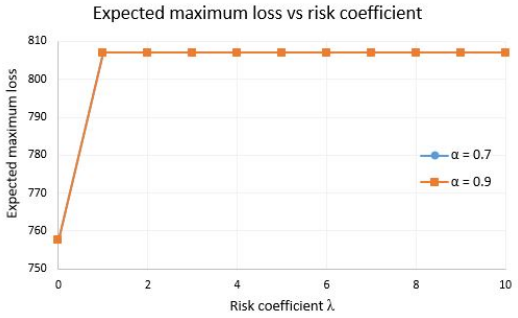
An important finding of our research is that the effect of λ parameter on $CVaR_\alpha$ depends on the distribution of the attacker budget. If the distribution is symmetric as shown in Figure 7a, the $CVaR_\alpha$ decreases by a smaller amount as the value of λ increases compared to a right-tailed asymmetric distribution. We see from Figure 7a that the $CVaR_{0.9}$ decreases by 10.56 %



(a) Mean-risk expected maximum loss vs λ



(b) Conditional-value-at-risk vs λ



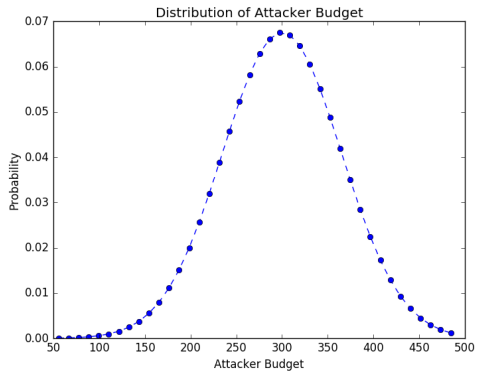
(c) Expected maximum loss vs λ. Here, the variation of expected maximum loss with λ is same for two α levels.

Figure 6: Variation of mean-risk expected maximum loss, expected maximum loss, and conditional-value-at-risk with risk coefficient λ for two different confidence levels, $\alpha = 0.7$ and $\alpha = 0.9$. Other parameters are: $|\mathcal{N}| = 50, |\mathcal{S}| = 40, b_d = 255, b_a \sim w(50, 500)$.

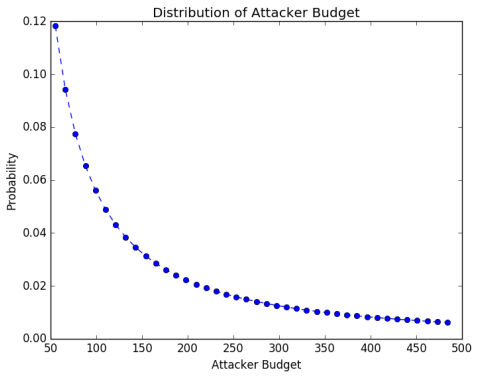
as the value of λ increases from 0 to 10. In a symmetric distribution of the attacker budget, the probability of the scenarios with large attacker budget is high. Therefore, the number of scenarios with high attacker budget is large and thus the frequency of the larger losses is also high in the distribution of losses from the scenarios. In this case, the objective of minimizing the expectation criteria is also partially taking into account the objective of minimization of the expected maximum loss from the most damaging attack scenarios. In other words, providing more importance to minimize the expected value of the larger losses separately is not highly significant as the optimal interdiction decision under a risk-neutral preference is also taking into account the minimization of larger losses.

On the other hand, if the distribution of attacker budget is a right-tailed asymmetric distribution (Weibull), the percentage decrease in $CVaR_\alpha$ becomes larger as the value of λ increases. We see from Figure 7b that for the right-tailed attacker budget distribution, the reduction in $CVaR_{0.9}$ is 29.98% as the value of λ increases from 0 to 10. In a right-tailed asymmetric distribution, the probability of the scenarios with large attacker budget is very small. Therefore, only few scenarios have an extremely large attacker budget, and the frequency of larger losses is also low. As the expectation criteria is concerned with minimizing only the expected maximum loss, in this case, the risk-neutral approach is minimizing only the expected maximum loss over all scenarios and thus is not considering minimization of the few larger loss scenarios. Therefore, as the value of λ increases, more importance is given to minimize the mean of the larger loss

scenarios, and therefore the optimal interdiction decision is made to minimize the $CVaR_\alpha$.



(a) Symmetric distribution of attacker budget. At $\lambda = 0$, $CVaR_{0.9} = 1003$ and at $\lambda = 10$, $CVaR_{0.9} = 897$



(b) Asymmetric distribution of attacker budget. At $\lambda = 0$, $CVaR_{0.9} = 1521$ and at $\lambda = 10$, $CVaR_{0.9} = 1065$

Figure 7: Variation of the effect of λ on $CVaR_\alpha$ with the distribution of attacker budget

Moreover, in reality, most cyber-attacks are conducted by attackers with low attack capability, which we represent in our problem as attacker budget. Only a few attackers have the skill to cause a severe loss to an organization, meaning that the distribution of real-world attacker capabilities is also likely to be a right-tailed asymmetric distribution. Therefore, our model better represents this real-world context.

5.5 Comparison with Deterministic and Risk-Neutral Methods

In this section, we compare the performance of our risk-averse stochastic programming approach with two existing approaches: 1) a deterministic approach that accounts for only a single attacker with a fixed budget and 2) the risk-neutral approach that ignores the risk of the most damaging cyber-attacks.

5.5.1 Comparison with Deterministic Approach

To evaluate our risk-averse approach against the existing deterministic approach, we compare our method with Nandi et al. (2016), where the authors model the network interdiction problem accounting for a single attacker with a fixed budget. This comparison answers the following research question: how much improvement in solution quality do we obtain by modeling multiple attackers via an uncertain attacker budget as opposed to a single attacker with a known fixed budget? In standard stochastic programming with a risk-neutral objective, this question is answered by computing a metric called value of stochastic solution (VSS), which measures the cost of not considering the randomness in the stochastic parameters. This VSS is computed as: $VSS = \frac{EVP-SP}{SP}$, where the EVP stands for the expected cost that results from using the expected value problem solution and SP stands for the expected cost provided by the stochastic problem solution. The expected value problem (a.k.a, mean value problem) is a deterministic

problem that takes the expected value of the random parameters and then solves the resulting deterministic program. The expected value problem of our risk-averse stochastic network interdiction model is the same as the deterministic model of Nandi et al. (2016).

As in this paper, we are concerned about risk-averse stochastic programming problem, we adopt an equivalent measure of VSS for mean-risk stochastic programs proposed by Noyan (2012). We refer to this measure as mean-risk value of stochastic solution (MRVSS), which is computed using the following formula:

$$MRVSS = \frac{MREVP - MRSP}{MRSP} \tag{15}$$

$$MREVP = \mathbb{E}(f(\bar{\mathbf{x}}(\bar{\xi}), \xi) + \lambda CVaR_{\alpha}(f(\bar{\mathbf{x}}(\bar{\xi}), \xi) \tag{16}$$

$$MRSP = \mathbb{E}(f(\mathbf{x}, \xi)) + \lambda CVaR_{\alpha}(f(\mathbf{x}, \xi)) \tag{17}$$

where $\bar{\mathbf{x}}(\bar{\xi})$ represents the optimal solution of the expected value problem, and $\mathbf{x}$ stands for the optimal solution of the stochastic problem. MREVP and MRSP represent the mean-risk expected cost resulting from using the expected value problem solution and the stochastic problem solution, respectively. The larger the value of MRVSS, the higher the significance of incorporating randomness in the parameters of a risk-averse stochastic programming model with a specified risk measure compared to solving the deterministic problem.

We conduct computational experiments to explore the significance of incorporating uncertainty (randomness) in the attacker budget in our mean-risk network interdiction model and thus evaluate the performance of our model against Nandi et al. (2016). As the attackers have different budget values, an interdiction plan generated based on a fixed attacker budget (or the expected attacker budget) is likely to be erroneous and results a positive MRVSS. Table 6 demonstrates the MRVSS for different values of λ and α, where we see that the value of stochastic solution for the risk neutral approach ($\lambda = 0$) is small. This means that modeling randomness in the attacker budget using a risk-neutral approach does not provide a more robust interdiction plan than the deterministic problem. In other words, the risk-neutral special case of our model provides small improvement in reducing the loss from cyber-attacks compared to the deterministic approach of Nandi et al. (2016).

However, in more risk-averse circumstances, a risk-averse approach results a larger MRVSS than a risk-neutral approach, as shown in Table 6. We see that the MRVSS increases as the risk parameters λ and α increases. As the risk coefficient (λ) increases, meaning preferences for risk-aversion increases, the deterministic model performs poorly compared to the risk-averse stochastic programming model. In other words, as the decision maker becomes more risk-averse, the cost of ignoring uncertainty in attacker budget increases, and thus the significance of solving a mean-risk stochastic model increases. This is because, unlike the deterministic and risk-neutral model, with higher risk-aversion preferences, the risk-averse model prioritize minimizing the expected value of the larger losses more compared to minimizing the overall expected loss. This results in a lower mean-risk expected maximum loss in using a risk-averse solution as opposed to using a deterministic or a risk-neutral solution. Therefore, in the presence

of uncertainty in attacker capabilities, a risk-averse approach provides a more robust interdiction decision than a deterministic as well as a risk-neutral approach. Thus we can claim that our risk-averse approach significantly outperforms the deterministic approach of Nandi et al. (2016), particularly in a risk-averse situation.

Table 6: Variation of MRVSS with risk parameters. Other parameters are: $|\mathcal{N}| = 50, |\mathcal{S}| = 40, b_d = 255, b_a \sim w(50, 500)$.

Risk Coefficient (λ)	**MRVSS (%)**	
	$\alpha = \mathbf{0.7}$	$\alpha = \mathbf{0.9}$
0	6.74	6.74
0.1	6.79	6.98
0.5	16.13	19.19
1	23.01	27.40
5	35.08	41.82
10	37.56	44.78

5.5.2 Comparison with Risk-Neutral Approach

We evaluate the performance of our risk-averse stochastic programming approach with the risk-neutral approach that does not account for the risk of most damaging cyber-attacks while computing the optimal interdiction policy. To demonstrate the cost of ignoring the risk of larger losses from the most damaging cyber-attacks, when the network defender is actually risk-averse, we introduce a metric called value of risk-aversion (VRA) and compute using the following formula:

$$VRA = \frac{RNSP - MRSP}{MRSP} \tag{18}$$

$$RNSP = \mathbb{E}(f(\mathbf{x}^{\mathrm{RN}}, \xi)) + \lambda CVaR_{\alpha}(f(\mathbf{x}^{\mathrm{RN}}, \xi)) \tag{19}$$

where $\mathbf{x}^{\mathrm{RN}}$ and RNSP stands for the optimal solution of the risk-neutral stochastic program and the mean-risk expected maximum loss resulting from using $\mathbf{x}^{\mathrm{RN}}$ in a risk-averse situation, respectively. Table 7 shows the variation of VRA with the risk parameters– risk coefficient, λ and level of confidence, α, where we see that the VRA increases as the value of λ and α increases. This means that as the network defender becomes more risk-averse, the risk-neutral stochastic programming solution performs increasingly worse compared to the mean-risk stochastic programming solution. With a larger λ, the network defender is more risk-averse; thus, using a risk-neutral solution cannot minimize the expected value of the larger losses (CVaR), and results a higher mean-risk expected maximum loss. On the other hand, as a risk-averse model prioritizes minimizing the risk of larger losses while computing the optimal interdiction policy, the resulting risk-averse solution can better minimize the expected value of the larger losses in addition to minimizing the expected maximum loss over all scenarios. Though the risk-averse solution results in an increase in the expected maximum loss over all scenarios (Figure 6c) when compared to a risk-neutral solution, the overall objective value (MRExpLoss) from using a risk-averse solution is much smaller as opposed to that of risk-neutral solution. This is because

a risk-averse solution minimizes the CVaR component by a larger percentage (Figure 6b), offsetting the increase in expected maximum loss. Thus, our results indicate that it is important to use a risk-averse model to model a risk-averse network defender; a risk-neutral model is not an adequate proxy.

With a larger α, the network defender is concerned about minimizing the larger loss scenarios, which requires them to use a risk-averse solution. Therefore, the significance of using a risk-averse solution over a risk-neutral solution increases as α increases. Thus, in the presence of uncertainty and high variability in attacker budget, a risk-averse model provides a more robust solution than a risk-neutral one.

Table 7: Variation of VRA with the risk parameters. Other parameters are: $|\mathcal{N}| = 50, |\mathcal{S}| = 40, b_d = 255, b_a \sim w(50, 500)$.

Risk Coefficient (λ)	**VRA** (%)	
	$\alpha = \mathbf{0.7}$	$\alpha = \mathbf{0.9}$
0.1	0	0
0.5	9.90	12.97
1	16.88	21.33
5	29.03	35.87
10	31.53	38.86

6 Conclusion

We studied the problem of optimally interdicting the cyber network of an organization from the standpoint of a risk-averse network defender. In this paper, we presented a mean-risk bi-level stochastic network interdiction model based on the concept of an attack graph. We adopted the conditional-value-at- risk as a risk measure in our risk-averse model. Both inner and outer levels of our model were formulated as mixed-integer linear programs. We developed a customized constraint and column generation algorithm to incorporate the stochasticity and risk-aversion. Several novel enhancement techniques were proposed to improve the computational efficiency of the base algorithm. We also employed a heuristic to provide a warm start to the master problem solver.

Computational experiments show that the acceleration techniques significantly improve runtime. Implementing all the acceleration techniques together (All_ACC) provides better computational speed than the individual techniques and better solution quality than the master problem node limit method (MNl) and the master problem with trust region cut, sub-problem separation, and node limit added together (MTrSsNl). Though MNl and MTrSsNl provides better reduction of the master problem computation time than the All_ACC method, their solution quality is poor. When applied individually, the master problem node limit method provides the best computational efficiency of all the acceleration techniques. However, this node limit method also impairs the solution quality. The novel acceleration technique, sub-problem separation, provides an average reduction of 18% to the master problem solution time without affecting the solution quality. In the accelerated algorithm (All_ACC method), the major contribution to the total computation time is from the sub-problem solution time. This is because, the acceleration

techniques only reduces the master problem solution time. Solution time increases with network size and number of scenarios.

We also found that the mean-risk expected maximum loss decreases with the defender's budget. However, the rate of change is not uniform with the defender's budget. The mean-risk expected maximum loss decreases sharply at the beginning when it is easy for the defender to overcome the relatively easy security challenges with a small increment of budget.

Our experiments provide insights into the effects of risk-aversion on the optimal interdiction decision. The larger the value of the risk parameters, the more conservative the interdiction policies. As the risk coefficient (λ) increases, the optimal interdiction plan is generated by assigning relatively more importance to minimizing the expected maximum loss from the most damaging attack scenarios instead of minimizing expected maximum loss over all scenarios, and thus conditional-value-at-risk at level α decreases. However, the effect of λ on optimal interdiction policy and in turn on conditional-value-at-risk at level α depends on the distribution of attacker budget. The effect of λ on conditional-value-at-risk is more evident when a few attackers have extremely high budget, i.e., the distribution is heavy right-tailed. As the value of α increases, the decision maker is more concerned with minimizing the larger loss scenarios, leading to an increase in the value-at-risk and, in turn, increases the conditional-value-at-risk.

Taken together, the experimental results demonstrate the significance of modeling uncertainty in attacker budget through a mean-risk stochastic model. Modeling multiple attackers via uncertain attacker budget in a risk-averse stochastic programming model provides more robust interdiction decision than that of a deterministic model with a single attacker. Results demonstrate that our risk-averse stochastic programming approach substantially outperforms the existing deterministic approach of Nandi et al. (2016). The mean-risk value of stochastic solution increases as the decision maker becomes more risk-averse. Results also show that the risk-averse model provides substantially robust interdiction decision than the risk-neutral model. The value of risk-aversion increases as the network defender becomes more risk-averse.

6.1 Future Work

In this study, we assumed perfect interdiction– no attack is possible if a countermeasure is deployed on an arc. However, this assumption could be relaxed to incorporate imperfect, multi-level interdiction, where the higher the investment in countermeasure deployment on an arc, the lower the probability of success of attack through that arc. This could be incorporated by formulating the model as a stochastic program with decision–dependent uncertainty. In this context, the first-stage decision of investing on countermeasure deployment affects the probability of success of attacks and in turn affects the scenario probability. For details of this decision–dependent uncertainty framework, we refer readers to Medal et al. (2016); Bhuiyan et al. (2019). From a computational standpoint, we implemented enhancements to reduce the master problem solution time only. Another extension could be to reduce the overall sub-problem solution time. One way of doing this is to implement a progressive hedging algorithm that is easy to parallelize. Also, the proposed model and algorithms could be implemented on real cyber networks rather than synthetic ones.

References

Ahmed, S. (2006). Convexity and decomposition of mean-risk stochastic programs. *Mathematical Programming*, 106(3):433–446.

Alderson, D. L., Brown, G. G., and Carlyle, W. M. (2014). Assessing and improving operational resilience of critical infrastructures and other systems. *Tutorials in Operations Research*, pages 180–215.

Alhomidi, M. and Reed, M. (2013). Finding the minimum cut set in attack graphs using genetic algorithms. In *2013 International Conference on Computer Applications Technology*, pages 1–6. IEEE.

Allain, R. J. (2016). An evolving asymmetric game for modeling interdictor-smuggler problems. Master's thesis, Naval Postgraduate School.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3):203–228.

Bernard, T. S., Hsu, T., Perlroth, N. P., and Lieber, R. (2017, September 07). Equifax Says Cyberattack May Have Affected 143 Million in the U.S. *The New York Times*, Retrieved from https://www.nytimes.com/2017/09/07/business/equifax-cyberattack.html.

Berr, J. (2017, May 16). "WannaCry" ransomware attack losses could reach $ 4 billion. *CBS News*, Retrieved from https://www.cbsnews.com/news/wannacry-ransomware-attacks-wannacry-virus-losses/.

Bhuiyan, T. H., Moseley, M. C., Medal, H. R., Rashidi, E., and Grala, R. K. (2019). A stochastic programming model with endogenous uncertainty for incentivizing fuel reduction treatment under uncertain landowner behavior. *European Journal of Operational Research*, 277(2):699–718.

Bhuiyan, T. H., Nandi, A. K., Medal, H., and Halappanavar, M. (2016). Minimizing expected maximum risk from cyber-attacks with probabilistic attack success. In *2016 IEEE Symposium on Technologies for Homeland Security*, pages 1–6. IEEE.

Bistarelli, S., Fioravanti, F., and Peretti, P. (2006). Defense trees for economic evaluation of security investments. In *First International Conference on Availability, Reliability and Security*, pages 416–423. IEEE.

Brown, G. G., Carlyle, W. M., Harney, R. C., Skroch, E. M., and Wood, R. K. (2009). Interdicting a nuclear-weapons project. *Operations Research*, 57(4):866–877.

Collado, R., Meisel, S., and Priekule, L. (2017). Risk-averse stochastic path detection. *European Journal of Operational Research*, 260(1):195–211.

Dewri, R., Poolsappasit, N., Ray, I., and Whitley, D. (2007). Optimal security hardening using multi-objective optimization on attack tree models of networks. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 204–213. ACM.

Dewri, R., Ray, I., Poolsappasit, N., and Whitley, D. (2012). Optimal security hardening on attack tree models of networks: a cost-benefit analysis. *International Journal of Information Security*, 11(3):167–188.

Durkota, K., Lisỳ, V., Bošanskỳ, B., and Kiekintveld, C. (2015a). Approximate solutions for attack graph games with imperfect information. In Khouzani, M., Panaousis, E., and Theodorakopoulos, G., editors, *Decision and Game Theory for Security*, pages 228–249. Springer.

Durkota, K., Lisý, V., Bošanský, B., and Kiekintveld, C. (2015b). Optimal network security hardening using attack graph games. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 526–532.

Gurobi Optimization Inc. (2017). Gurobi optimizer reference manual. URL: http://www. gurobi. com.

Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160.

Israeli, E. and Wood, R. K. (2002). Shortest-path network interdiction. *Networks*, 40(2):97–111.

Jiang, J. and Liu, X. (2018). Multi-objective stackelberg game model for water supply networks against interdictions with incomplete information. *European Journal of Operational Research*, 266(3):920–933.

Lei, X., Shen, S., and Song, Y. (2018). Stochastic maximum flow interdiction problems under heterogeneous risk preferences. *Computers & Operations Research*, 90:97–109.

Liberatore, F., Scaparra, M. P., and Daskin, M. S. (2011). Analysis of facility protection strategies against an uncertain number of attacks: the stochastic r-interdiction median problem with fortification. *Computers & Operations Research*, 38(1):357–366.

Medal, H. R., Pohl, E. A., and Rossetti, M. D. (2016). Allocating protection resources to facilities when the effect of protection is uncertain. *IIE Transactions*, 48(3):220–234.

Moore, J. T. and Bard, J. F. (1990). The mixed integer linear bilevel programming problem. *Operations Research*, 38(5):911–921.

Morton, D. P., Pan, F., and Saeger, K. J. (2007). Models for nuclear smuggling interdiction. *IIE Transactions*, 39(1):3–14.

Mousavian, S., Valenzuela, J., and Wang, J. (2015). A probabilistic risk mitigation model for cyber-attacks to PMU networks. *IEEE Transactions on Power Systems*, 30(1):156–165.

Nandi, A. K. and Medal, H. R. (2016). Methods for removing links in a network to minimize the spread of infections. *Computers & Operations Research*, 69:10–24.

Nandi, A. K., Medal, H. R., and Vadlamani, S. (2016). Interdicting attack graphs to protect organizations from cyber attacks: A bi-level defender–attacker model. *Computers & Operations Research*, 75:118–131.

Nehme, M. V. (2009). *Two-person games for stochastic network interdiction: models, methods, and complexities*. PhD thesis, The University of Texas at Austin.

Nguyen, T. H., Wright, M., Wellman, M. P., and Singh, S. (2018). Multistage attack graph security games: Heuristic strategies, with empirical game-theoretic analysis. *Security and Communication Networks*, 2018:1–28.

Noel, S. and Jajodia, S. (2008). Optimal IDS sensor placement and alert prioritization using attack graphs. *Journal of Network and Systems Management*, 16(3):259–275.

Noyan, N. (2012). Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39(3):541–559.

O´ Hanley, J. R. and Church, R. L. (2011). Designing robust coverage networks to hedge against worst-case facility losses. *European Journal of Operational Research*, 209(1):23–36.

Pan, F. and Morton, D. P. (2008). Minimizing a stochastic maximum-reliability path. *Networks*, 52(3):111–119.

Pay, B. S., Merrick, J. R., and Song, Y. (2019). Stochastic network interdiction with incomplete preference. *Networks*, 73(1):3–22.

Phillips, C. and Swiler, L. P. (1998). A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 Workshop on New Security Paradigms*, pages 71–79. ACM.

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

Qiao, J., Jeong, D., Lawley, M., Richard, J.-P. P., Abraham, D. M., and Yih, Y. (2007). Allocating security resources to a water supply network. *IIE Transactions*, 39(1):95–109.

Reed, B. K. (1994). Models for proliferation interdiction response analysis. Master's thesis, Naval Postgraduate School.

Roy, A., Kim, D. S., and Trivedi, K. S. (2010). Cyber security analysis using attack countermeasure trees. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, pages 28–31. ACM.

Salmeron, J., Wood, K., and Baldick, R. (2004). Analysis of electric grid security under terrorist threat. *IEEE Transactions on Power Systems*, 19(2):905–912.

Salmeron, J., Wood, K., and Baldick, R. (2009). Worst-case interdiction analysis of large-scale electric power grids. *IEEE Transactions on Power Systems*, 24(1):96–104.

Scaparra, M. P. and Church, R. L. (2008). A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, 35(6):1905–1923.

Schultz, R. and Tiedemann, S. (2006). Conditional value-at-risk in stochastic programs with mixed-integer recourse. *Mathematical Programming*, 105(2-3):365–386.

Serra, E., Jajodia, S., Pugliese, A., Rullo, A., and Subrahmanian, V. (2015). Pareto-optimal adversarial defense of enterprise systems. *ACM Transactions on Information and System Security*, 17(3):11:1–11:39.

Shields, T. and Newcomer, E. (2018, April 12). Uber's 2016 Breach Affected More Than 20 Million U.S. Users. *Bloomberg*, Retrieved from https://www.bloomberg.com/news/articles/2018-04-12/uber-breach-exposed-names-emails-of-more-than-20-million-users.

Song, Y. and Shen, S. (2016). Risk-averse shortest path interdiction. *INFORMS Journal on Computing*, 28(3):527–539.

Sullivan, K. M., Morton, D. P., Pan, F., and Cole Smith, J. (2014). Securing a border under asymmetric information. *Naval Research Logistics*, 61(2):91–100.

Toyoda, Y. (1975). A simplified algorithm for obtaining approximate solutions to zero-one programming problems. *Management Science*, 21(12):1417–1427.

Wood, R. K. (1993). Deterministic network interdiction. *Mathematical and Computer Modelling*, 17(2):1–18.

Xiao, K., Zhu, C., Xie, J., Zhou, Y., Zhu, X., and Zhang, W. (2018). Dynamic defense strategy against stealth malware propagation in cyber-physical systems. In *IEEE Conference on Computer Communications*, pages 1790–1798. IEEE.

Zhang, R., Zhu, Q., and Hayel, Y. (2017). A bi-level game approach to attack-aware cyber insurance of computer networks. *IEEE Journal on Selected Areas in Communications*, 35(3):779–794.

Zhang, Z., Lv, K., and Hu, C. (2018). Establishing an optimal network defense system: A Monte Carlo graph search method. In Liu, F., Xu, S., and Yung, M., editors, *Science of Cyber Security*, pages 181–190. Springer.

Zheng, K. and Albert, L. A. (2019). A robust approach for mitigating risks in cyber supply chains. *Risk Analysis*, (In press).

Zheng, K., Albert, L. A., Luedtke, J. R., and Towle, E. (2019). A budgeted maximum multiple coverage model for cybersecurity planning and management. *IISE Transactions*, (In press).

Zonouz, S. A., Khurana, H., Sanders, W. H., and Yardley, T. M. (2014). RRE: A game-theoretic intrusion response and recovery engine. *IEEE Transactions on Parallel and Distributed Systems*, 25(2):395–406.

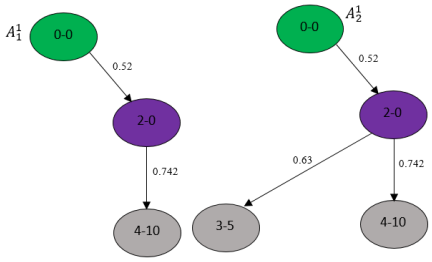
Appendices

A Algorithm Example

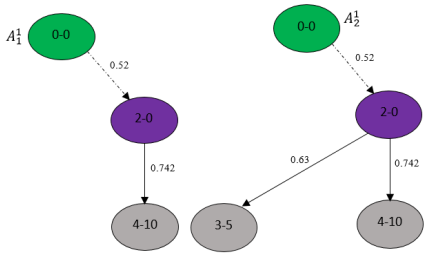
To demonstrate the solution algorithm, we describe the steps of the algorithm with the simple example attack graph shown in Figure 3, where we consider a risk-averse network defender with a budget of 1 unit. We consider two attackers with budgets of 2 and 3 units, respectively. The attack cost and countermeasure deployment cost is 1 unit for all arcs. The risk coefficient and the level of confidence are assumed to be 0.1 unit and 90 percent, respectively. Figure A.1 demonstrates the steps of the algorithm, where each of the sub-Figures of Figure A.1 represents either the attacker's solution or the defender's solution. The dashed arcs represent the arcs interdicted by the defender.

At iteration 1, there is no interdiction of arcs, the two attackers choose their optimal attack plans to maximize the loss to the defender. Attacker 1 use the set of arcs {(0, 2), (2, 4)} and the attacker 2 use the set of arcs {(0, 2), (2, 3), (2, 4)} as shown in Figure A.1a. These attack plans of the two attackers result in a mean-risk expected maximum loss of 5.29 ($= 0.5 \times 3.86 + 0.5 \times 5.496 + 0.1(5.33 + \frac{1}{1-0.9} \times 0.5 \times 0.17)$) to the defender, which is the new upper bound. Now, at the beginning of iteration 2, the defender takes into account the previous attack plans of the two attackers and generates an interdiction plan that minimizes the mean-risk expected maximum loss. Here, the optimal interdiction plan of the defender is to interdict arc (0, 2) as shown in Figure A.1b, which protects the defender from incurring any loss. Given that arc (0, 2) is interdicted, the two attackers choose new attack plans as shown in Figure A.1c that results in a new upper bound of 3.22.

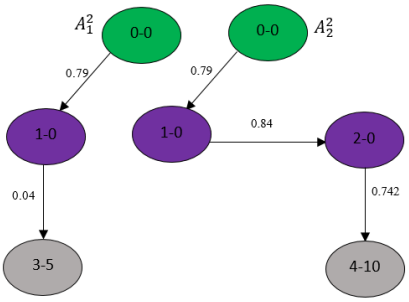
At iteration 3, the defender observes all the previous attack plans of iteration 1 and of iteration 2 and interdict arc (2, 4), which results in a lower bound of 0.98. We see that as the algorithm proceeds, the upper and lower bounds of the algorithm are updated gradually. This process continues until the bounds of the algorithm converges to an optimal objective value, which is 2.36 resulting from interdicting arc (2, 4).



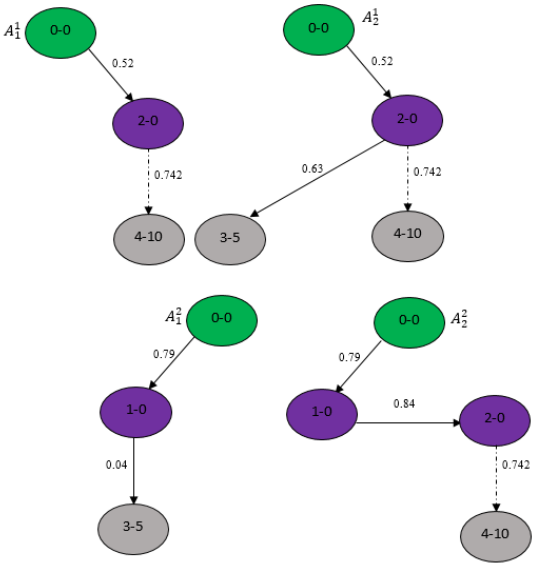
(a) Iteration 1: Attacker's solution. No interdiction, the attack plans of two attackers result in an upper bound = 5.29



(b) Iteration 2: Defender's solution. Optimal interdiction results in a lower bound = 0



(c) Iteration 2: Attacker's solution. New attack plans results the new upper bound = 3.22



(d) Iteration 3: Defender's solution.
New interdiction plan results in a lower bound = 0.98

Figure A.1: Algorithm Example

B Proofs

Lemma 1. *The master problem produces a new solution at each iteration until convergence.*

Proof. We know that at each iteration of the algorithm, a set of $|\mathcal{S}|$ attack plans are generated and added to the master problem, where $s \in \mathcal{S}$ is the scenario index. Assume that the master problem solution $\hat{\mathbf{x}}^k$ at iteration k and the master problem solution $\hat{\mathbf{x}}^m$ at iteration m are the same, where $m < k$. In this case, the set of the sub-problem solutions $\hat{\mathbf{f}}^m$ from iteration m repeats at iteration k. Therefore, the optimal attack plans $\hat{\mathbf{f}}^{ms}$ in the set of sub-problem solutions from iteration m are not interdicted by the master problem solution at iteration k. Now, the master problem objective value $f(\bar{\mathcal{A}}^k)$ at iteration k is at least as large as the mean-risk expected maximum loss $Q(\hat{\mathbf{x}}^m)$ from the set of sub-problem solutions at iteration m. As the current $lb \geq f(\bar{\mathcal{A}}^k) \geq Q(\hat{\mathbf{x}}^m)$ and current $ub \leq Q(\hat{\mathbf{x}}^m)$, $lb \geq ub$ which is the convergence criteria of the algorithm. Therefore, at each iteration, the master problem must generate a new solution to interdict the set of attack plans generated until convergence. □

Theorem 2. *The CCG algorithm converges within a finite number of iterations.*

Proof. From Lemma 1, we see that at each iteration of the algorithm, the master problem generates a new feasible interdiction plan. Suppose there are $\mathbb{Z}$ feasible interdiction plans. The algorithm will continue through a maximum of $\mathbb{Z}$ iterations and add all feasible sets of attack plans. Thus, running through $\mathbb{Z}$ iterations, the algorithm will explore all the possible attack plans, implying that the algorithm converges within $\mathbb{Z}$ iterations. □