A budgeted maximum multiple coverage model for cybersecurity planning and management

Kaiyue Zheng
z.kaiyue0421@gmail.com
Amazon, Seattle, WA

Laura A. Albert*, James R. Luedtke, Eli Towle
laura@engr.wisc.edu, jim.luedtke@wisc.edu, etowle@wisc.edu
Department of Industrial and Systems Engineering
University of Wisconsin–Madison

Abstract

This paper studies how to identify strategies for mitigating cyber-infrastructure vulnerabilities. We propose an optimization framework that prioritizes the investment in security mitigations to maximize the coverage of vulnerabilities. We use multiple coverage to reflect the implementation of a layered defense, and we consider the possibility of coverage failure to address the uncertainty in the effectiveness of some mitigations. Budgeted maximum multiple coverage (BMMC) problems are formulated, and we demonstrate that the problems are submodular maximization problems subject to a knapsack constraint. Other variants of the problem are formulated given different possible requirements for selecting mitigations, including unit cost cardinality constraints and group cardinality constraints. We design greedy approximation algorithms for identifying near-optimal solutions to the models. We demonstrate an optimal $(1 - 1/e)$-approximation ratio for BMMC and a variation of BMMC that considers the possibility of coverage failure, and a 1/2-approximation ratio for a variation of BMMC that uses a cardinality constraint and group cardinality constraints. The computational study suggests that our models yield robust solutions that use a layered defense and provide an effective mechanism to hedge against the risk of possible coverage failure. We also find that the approximation algorithms efficiently identify near-optimal solutions, and that a Benders branch-and-cut algorithm we propose can find provably optimal solutions to the vast majority of our test instances within an hour for the variations of the proposed models that consider coverage failures.

Keywords: cyber-security; submodular optimization; coverage models; critical infrastructure protection

1 Introduction

The information systems in the United States rely on a fragile information technology (IT) infrastructure that is vulnerable to numerous security risks. According to a U.S. Government Account-

*Corresponding author

ability Office (GAO) report, threats to the federal IT supply chains include installation of counterfeit hardware or software, disruption in the production or distribution of a critical product, reliance on malicious or unqualified service providers, and poorly trained employees [1]. Recently, reports from various governing agencies have expressed similar concerns, highlighting the importance of enhancing the security of cyber-infrastructure and the federal IT supply chain. The Director of National Intelligence listed managing the "enormous vulnerabilities within the IT supply chain" as one of the U.S.'s top two cybersecurity challenges [2]. Several U.S. GAO reports echo this sentiment [1, 3, 4]. The White House proposed new policy directives for securing critical IT physical assets, signaling growing awareness of the increasing role of cybersecurity in critical infrastructure [5–7]. The White House has also started to direct federal funding toward global supply chain risk management [8]. These reports underscore the need to invest in security mitigations that can enhance the security of cyber-infrastructure and the federal IT supply chain in a cost-effective manner.

An effective way to improve the trustworthiness of the federal IT infrastructure is to deploy security mitigations to reduce vulnerabilities in the supply chain life cycle [9]. Cybersecurity mitigations are countermeasures implemented by the decision maker to proactively protect the system against attacks. Examples of cybersecurity mitigations include replacing vulnerable physical components of the IT infrastructure, requiring suppliers to provide tamper protection for certain hardware components, establishing and implementing access control procedures, storing sensitive information at alternative sites, and improving security training for employees. Mitigations "cover" vulnerabilities in the system by increasing the difficulty of attacks or reducing their consequences. Some mitigations may work in similar ways and may cover the same vulnerabilities. Federal organizations have limited budgets for deploying mitigations. Although efforts have been made toward assessing cybersecurity risks to the U.S. [5, 6], comprehensive security policies and mitigations have not been developed and implemented [4]. The uncertainty in mitigation effectiveness and the interplay between different mitigations make it very challenging to determine which combination of mitigations yields the maximum benefit given a limited budget.

Recently, the National Institute of Standards and Technology (NIST) published supply chain risk management practices for improving the security of federal information systems and organizations [10]. NIST proposes a threat scenario analysis framework that identifies vulnerabilities and mitigations to address them. They demonstrate how this generic framework has practical applications, including telecommunications counterfeiting and industrial espionage. The NIST framework

provides decision makers with cost-effective strategies for risk reduction. Given the considerable risks associated with federal information systems, NIST suggests that decision makers manage supply chain risk with a more structured approach that uses well-defined goals and scope to represent threat scenarios.

In this paper, we address the critical challenge of structured supply chain risk management by proposing an optimization framework for implementing security mitigations to cover supply chain vulnerabilities. This optimization framework supports a formal supply chain risk assessment framework by prescribing procedures that can reduce risk. We are motivated by a federal planning problem in which trustworthy computing processes are designed for a long-term phased rollout [11, 12]. In this paper, we do not consider real-time decisions for intrusion detection, response protocols for security incidents, or acquisition decisions. The results of this paper are of interest to federal decision makers responsible for managing policies and investments in IT supply chain security.

We propose and study new models that account for the effect of multiple coverage and the possibility of coverage failure. The first of these is a model for the deterministic budgeted maximum multiple coverage problem (BMMC). The multiple coverage feature of this model provides an incentive to implement complementary mitigations to reduce vulnerability. We extend this model to the expected-value budgeted maximum multiple coverage problem (EBMMC). This model seeks a set of mitigations under the same conditions as BMMC while considering uncertainty in mitigation effectiveness. Such uncertainties can arise if the effectiveness of mitigations relies on historical evidence or the judgment of subject matter experts (SMEs). Additionally, we consider model variations with other restrictions on mitigation selection, such as limiting decision makers to choosing no more than one mitigation from each of a number of pre-specified groups (group cardinality constraints). We demonstrate that the proposed models are submodular maximization problems subject to a linear or matroid constraint. We propose approximation algorithms for identifying near-optimal solutions to the models. The algorithms use a number of objective function evaluations that is polynomial in the number of available mitigations. We also propose a Benders branch-and-cut algorithm for exactly solving the expected-value version of this problem.

At present, there are few optimization models that study cybersecurity risk mitigation using strategic planning or investment. Leskovec et al. present an optimization model for cost-effective detection of outbreaks in networks [13]. The model has applications to detecting the spread of contaminants in a physical network or malicious ideas in a social network. Afful-Dadzie and Allen propose a

Markov decision process model to manage data-driven maintenance policies for IT network security when data are scarce [14]. Zhuo and Solak propose a stochastic programming model to optimize a firm's cybersecurity budget in an investment portfolio [15]. Their model addresses uncertainty in the effectiveness of the countermeasures. Nagurney et al. propose game theoretic models to manage vulnerability in electronic Internet transactions [16]. In this model, retailers attempt to maximize their profits by determining the security of transactions and the product. Meanwhile, consumers adjust the price they are willing to pay for the product based on demand and supply chain security. In a more recent paper, Nagurney and Shukla [17] propose both competitive and cooperative game theoretic models to investigate the value of information sharing for firms.

The models proposed in this paper generalize the maximum coverage problem, where k facilities are located to maximize the overall coverage of the facilities [18]. Coverage models are applicable to a wide variety of problems, including allocation of emergency response resources [19, 20]. They are also relevant to homeland security applications; for example, coverage models have been used to optimally screen checked baggage on commercial aviation flights [21, 22]. A comprehensive study of coverage models and their applications can be found in Daskin [23]. The maximum coverage problem is NP-hard and is an instance of monotone submodular maximization subject to a cardinality constraint. Nemhauser et al. show that a greedy heuristic achieves an approximation ratio of $(1-1/e)$ for this class of problems [24]. The budgeted maximum coverage problem (BMC) generalizes the maximum coverage problem by replacing the cardinality constraint with a knapsack constraint. Khuller et al. show that a greedy heuristic achieves an approximation ratio of $(1-1/\sqrt{e})$ [25]. This can be improved to an optimal $(1-1/e)$-approximation ratio by incorporating the greedy heuristic into a partial enumeration scheme. Sviridenko generalizes these results to any submodular maximization problem subject to a knapsack constraint [26]. Fisher et al. show that a greedy heuristic achieves an approximation ratio of $1/2$ when maximizing a monotone submodular set function subject to a matroid constraint [27]. This result is improved to an optimal $(1-1/e)$-approximation factor by Vondrak [28] and Calinescu et al. [29] by adapting a continuous greedy algorithm and a pipage rounding procedure [30]. Two recent papers by Badanidiyuru and Vondrak [31] and Buchbinder et al. [32] propose faster algorithms with similar approximation ratios. For the study of approximation algorithms for general submodular maximization subject to multiple knapsack constraints, we refer the readers to Kulik et al. [33].

In summary, this paper makes the following contributions:

1. We propose new coverage models, BMMC and EBMMC, for selecting an optimal portfolio of security mitigations to reduce threat vulnerability. These models facilitate a layered defense of cyber-infrastructure and federal IT supply chains. EBMMC addresses the uncertainty of mitigation effectiveness.

2. We demonstrate that the proposed coverage models are submodular maximization problems subject to a knapsack constraint. We formulate variants of BMMC and EBMMC and demonstrate that these models are also specific instances of submodular maximization problems.

3. We present polynomial-time heuristics for identifying near-optimal solutions to the proposed models. We demonstrate an optimal $(1 - 1/e)$-approximation ratio for BMMC and EBMMC and a 1/2-approximation ratio for the BMMC and EBMMC variants that incorporate a cardinality constraint and group cardinality constraints. We also propose a Benders branch-and-cut algorithm for solving the expected-value versions of the proposed models.

4. We perform a computational study to compare the solutions and runtimes of approximation algorithms to methods for solving the proposed methods to optimality. The greedy heuristics identify near-optimal solutions quickly. On the other hand, the proposed Benders branch-and-cut is also able to solve large instances of the expected-value problem to optimality, providing an option for modelers wishing to solve the proposed models exactly. We demonstrate the practical value of the models by comparing their effectiveness in cases where vulnerabilities can be covered multiple times or mitigations could fail.

5. We discuss how to modify the approximation algorithms to produce a set of naturally "nested" solutions that can be used by decision makers to weigh the trade-offs between cost and vulnerability reduction. This set of solutions can be efficiently generated.

This paper is organized as follows. We define the problems and their mixed-integer linear programming (MIP) formulations in Section 2. In Section 3, we show that all of the problems are monotone submodular maximization problems subject to a linear or matroid constraint. We introduce polynomial-time algorithms with constant approximation ratios. In Section 4, we propose a Benders branch-and-cut algorithm for solving the stochastic versions of the formulations. In Section 5, we conduct a numerical study to test the practical efficiency and solution quality of the proposed algorithms. Section 6 concludes the paper.

2 Problem description

In this section, we introduce several optimization models for prioritizing mitigations to protect IT supply chains. Cyber-attacks exploit vulnerable components of the supply chain network. We explicitly consider the steps taken to carry out a complete attack on system vulnerabilities. Accordingly, we formulate attacks as "attack paths," each of which contains multiple nodes representing vulnerabilities. We do not consider the sequence of steps (nodes) on a path to be important. We instead focus on preventing the success of each attack when considered as an entire attack path. We also assume different attack paths may contain the same vulnerability. For example, hardware may contain a vulnerability that can be exploited in a variety of ways.

Attack modeling is an important first step of our problem. In traditional cyber-vulnerability analysis, threats are modeled using attack trees. Introduced by Schneier [34] and formalized by Mauw and Oostdijk [35], attack trees are used to characterize the possible attacks against a system and identify protections against such attacks. A node of an attack tree represents the event of an attack, and an arc represents the transition of states after a step of an attack occurs. The root node represents the target of the attack. Each path in the tree from a leaf node to the root node defines an attack path. The level of aggregation and exact construction of an attack tree depends on the scope of the risk analysis conducted by the SMEs [11]. An attack path is often constructed manually based on SME knowledge of the attack profiles. This process is tedious and error-prone [12]. As such, methods for automatically generating attack trees have been explored (e.g., [36]). We refer the readers to Shostack for a recent survey on attack modeling and for examples of attack trees and mitigations [37].

In our models, we assume the set of attacks paths is enumerated and given as a complete list. Although the number of attack paths may be very large in general, many classes of IT attacks must be carried out in very specific ways, resulting in a manageable number of enumerated paths. Work with collaborators at Sandia National Laboratory (SNL) suggests that this assumption is reasonable for our application, as the sizes of the attack trees are expected to be moderate. Given a ground set of vulnerability nodes, each attack path consists of the subset of nodes necessary to successfully complete an attack. Mitigations "cover" vulnerability nodes, which in turn cover a portion of the associated attack paths. Multiple coverage is accomplished by covering multiple different nodes in an attack path. This coverage structure, representing a layered defense, is motivated by our

discussion with collaborators at SNL. In addition, we assume that coverage is nondecreasing with respect to the number of nodes covered on an attack path. In essence, the system cannot be made more vulnerable by covering additional nodes. We also assume the marginal improvement in coverage decreases with the number of nodes covered on an attack path. Mitigations provide more marginal improvement to overall security when other vulnerabilities on the same attack path are not covered by other mitigations.

We focus on optimal ways to cover vulnerabilities in the attack paths by deploying security mitigations in a layered defense. To this end, we present two types of models: a deterministic model that assumes the mitigation coverage is always effective and a stochastic model that allows for uncertainty in mitigation effectiveness. We also consider variations of these models. These variants consider constraints that limit decision makers to selecting only one mitigation each from a number of mitigation groups.

2.1 Deterministic models

Let S denote the set of attack paths and let N denote the set of vulnerabilities (nodes). Let $N_s \subseteq N$ denote the subset of vulnerabilities in attack path $s \in S$. Let $c_n \in \mathbb{R}_+$ represent the criticality of vulnerability $n \in N$. A higher weight c_n corresponds to a more important vulnerability. We use these critical vulnerability levels to quantify the coverage of each attack path. Let M denote the set of available mitigations, and let $M_n \subseteq M$ denote the set of mitigations that cover vulnerability node $n \in N$. Let $B \in \mathbb{R}_+$ be the total mitigation budget. Each mitigation $m \in M$ has an implementation cost $b_m \leq B$. In many situations, much of this underlying data may be collected from SMEs who have knowledge of the underlying processes, possible mitigations, and mitigation effectiveness [12, 38].

Let variable x_m be 1 if mitigation $m \in M$ is chosen, and 0 otherwise. Let z_n be 1 if node $n \in N$ is covered by a mitigation, and 0 otherwise. Let y_s be the number of vulnerability nodes in attack path $s \in S$ that are covered, weighted by criticality level. Specifically, $y_s = \sum_{n \in N_s} c_n z_n$. We introduce a function $f_s(y_s)$, $s \in S$ that captures the coverage of attack path $s \in S$ with respect to y_s. By assumption, $f_s(\cdot)$ is nondecreasing and concave in y_s for each $s \in S$. Note that $f_s(\cdot)$ might not be identical across all attack paths $s \in S$, since it can reflect the likelihood of an attack occurring and the perceived consequence of the attack.

The budgeted maximum multiple coverage problem (BMMC) is formulated as a mixed-integer programming model:

$$\max_{x,y,z} \quad \sum_{s\in S} f_s(y_s) \tag{1a}$$

$$\text{s.t.} \quad \sum_{m\in M} b_m x_m \leq B \tag{1b}$$

$$y_s = \sum_{n\in N_s} c_n z_n, \qquad s \in S \tag{1c}$$

$$z_n \leq \sum_{m\in M_n} x_m, \qquad n \in N \tag{1d}$$

$$x_m \in \{0,1\}, \qquad m \in M \tag{1e}$$

$$z_n \in \{0,1\}, \qquad n \in N. \tag{1f}$$

The objective (1a) is to maximize the total coverage over all attack paths. (1b) is the budget constraint. Constraint family (1c) defines y_s as the weighted number of nodes covered in attack path $s \in S$. Constraints (1d) ensure that a node is covered when at least one mitigation covering it is selected. The parameters of BMMC can be obtained from SMEs and risk analysts. The coverage function is determined by decision makers depending on the anatomy of each attack. For example, some attack paths may vary in severity or likelihood.

BMMC is NP-hard since it generalizes the budgeted maximum coverage (BMC) problem, a known NP-hard problem [25]. This occurs when each attack path is composed of a single unique node ($|N| = |S|$ and $N_s = \{s\}$ for all $s \in S$), and there is single coverage ($f_s(0) = 0$ and $f_s(1) = 1$ for all $s \in S$).

2.2 Stochastic models

The deterministic model makes the assumption that mitigation coverage is always effective. However, cyber-threats are dynamic and persistent. Therefore, past information on mitigation controls is often incomplete, inaccurate, and not predictive of future mitigation efforts. One of our goals is to identify a set of mitigations that maximizes expected coverage given coverage uncertainty. Therefore, we consider the possibility that mitigations could be ineffective. Let $\boldsymbol{\xi}$ be a random vector on the probability space $(\Omega, \Sigma, \mathbb{P})$ such that $\boldsymbol{\xi}\colon \Omega \to \{0,1\}^{|M|\times|N|}$. $\boldsymbol{\xi}_{mn}$ equals 1 if the cov-

erage of mitigation $m \in M$ is effective on node $n \in N$, and 0 otherwise. The total coverage of the attack paths depends on the selected mitigations x and the random variable $\boldsymbol{\xi}$, and is denoted by $f(x, \boldsymbol{\xi})$. We are interested in improving the overall security of the system with respect to coverage uncertainty. Hence, we formulate a model to maximize the expected coverage $\mathbb{E}_{\boldsymbol{\xi}}[f(x, \boldsymbol{\xi})]$. The expected-value budgeted maximum multiple coverage problem (EBMMC) is formulated as the following two-stage stochastic mixed-integer programming model:

$$\max_{x} \quad \mathbb{E}_{\boldsymbol{\xi}}[f(x, \boldsymbol{\xi})] \tag{2a}$$

$$\text{s.t.} \quad \sum_{m \in M} b_m x_m \leq B \tag{2b}$$

$$x_m \in \{0, 1\}, \qquad m \in M, \tag{2c}$$

where

$$f(x, \xi) = \max_{y, z} \quad \sum_{s \in S} f_s(y_s) \tag{3a}$$

$$\text{s.t.} \quad y_s = \sum_{n \in N_s} c_n z_n, \qquad s \in S \tag{3b}$$

$$z_n \leq \sum_{m \in M_n} \xi_{mn} x_m, \qquad n \in N \tag{3c}$$

$$z_n \in \{0, 1\}, \qquad n \in N. \tag{3d}$$

The objective (2a) of EBMMC is to maximize the expected value of total coverage across all attack paths. The recourse problem (3) evaluates the coverage of a solution x for a given realization ξ of the random variable $\boldsymbol{\xi}$. The realization of the random variable $\boldsymbol{\xi}$ appears in the multiple coverage constraints (3c). Its presence ensures that a node cannot be considered covered if no effective mitigations covers it. EBMMC introduces an incentive to select two mitigations that cover the same node if that node is critical and the mitigations covering that node may be ineffective. EBMMC is NP-hard, because it generalizes BMMC.

If $\{\xi \in \{0, 1\}^{|M| \times |N|} \colon \mathbb{P}(\boldsymbol{\xi} = \xi) > 0\}$ is too large of a set, a deterministic equivalent formulation of (2) is intractable. One way to address this issue is to solve an approximate stochastic problem using sample average approximation [39]. In sample average approximation, we take a finite set of samples $\{\xi^1, \ldots, \xi^K\}$ of $\boldsymbol{\xi}$. We then approximate the expected value function with a sample average

function:

$$\mathbb{E}_{\boldsymbol{\xi}}[f(x,\boldsymbol{\xi})] \approx \frac{1}{K}\sum_{j=1}^{K} f(x,\xi^j).$$

Given this approximation, we can derive a deterministic equivalent formulation (DEF) of EBMMC. First, we define coverage variables for each scenario $j = 1,\ldots,K$. Let z_n^j be 1 if node $n \in N$ is covered by a mitigation in scenario $j = 1,\ldots,K$, and 0 otherwise, Let y_s^j be the weighted number of nodes in attack path $s \in S$ that are covered in scenario $j = 1,\ldots,K$. The DEF of the sample average approximation of EBMMC can be formulated as the following mixed-integer programming model:

$$\max_{x,y,z} \quad \frac{1}{K}\sum_{j=1}^{K}\sum_{s\in S} f_s(y_s^j) \tag{4a}$$

$$\text{s.t.} \quad \sum_{m\in M} b_m x_m \le B \tag{4b}$$

$$y_s^j = \sum_{n\in N_s} c_n z_n^j, \qquad s \in S,\ j = 1,\ldots,K \tag{4c}$$

$$z_n^j \le \sum_{m\in M_n} \xi_{mn}^j x_m, \qquad n \in N,\ j = 1,\ldots,K \tag{4d}$$

$$x_m \in \{0,1\}, \qquad m \in M \tag{4e}$$

$$z_n^j \in \{0,1\}, \qquad n \in N,\ j = 1,\ldots,K. \tag{4f}$$

As the number of sample scenarios K increases, an optimal solution of (4) converges to the optimal solution of EBMMC with probability one [39]. In the computational study, we solve the sample average approximation of EBMMC with large samples to accurately approximate the true optimal solution to EBMMC. Although we conduct a computational study on the sample average approximation problem, all the theoretical results associated with the approximation algorithms in Section 3 apply to EBMMC. Henceforth, any mention of solving a stochastic model refers to solving the corresponding sample average approximation formulation (e.g., (4) for EBMMC).

We are also interested in the k-cardinality constrained maximum multiple coverage problem (kMMC), in which the budget constraint is replaced with a cardinality constraint. That is, $B \coloneqq k$ for some $k \in \mathbb{N}$ and $b_m \coloneqq 1$ for all $m \in M$. Similarly, we consider the k-cardinality constrained expected-value maximum multiple coverage problem (kEMMC). kEMMC is the stochastic extension of kMMC.

The number of variables required to formulate our approximations of EBMMC and kEMMC grows linearly with the sample size K. We solve formulations with thousands of scenarios to ensure our solutions approximate the true optimal solutions well. Formulations of this size can be difficult to solve directly in a reasonable period of time. We thus introduce polynomial-time approximation algorithms with constant worst-case approximation ratios for solving EBMMC and its variants in Section 3. Additionally, we propose a Benders based branch-and-cut algorithm to solve (4) in Section 4.

2.3 Group cardinality models

In real settings, there are often additional requirements for mitigation selection other than a budget constraint. We consider one such requirement here – group cardinality constraints. Group cardinality constraints, or multiple choice constraints, are motivated by the practical concern that some mitigations have conflicting effects and cannot be implemented together. For example, a mitigation for replacing an untrustworthy vendor and a mitigation for improving this vendor's security procedure cannot both be selected.

Let $M_1, M_2, \dots M_\ell$ be a partition of the mitigation set M, where $\cup_{i=1}^{\ell} M_i = M$ and $M_i \cap M_j = \emptyset$ for all distinct $i, j \in \{1, \dots, \ell\}$. Group cardinality constraints stipulate that no more than one mitigation can be selected from partition group M_i $(i = 1, \dots, \ell)$. The constraints are written as follows:

$$\sum_{m \in M_i} x_m \leq 1, \qquad i = 1, \dots, \ell. \tag{5}$$

The k-cardinality constrained expected-value maximum multiple coverage problem with group cardinality constraints (kEMMCG) can be modeled as an integer programming model that maximizes expected coverage (2a) subject to (3b)–(3d), (5), and the k-cardinality constraint (6):

$$\sum_{m \in M} x_m \leq k. \tag{6}$$

We also introduce the deterministic variant of kEMMCG, the k-cardinality maximum multiple coverage problem with group cardinality constraints (kMMCG). kMMCG can be considered as kEMMCG without coverage uncertainty. Both kMMCG and kEMMCG generalize the cardinality-constrained maximum coverage problem with group cardinality constraints [40]. In the next section,

we show that a greedy heuristic achieves a 1/2-approximation ratio for not only kEMMCG, but any nondecreasing submodular maximization problem with a cardinality constraint and group cardinality constraints.

3 Approximation algorithms

In this section, we present polynomial-time approximation algorithms for the models of Section 2. We prove constant worst-case approximation ratios for the algorithms. The proofs presented in Section 3.1 are associated with EBMMC. We also show that they can be easily adapted for BMMC, kEMMC and kMMC. In Section 3.2, we present a greedy heuristic for kMMCG and kEMMCG.

3.1 EBMMC and variants

We first define a submodular function.

Definition 1. *Let $A_1, A_2 \subseteq \mathcal{N}$ and $A_1 \subseteq A_2$. A set function $g(\cdot)$ is submodular if, for any $a \in \mathcal{N} \setminus A_2$, $g(A_1 \cup \{a\}) - g(A_1) \geq g(A_2 \cup \{a\}) - g(A_2)$.*

Let $\chi^A \in \{0,1\}^{|M|}$ be the characteristic vector of a set $A \subseteq M$, where $\chi^A_m = 1$ if $m \in A$, and 0 otherwise. We define the set function $g(\cdot) : 2^M \to \mathbb{R}$ as follows:

$$g(A) = \mathbb{E}_{\boldsymbol{\xi}}[f(\chi^A, \boldsymbol{\xi})]. \tag{7}$$

With function (7), we can reformulate EBMMC as maximizing the set function $g(\cdot)$ subject to a knapsack constraint:

$$\max_{A \subseteq M} \Big\{ g(A) \colon \sum_{m \in A} b_m \leq B \Big\}. \tag{8}$$

A $(1 - 1/e)$-approximation ratio can be achieved in polynomial time for any submodular maximization problem subject to a knapsack constraint [26] or a cardinality constraint [24]. In the following theorem, we show that EBMMC can be reformulated as a submodular maximization problem subject to a knapsack constraint.

Theorem 2. *EBMMC is a nondecreasing submodular maximization problem subject to a knapsack constraint.*

Proof. Consider two sets $A_1, A_2 \subseteq M$ satisfying $A_1 \subseteq A_2$. Let $m \in M \setminus A_2$. It suffices to show $g(A_2) \geq g(A_1)$ and $g(A_1 \cup \{m\}) - g(A_1) \geq g(A_2 \cup \{m\}) - g(A_2)$.

We first show that for fixed uncertainty effectiveness $\xi \in \{0,1\}^{|M|\times|N|}$, it holds that $f(\chi^{A_2}, \xi) \geq f(\chi^{A_1}, \xi)$ and $f(\chi^{A_1\cup\{m\}}, \xi) - f(\chi^{A_1}, \xi) \geq f(\chi^{A_2\cup\{m\}}, \xi) - f(\chi^{A_2}, \xi)$. $f(\chi^{A}, \xi)$ is the total coverage with respect to ξ when a set of mitigations $A \subseteq M$ is selected. Let y_s^A be the weighted number of nodes covered on attack path $s \in S$ when A is selected. We have $f(\chi^A, \xi) = \sum_{s\in S} f_s(y_s^A)$. Since $f_s(\cdot)$ is nondecreasing, we have $f_s(y_s^{A_2}) - f_s(y_s^{A_1}) \geq 0$, implying

$$f(\chi^{A_2}, \xi) \geq f(\chi^{A_1}, \xi). \tag{9}$$

For the same reason, we have $f_s(y_s^{A_1\cup\{m\}}) - f_s(y_s^{A_1}) \geq 0$ and $f_s(y_s^{A_2\cup\{m\}}) - f_s(y_s^{A_2}) \geq 0$. Because $f_s(\cdot)$ is concave, the marginal increase of the objective value is nonincreasing. Therefore, $f_s(y_s^{A_1\cup\{m\}}) - f_s(y_s^{A_1}) \geq f_s(y_s^{A_2\cup\{m\}}) - f_s(y_s^{A_2})$. It follows that

$$f(\chi^{A_1\cup\{m\}}, \xi) - f(\chi^{A_1}, \xi) \geq f(\chi^{A_2\cup\{m\}}, \xi) - f(\chi^{A_2}, \xi). \tag{10}$$

By definition, $g(A) = \mathbb{E}_{\boldsymbol{\xi}}[f(\chi^A, \boldsymbol{\xi})]$. Taking the expected value with respect to $\boldsymbol{\xi}$ on both sides of inequalities (9) and (10) yields $g(A_2) \geq g(A_1)$ and $g(A_1 \cup \{m\}) - g(A_1) \geq g(A_2 \cup \{m\}) - g(A_2)$, as desired. □

Corollary 3. *kEMMC and kMMC are submodular maximization problems subject to a cardinality constraint.*

Theorem 2 allows us to apply well-known results to our model. We show how to adapt the two approximation algorithms in Khuller et al. [25] to EBMMC to achieve $(1 - 1/e)$- and $(1 - 1/\sqrt{e})$-approximation ratios for kEMMC and EBMMC, respectively. Let $b(A) \coloneqq \sum_{m\in A} b_m$ be the total cost of selecting mitigations $A \subseteq M$, and let $\Delta g_m(A) \coloneqq g(A\cup\{m\}) - g(A)$ be the marginal increase of the objective value when $m \in M$ is added to A.

Given a current solution A and a set of available mitigations T, the greedy algorithm Greedy selects a budget-feasible mitigation with the largest ratio $\Delta g_m(A)/b_m$ in each iteration until no more mitigations can be selected. Greedy achieves an optimal $(1 - 1/e)$-approximation ratio for kEMMC and kMMC [24]. This is the best possible approximation ratio unless P = NP [41].

Algorithm: GREEDY(A, T)

1 **while** $T \neq \emptyset$ **do**
2 | Compute $\Delta g_m(A)$ for all $m \in T$
3 | Compute $m^* \in \text{argmax}_{m \in T}\{\Delta g_m(A)/b_m\}$
4 | **if** $b(A \cup \{m^*\}) \leq B$ **then**
5 | | $A \leftarrow A \cup \{m^*\}$
6 | $T \leftarrow T \setminus \{j \in T : b(A \cup \{j\}) > B\}$
7 **return** A

GREEDY(A, T) computes $\Delta g_m(A)$ for all $m \in T$, which is potentially a large number of computations. This computational time can be improved by reducing the number of evaluations considered in line 2 of GREEDY using the "lazy evaluations" procedure proposed by Leskovec et al. [13]. This procedure exploits submodularity by computing $\Delta g_m(A)/b_m$ for the available mitigations in decreasing order based on the previous iteration that led to the selection of mitigation m^*. The procedure terminates when a new value of $\Delta g_m(A)/b_m$ is greater than all such values from the previous iteration. The correctness of this procedure follows from submodularity, since $\Delta g_m(A)/b_m$ can never increase as the set of selected mitigations A grows. The procedure of Leskovec et al. can drastically reduce the number of computations in line 2 of GREEDY. However, it does not change the worst-case time complexity.

Algorithm 1 is the first approximation algorithm for EBMMC and BMMC. It selects the better of a greedy solution and the single best mitigation. Algorithm 1 has an approximation ratio of $(1 - 1/\sqrt{e})$ for EBMMC and BMMC and requires $O(|M|^2)$ objective function evaluations.

Algorithm 1: Construct approximate solution to BMMC/EBMMC

1 $A \leftarrow \emptyset$, $T \leftarrow M$
2 $A \leftarrow$ GREEDY(A, T)
3 Compute $m^* \in \text{argmax}_{m \in T}\{g(\{m\})\}$
4 **if** $g(A) \geq g(\{m^*\})$ **then**
5 | **return** A
6 **else**
7 | **return** $\{m^*\}$

GREEDY and Algorithm 1 can provide insight when the total budget is not available all at once and instead becomes available over time [42]. First, consider the cardinality-constrained problems kEMMC and kMMC. Each time the budget increases by one unit, GREEDY selects the mitigation providing the best marginal increase in objective function value. If we run GREEDY until all

mitigations are selected, we obtain a set of naturally nested solutions, each of which approximates the problem with a constant factor $(1 - 1/e)$. Therefore, the algorithm provides a prioritized list of mitigations to decision makers [43]. For any $k \in \mathbb{N}$, the set of k mitigations selected by GREEDY form a naturally nested set.

The nested solution obtained from GREEDY is not maintained for EBMMC and BMMC. In these cases, mitigation costs are not identical. As such, a different set of mitigations may be selected by GREEDY for different values of the budget. However, Algorithm 1 can be adapted to construct a set of solutions that approximate the efficient frontier between coverage and cost, thus yielding a set of "good" solutions useful for comparing trade-offs between competing objectives. This can be achieved using Algorithm 1 by setting $B \coloneqq \sum_{m \in M} b_m$ and saving the solution, objective function value, and total mitigation cost after each mitigation is added. This set of solutions is naturally nested. Moreover, it can serve as a warm start for finding a set of mitigations at any intermediate budget level. Specifically, if the lists of solution and budget values are $\{A_1, A_2, \ldots, A_{|M|}\}$ and $\{B_1, B_2, \ldots, B_{|M|}\}$, respectively, and we are given intermediate budget value B with $B_i < B < B_{i+1}$, we can run Algorithm 1 by changing line 1 to initialize $A \leftarrow A_i$, $T \leftarrow M \setminus A_i$, and $B \leftarrow B - B_i$.

Algorithm 2 uses GREEDY in a partial enumeration scheme to achieve an approximation ratio of $(1 - 1/e)$ for BMMC and EBMMC [26]. This algorithm requires $O(|M|^5)$ objective function evaluations, making it less efficient than Algorithm 1.

Algorithm 2: Construct approximate solution to BMMC/EBMMC (partial enumeration)

```
1 A* ← ∅, q ← 3
2 for A ⊆ M s.t. |A| < q and b(A) ≤ B do
3 |   if g(A) > g(A*) then
4 |   |   A* ← A
5 for A ⊆ M s.t. |A| = q and b(A) ≤ B do
6 |   W ← Greedy(A, M \ A)
7 |   if g(W) > g(A*) then
8 |   |   A* ← W
9 return A*
```

In Section 5, we compare the solution quality and computational times of Algorithms 1 and 2.

3.2 A greedy heuristic for group cardinality models

We present a modified greedy algorithm that accounts for group cardinality constraints, and we demonstrate that it achieves a 1/2-approximation ratio for kEMMCG and kMMCG.

Using the set function $g(\cdot)$ defined for EBMMC, we reformulate kEMMCG as follows:

$$\max_{A \subseteq M} \{g(A) \colon |A| \leq k,\ |A \cap M_i| \leq 1, i = 1, \ldots, \ell\}. \tag{11}$$

Without loss of generality, we assume $g(\emptyset) = 0$. We also assume $\ell > k$. If not, the k-cardinality constraint is redundant. kEMMCG formulation (11) maximizes a nondecreasing submodular set function subject to a uniform matroid constraint and a partition matroid constraint, the combination of which is equivalent to a truncated partition matroid constraint. A truncated partition matroid is a matroid. Fisher et al. [27] demonstrate that a greedy heuristic achieves a 1/2-approximation ratio for maximizing a monotone submodular set function subject to a matroid constraint. Chekuri and Kumar present a greedy heuristic with a 1/2-approximation ratio for solving the maximum coverage problem subject to a cardinality constraint and group cardinality constraints (a special case of kEMMCG) [40]. Their analysis is based on the multiple knapsack problem. Using a similar analysis, we show that the greedy heuristic of Chekuri and Kumar can be adapted to the general monotone submodular maximization problem subject to a cardinality constraint and group cardinality constraints, of which kEMMCG is an instance. The adapted algorithm achieves the same approximation ratio of 1/2.

Recall $\Delta g_m(A) = g(A \cup \{m\}) - g(A)$ is the marginal increase in objective value when m is added to $A \subseteq M$. Algorithm 3 iteratively selects the mitigation that provides the most marginal improvement in objective value. However, it considers mitigations only from those groups from which a mitigation has not already been selected. For all $m \in M$, let $I(m) \in \{1, \ldots, \ell\}$ satisfy $m \in M_{I(m)}$. That is, $I(m)$ indexes the unique group to which m belongs

Algorithm 3 requires $O(k|M|)$ objective function evaluations. We present the following theorem to demonstrate that Algorithm 3 obtains a 1/2-approximation ratio for kEMMCG and kMMCG. The result is proven for the general monotone submodular maximization problem subject to a cardinality constraint and group cardinality constraints. Note that this result matches the approximation ratio in Fisher [27] for monotone submodular maximization subject to a matroid constraint.

Algorithm 3: Construct approximate solution to kMMCG/kEMMCG

$A \leftarrow \emptyset$, $T \leftarrow M$
while $|A| < k$ **do**
  Compute $\Delta g_m(A)$ for all $m \in T$
  Compute $m^* \in \operatorname{argmax}_{m \in T} \Delta g_m(A)$
  $A \leftarrow A \cup \{m^*\}$
  $T \leftarrow T \setminus M_{I(m^*)}$
return A

Theorem 4. *Algorithm 3 achieves an approximation ratio of* $1/2$ *for maximizing a monotone submodular function subject to a cardinality constraint and group cardinality constraints.*

Proof. We consider the submodular maximization problem (11) for a general monotone submodular function g.

Observe Algorithm 3 runs for $k < \ell$ iterations and returns a set of mitigations $A \subseteq M$. Without loss of generality, assume the partition $M_1, \ldots, M_\ell$ is ordered such that the mitigation m_j selected by Algorithm 3 in iteration j is an element of M_j, $j = 1, \ldots, k$. In this ordering, Algorithm 3 selects no mitigation from groups $M_{k+1}, \ldots, M_\ell$. For $j = 1, \ldots, k$, let $A_j \coloneqq \cup_{i=1}^{j} \{m_i\}$. Observe $A_k = A$, and let $A_0 \coloneqq \emptyset$. Let $A^* = \{m_1^*, \ldots, m_k^*\} \subseteq M$ be the optimal solution, indexed such that $I(m_j^*) \geq j$ for $j = 1, \ldots k$. That is, mitigation m_j^* was a candidate for selection in iteration j of Algorithm 3. Such an indexing trivially exists.

In each iteration $j = 1, \ldots, k$ of Algorithm 3, we select the mitigation that has the largest marginal increase in objective value with respect to the previously selected mitigations A_{j-1}. It follows that

$$g(A_j) - g(A_{j-1}) \geq g(A_{j-1} \cup \{m_j^*\}) - g(A_{j-1}) \qquad j = 1, \ldots, k.$$

Because $A_j \subseteq A$ for all $j = 1, \ldots, k$ and $g(\cdot)$ is a nondecreasing submodular set function, we have

$$g(A_j) - g(A_{j-1}) \geq g(A \cup \{m_j^*\}) - g(A) \qquad j = 1, \ldots, k. \tag{12}$$

Summing both sides of (12) over all groups yields

$$g(A) = \sum_{j=1}^{k} [g(A_j) - g(A_{j-1})] \geq \sum_{j=1}^{k} [g(A \cup \{m_j^*\}) - g(A)]$$

$$\geq g(A \cup A^*) - g(A) \tag{13}$$

$$\geq g(A^*) - g(A). \tag{14}$$

Inequality (13) follows from the fact that $g(\cdot)$ is a submodular set function, so the marginal increase in g when adding a set A^* to A is smaller than the sum of the marginal increase in g when adding each element in A^* to A individually. Inequality (14) holds because $A^* \subseteq A^* \cup A$ and $g(\cdot)$ is nondecreasing.

Hence, $g(A) \geq g(A^*)/2$. □

4 Benders based branch-and-cut algorithm

In this section, we propose a Benders based branch-and-cut algorithm for solving EBMMC. Because of the presence of nonlinear functions in the recourse problem (3), we use the generalized Benders decomposition framework [44]. A Benders decomposition algorithm requires continuous second-stage variables, which is not satisfied by the variables z_n ($n \in N$) defined in (3d). However, the problem in which each z_n variable is relaxed to be a continuous variable on the interval $[0, 1]$ is equivalent. Indeed, for any feasible first-stage solution x and $\xi \in \{0,1\}^{|M|\times|N|}$, the right-hand side of (3c) is a nonnegative integer. If this expression equals 0, z_n is forced to 0. If this expression is greater than or equal to 1, the objective value (3a) can only be improved by setting z_n equal to its upper bound of 1, because $f_s(\cdot)$ is nondecreasing for all $s \in S$ and the weights c_n ($n \in N$) are nonnegative. Thus, we are able to solve EBMMC via a Benders algorithm by considering $z_n \in [0, 1]$.

The Benders decomposition algorithm is based on the following Benders master problem:

$$\max_{x,\theta} \quad \frac{1}{K}\sum_{j=1}^{K}\theta^j \tag{15a}$$

$$\text{s.t.} \quad \sum_{m\in M} b_m x_m \leq B \tag{15b}$$

$$\theta^j \leq f(\bar{x}, \xi^j) + \sum_{n\in N}\bar{\lambda}_n \sum_{m\in M_n}\xi^j_{mn}(x_m - \bar{x}_m), \qquad (\bar{x}, \bar{\lambda}) \in T^j,\ j = 1, \ldots, K \tag{15c}$$

$$x_m \in \{0, 1\}, \qquad m \in M, \tag{15d}$$

where $f(x, \xi^j)$ is the optimal value of the Benders subproblem for scenario $j = 1, \ldots, K$, as defined

in (3). The set T^j consists of pairs $(\bar{x}, \bar{\lambda})$, where $\bar{x}$ was previously obtained as an optimal feasible solution to the master problem and $\bar{\lambda}$ encodes the optimal dual multipliers corresponding to constraint (3c) of subproblem (3) solved to evaluate $f(\bar{x}, \xi^j)$.

Observe (3) has a feasible solution for any first-stage solution x (i.e., it has relatively complete recourse), because $y = 0$, $z = 0$ satisfies (3b)–(3d) for any $x \in \{0,1\}^{|M|}$. Since the integrality restriction (3d) are relaxed, the subproblem (3) is a convex program defined by linear constraints. Thus, Slater's condition is satisfied, and strong duality holds. It follows that a generalized Benders algorithm on EBMMC converges to the optimal solution.

Because the first-stage variables x are binary, Benders decomposition must be implemented within a branch-and-cut algorithm to solve EBMMC. At each node of the master problem branch-and-bound tree where an integral first-stage solution $\bar{x}$ is obtained as an optimal solution to the continuous relaxation, we solve the EBMMC subproblem $f(\bar{x}, \xi^j)$ for each scenario $j = 1, \ldots, K$ and obtain the values of the optimal dual multipliers $\bar{\lambda}^j_n$, $n \in N$ corresponding to constraints (3c) of $f(\bar{x}, \xi^j)$. Violated Benders cuts of the form (15c) are added to the continuous relaxation of the node and it is re-solved. If no violated Benders inequalities exist for an integer-feasible solution at a node of the branch-and-bound tree, the upper bound for the problem is updated, and the node is pruned. In our experiments presented in Section 5, we implement this Benders algorithm using Gurobi's lazy constraint callback.

Following the approach of Bodur et al. [45], we solve the problem in two phases. First, the integrality restrictions on first-stage variables are removed, and the relaxed master problem is solved. The subproblems (3) are solved at the relaxed master problem solution for each scenario. Based on the dual solutions, any violated Benders cuts are added to the relaxed master problem. The relaxed master problem is solved again and the process repeats. Once no violated Benders cuts are found for a solution to the relaxed master problem, the integrality restriction on x is re-added. The model, including the Benders inequalities found in this initial phase, is given to a MIP solver to commence the branch-and-cut algorithm (where Benders cuts are added when integer solutions are found, as described in the previous paragraph). The advantage of this approach is that the MIP solver leverages the initial set of Benders cuts to derive and add its own general-purpose cuts to the formulation. This yields a stronger continuous relaxation, potentially resulting in more node pruning and a smaller tree. In our implementation, we initialize this process with $x = 0$ to prevent unboundedness in the first iteration's master problem.

5 Computational results

This section is divided into three parts, each of which focuses on a particular budgeted coverage model. Throughout the computational study, we analyze the solutions and performance of approximation algorithms and exact methods on the proposed models. All implementation was done with Python 2.7.12. The approximation algorithms were run with PyPy 5.10.0, an alternative Python compiler. Computational tests were run on a machine with two 2.00GHz Intel Xeon E7-4850 processors and 256GB RAM. A one-hour time limit was imposed. Models were solved with Gurobi 7.5.1. Exact algorithms were solved to Gurobi's default relative optimality gap of 10^{-4}. All algorithms were limited to a single thread.

As mentioned in Section 2, real data for building instances of our models are scarce or inaccessible. For this reason, we construct synthetic data of varying sizes in conjunction with our collaborators at SNL. The data were generated as follows. We randomly sample from the mitigation set M with replacement to generate the sets M_n ($n \in N$), enforcing $|M_n| \leq 3$. Similarly, we set the upper bound on the number of nodes in each attack path to five and randomly sample from N with replacement to generate a list of nodes in each attack path N_s, $s \in S$. In practice, there may be more than five nodes on some attack paths. However, feedback from our collaborators indicate that this is a reasonable upper bound, since there are relatively few access points for control in the supply chain attacks under consideration. We set $c_n = 1$ for all $n \in N$, representing a situation where all nodes are equally vulnerable. We set $f_s(y_s) = -a_s(y_s)^2 + 2a_s|N_s|y_s$, which is zero when no nodes in $s \in S$ are covered and achieves its maximum value of $a_s|N_s|^2$ when all $|N_s|$ nodes in attack path s are covered. The attack path weights a_s for $s \in S$ are generated from a $U(0, 10)$ distribution. The mitigation costs b_m, $m \in M$ are generated from a $U(0, 1)$ distribution. We create eight datasets consisting of the set covering data and the multiple coverage data. Each of the eight generated datasets has a unique size $|M| \times |N| \times |S|$, ranging from $20 \times 20 \times 10$ to $1000 \times 1000 \times 500$. As it becomes relevant throughout this section, we describe how the additional data was generated.

In general, the coverage models we consider are mixed-integer convex programs, and instances of the models can be directly solved with mixed-integer convex optimization solvers. Depending on the structure of f_s, $s \in S$, specialized solvers can be used (e.g., each f_s is quadratic or second-order cone representable). However, if $c_n \in \mathbb{Z}_+$ for all $n \in N$, as is the case for the test instances we consider, then by constraint (1c), the decision variables y_s are integer-valued for any feasible

solution to a particular coverage problem. Thus, we can replace each f_s in the nonlinear objective function (1a) with a concave piecewise linear function, where the piecewise function equals $f_s(y_s)$ at each feasible value of y_s. To this effect, we add the following set of constraints to the model:

$$r_s \leq f_s(t) + [f_s(t+1) - f_s(t)](y_s - t), \qquad t = 0, 1, \ldots, \sum_{n \in N_s} c_n - 1, \ s \in S, \tag{16}$$

where r_s is a new real-valued variable that captures the coverage of path $s \in S$. We then replace the original objective (1a) with

$$\max_{r,x,y,z} \quad \sum_{s \in S} r_s. \tag{17}$$

The piecewise linearization induced by constraints (16) and the objective (17) does not change the objective function values at feasible y_s ($s \in S$). Therefore, this linearization results in an equivalent formulation that applies to all models in this paper. If the values of c are large, then it may not be practical to directly add all constraints of the form (16) to the formulation. Instead, these inequalities can be implemented as lazy cuts in a branch-and-cut algorithm, where they are added as needed to the model at integer feasible nodes of the branch-and-bound tree. For the test instances we consider, we found that the linearized deterministic models solved much faster than the mixed-integer quadratic programming formulations. Note that if c is not integral, this linearization does not apply. Instead, inequalities built from the subderivatives of f_s at y_s can be added as lazy cuts in a method similar to the branch-and-cut procedure described for the integral case.

All models were solved with the proposed greedy algorithms. For comparison, deterministic models (BMMC, kMMC, and kMMCG) were solved using Gurobi's default branch-and-cut algorithm. This algorithm is denoted as BNC. Stochastic models (EBMMC and kEMMCG) were solved using two methods:

- DEF: Use Gurobi's default branch-and-cut algorithm to solve the DEF (e.g., (4) for EBMMC).
- BEN: Use the Benders branch-and-cut algorithm described in Section 4, implemented with Gurobi.

5.1 BMMC solutions

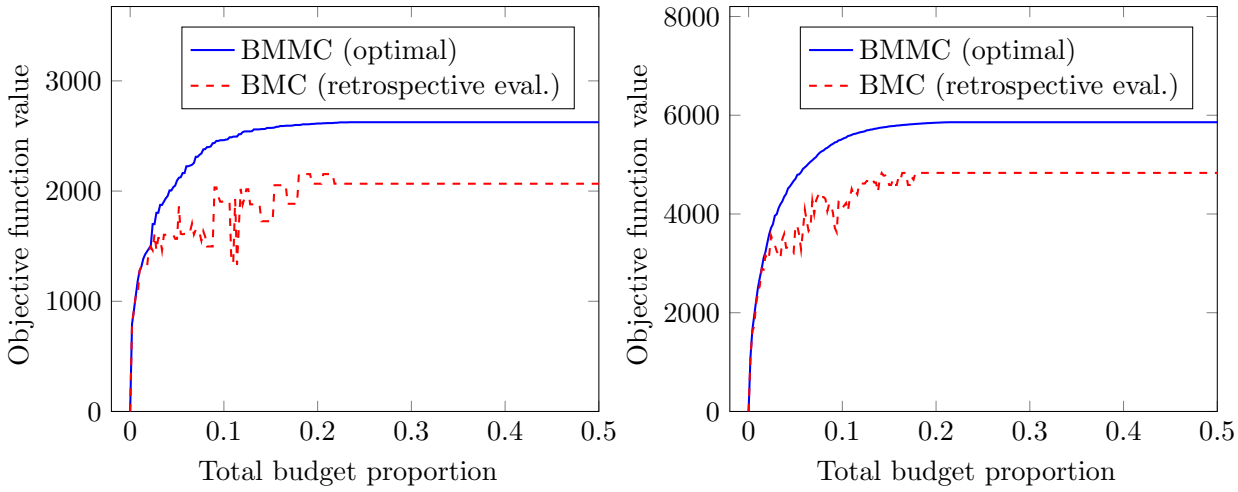
We start by solving BMMC to examine the effect of multiple coverage on mitigation selection. First, we investigate the potential benefit of (1), BMMC's multiple coverage formulation. Attack paths in our problem represent vulnerabilities. In a problem with single coverage (e.g., BMC), the objective function gives no incentive to cover a path more than once. In the multiple coverage problems proposed in this paper (e.g., BMMC), each attack path can be covered multiple times to encourage a layered defense. Attacks can be prevented in several ways by selecting different mitigations. This results in a more robust approach to cybersecurity planning that reflects the dynamic nature of security threats [46] more thoroughly than single coverage models.

We compare BMMC solutions with BMC solutions on two of the generated instances to illustrate the benefit of considering multiple coverage. The formulations of BMC and BMMC are identical except for the objective function. In our notation, the BMC coverage function is $f_s(y_s) = \min\{1, y_s\}$. While we cannot compare these two solutions directly given that the models optimize two different objective functions, we retrospectively evaluate the BMMC multiple coverage objective function with the optimal BMC solution.

Figures 1a and 1b show the multiple coverage objective function values associated with the BMMC and BMC solutions for a variety of budget values. Two problem instances from the generated dataset are shown. BMMC and BMC solutions perform similarly for low budget levels. As the budget increases, their performance with respect to the BMMC objective function diverges. Intuitively, once the budget becomes large enough that the BMMC objective function warrants covering some attack paths more than once, BMMC and BMC select different sets of mitigations. The gap between these two solution approaches represents the value added by considering a multiple coverage model.

Next, we test the performance of Algorithms 1 and 2 for solving BMMC on the eight main datasets. We set $B \coloneqq 0.05 \sum_{m \in M} b_m$. We compare the performance of these approximation algorithms to BNC. We also test the algorithms on kMMC using the same dataset by setting $b_m \coloneqq 1$ $(m \in M)$ and $B \coloneqq 0.1|M|$.

Table 1 reports the approximation ratios and computation times of Algorithms 1 and 2 on BMMC and kMMC instances. The approximation ratio is calculated as the approximate solution's objective value divided by the optimal objective value. We were able to obtain optimal solutions to some



(a) $|M| = 50$, $|N| = 50$, $|S| = 20$

(b) $|M| = 100$, $|N| = 100$, $|S| = 50$

Figure 1: BMMC solution vs. retrospective evaluation of optimal BMC solution for varying budget levels.

instances by allowing an exact solution algorithm to run longer. In this way, we obtained the approximation ratios necessary to evaluate the quality of the heuristic solutions. Throughout, for instances for which the optimal solution was never found but an approximate solution was obtained, we use an asterisk (*) to signify an unknown approximation ratio. We observe that Algorithm 1 and BNC performed comparably on all instances except the largest. BNC was unable to solve either the BMMC or kMMC version of this instance within the one hour time limit, though Algorithm 1 obtained optimal solutions quickly. Algorithm 1 solved all instances in under 20 seconds and returned heuristic solutions within 2.6% of optimal. Algorithm 2's partial enumeration scheme quickly became intractable as the number of possible mitigations grew, resulting in a poor performance overall. As such, we do not consider Algorithm 2 in the tests that follow. However, for the few instances it was able to solve, Algorithm 2 obtained the optimal solution.

(a) BMMC

			Approx. ratio		Solve time (s)		
$\|M\|$	$\|N\|$	$\|S\|$	Alg. 1	Alg. 2	Alg. 1	Alg. 2	BNC
20	20	10	100.00%	100.00%	<0.1	<0.1	<0.1
50	50	20	97.43%	100.00%	<0.1	0.6	<0.1
100	100	50	98.30%	100.00%	<0.1	345.4	0.2
200	200	100	99.06%	NA	0.1	>3600	0.4
500	500	200	99.71%	NA	0.7	>3600	0.3
600	600	300	99.63%	NA	1.6	>3600	1.6
800	800	400	99.35%	NA	3.3	>3600	0.8
1000	1000	500	99.89%	NA	18.2	>3600	>3600

(b) kMMC

			Approx. ratio		Solve time (s)		
$\|M\|$	$\|N\|$	$\|S\|$	Alg. 1	Alg. 2	Alg. 1	Alg. 2	BNC
20	20	10	100.00%	100.00%	<0.1	<0.1	0.2
50	50	20	99.56%	100.00%	0.1	4.3	<0.1
100	100	50	99.80%	100.00%	<0.1	388.8	0.1
200	200	100	99.86%	NA	<0.1	>3600	0.8
500	500	200	99.23%	NA	0.4	>3600	4.8
600	600	300	99.42%	NA	0.8	>3600	8.8
800	800	400	98.81%	NA	1.9	>3600	64.6
1000	1000	500	*	NA	9.0	>3600	>3600

Table 1: Performance of Algorithm 1, Algorithm 2, and BNC on BMMC and kMMC instances

To further examine the performance of Algorithm 1, we test it on larger problem instances. For the set covering data, we use the test instances posted in the online OR-Library [47]. We select five instances, scpnrg1-scpnrg5, from one of the hardest problem sets [48]. All instances have $|M| = 10000$, $|N| = 1000$, and $|S| = 500$. We generate the remaining data in the same manner described earlier. We use mitigation budget $B := 0.0001 \sum_{m \in M} b_m$. Table 2 compares the performance of Algorithm 1 with the standard Gurobi branch-and-cut algorithm. Algorithm 1 solved all of these larger instances in under seven seconds. The objective values of these heuristic solutions are all within 1% of the optimal objective value. Gurobi was unable to optimally solve any of the scpnrg instances within the one hour time limit. We instead report the relative optimality gap obtained after the one-hour time limit was reached. BNC closed the gap to within 1% on all instances after one hour.

	Alg. 1		BNC
Instance	Approx. ratio	Solve time (s)	MIP gap
scpnrg1	99.66%	6.5	0.26%
scpnrg2	99.85%	6.5	0.28%
scpnrg3	99.63%	5.8	0.81%
scpnrg4	99.99%	5.7	0.37%
scpnrg5	99.87%	5.8	0.74%

Table 2: Algorithm 1 and BNC performance on BMMC OR-Library instances

As mentioned in Section 3, Algorithm 1 can be used to identify a set of non-dominated solutions to BMMC that can help decision makers compare the trade-offs of competing objectives. To do so, we set $B \coloneqq \sum_{m \in M} b_m$ and run Algorithm 1 to obtain a list of mitigations $\{m_1, m_2, \ldots, m_L\}$ that are selected successively. We use Gurobi to optimally solve the model at the budget level $\sum_{i=1}^{j} b_{m_i}$ for $j = 1, \ldots, L$. We then compare these optimal solutions to the solutions $\{m_1, \ldots, m_j\}$ for $j = 1, \ldots, L$. Figure 2 demonstrates the quality of the nested Algorithm 1 solutions on an instance from the main dataset. The solutions were obtained by running Algorithm 1 a single time. At all budget levels, the difference between the optimal solution and Algorithm 1's solution is extremely small. The Algorithm 1 solutions are efficient to generate and give decision makers an overview of trade-offs between cost and vulnerability reduction. However, it is important to note that the nested property of an Algorithm 1 solution is only maintained when the budget increases to exactly $\sum_{i=1}^{j} b_{m_i}$ for $j \in \{1, \ldots, L\}$.

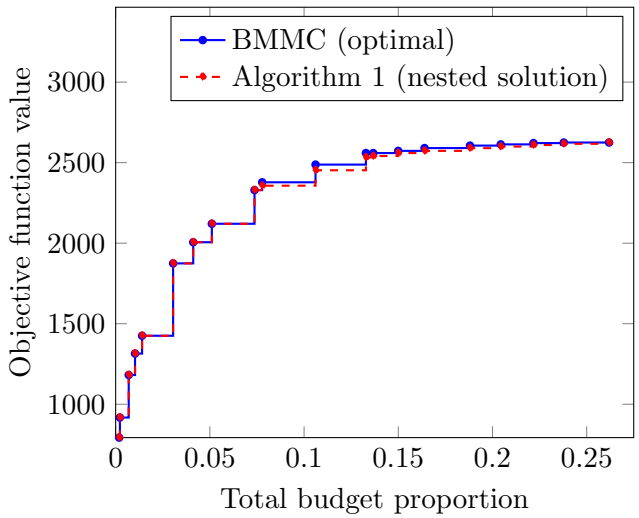


Figure 2: Naturally nested Algorithm 1 solutions compared to optimal solutions for BMMC at corresponding budget levels. $|M| = 50$, $|N| = 50$, $|S| = 20$.

5.2 EBMMC solutions

We examine the impact of uncertainty on mitigation selection by comparing the difference in node coverage between BMMC and EBMMC solutions. We consider (4), the sample average approximation EBMMC model. Throughout, we assume that $\boldsymbol{\xi}$ consists of independent and identically distributed (i.i.d.) Bernoulli random variables with success probability $\bar{p} = 0.5$. That is, $\mathbb{P}(\boldsymbol{\xi}_{mn} = 1) = 0.5$ for all $m \in M$, $n \in N$. We draw K samples $\{\xi^1, \ldots, \xi^K\}$ from $\boldsymbol{\xi}$'s distribution.

To demonstrate the differences in solutions when using the stochastic model, EBMMC, instead of the deterministic model, BMMC, we create a new set of three test instances with $|M| = |N| = |S| = 20$. In these instances, each attack path consists of a single node. Each mitigation covers no more than four nodes. Mitigation costs are selected from $\{6, 25\}$. We designate half of the attack paths $s \in S$ as "important" and assign them a weight of $a_s = 10$. The remaining attack paths are weighted by $a_s = 1$. We use $K = 1000$ scenarios for EBMMC and consider a variety of budget levels.

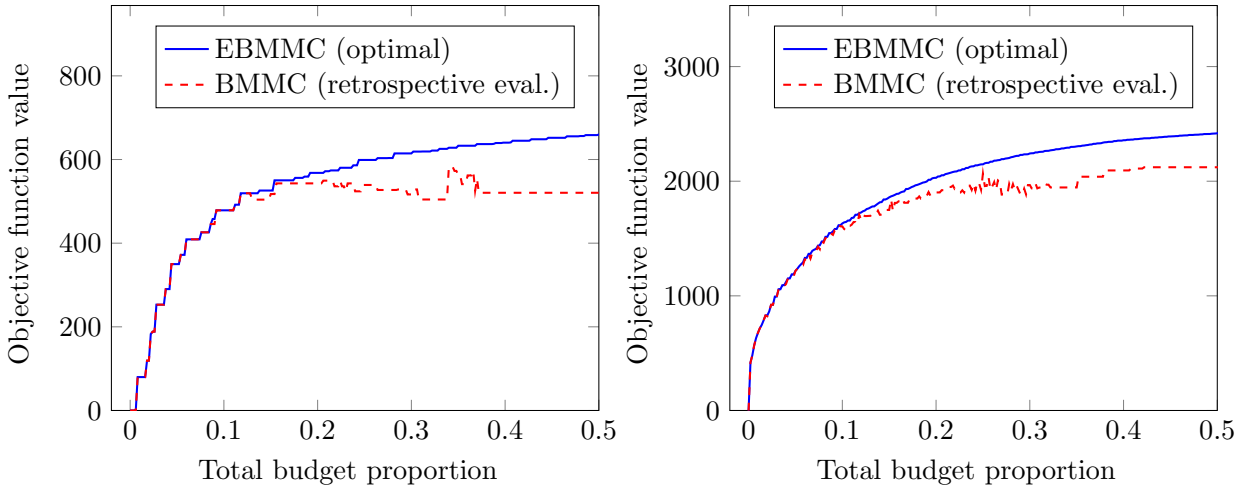
Table 3 reports the average number of times an attack path is covered across all attack paths and important attack paths. Since each attack path is composed of a single node, these results indicate the average number of times a node is covered. While the EBMMC solutions did not provide much benefit over BMMC in covering all nodes, they covered important nodes far more often on average. The stochastic model solutions hedge against the risk of coverage failure by selecting mitigations that cover important attack paths multiple times, as these attack paths contribute significantly more to the objective function. In BMMC, covering an important node a second time does not increase the objective function value.

Instance	$B/\sum_{m\in M} b_m$	Average node coverage (all)		Average node coverage (important)	
		BMMC	EBMMC	BMMC	EBMMC
1	0.2	0.55	1.00	0.85	1.70
	0.5	1.10	1.00	1.60	3.10
	0.8	1.10	1.00	2.80	3.20
2	0.2	0.55	1.00	0.85	1.70
	0.5	1.10	1.10	1.50	2.90
	0.8	1.10	1.10	2.50	3.00
3	0.2	0.50	0.90	0.80	1.60
	0.5	1.25	1.00	1.35	2.60
	0.8	1.05	1.10	2.30	2.70

Table 3: Coverage differences between BMMC and EBMMC solutions

We further demonstrate the value of the stochastic model by examining the performance of EBMMC and BMMC solutions when uncertain mitigation effectiveness is considered. Figure 3 shows the objective function values of the BMMC and EBMMC solutions for different budget values on two instances from the main dataset. We use $K = 1000$ scenarios. Since the objective function of BMMC does not include the lost coverage due to random mitigation failure, we can not directly compare the objective values of EBMMC and BMMC. Instead, we retrospectively evaluate the expected total objective function values associated with BMMC solutions using the same sample $\{\xi^1, \ldots, \xi^K\}$ used for EBMMC. Observe the optimal objective value of EBMMC is a nondecreasing function of the budget. The BMMC retrospective expected-value objective function values do not increase monotonically. This is because the BMMC solutions overlook the possibility of coverage failure and do not perform steadily under the scenarios. The gap between the optimal EBMMC objective value and the EBMMC objective function retrospectively evaluated with an optimal BMMC solution is the value added by solving the stochastic model.

Next, we test the proposed methods on four EBMMC instances from the datasets. We set $B \coloneqq 0.05 \sum_{m\in M} b_m$. We consider five different values of K for each instance. Table 4 reports the approximation ratio of Algorithm 1, the solve times of Algorithm 1, BEN, and DEF, and the relative MIP optimality gaps obtained after the one-hour time limit for BEN and DEF. For DEF, in many cases, the solver was unable to finish solving the root node within the time limit and thus did not even obtain a feasible solution. In these cases, we list the MIP gap as "na." In contrast, Algorithm 1 obtains heuristic solutions to all of the instances within 1200 seconds, and all



(a) $|M| = 20$, $|N| = 20$, $|S| = 10$

(b) $|M| = 50$, $|N| = 50$, $|S| = 20$

Figure 3: Optimal EBMMC solution vs. retrospective evaluation of optimal BMMC solution for varying budget levels.

of these solutions are within 1% of optimal. On the other hand, although significantly slower than Algorithm 1, the Benders branch-and-cut algorithm is able to solve nearly all the test instances within an hour, indicating that it may be a suitable solution method for decision-makers wishing to find a provably optimal solution.

				Approx. ratio	Solve time (s)			MIP gap	
$\|M\|$	$\|N\|$	$\|S\|$	K	Alg. 1	Alg. 1	BEN	DEF	BEN	DEF
20	20	10	1000	100.00%	0.2	18.9	46.3	0.00%	0.00%
20	20	10	2000	100.00%	0.1	40.1	106.0	0.00%	0.00%
20	20	10	3000	100.00%	0.2	64.7	226.8	0.00%	0.00%
20	20	10	4000	100.00%	0.3	86.9	516.9	0.00%	0.00%
20	20	10	5000	100.00%	0.5	110.4	638.7	0.00%	0.00%
50	50	20	1000	99.95%	1.4	128.1	851.2	0.00%	0.00%
50	50	20	2000	100.00%	3.9	312.0	2899.9	0.00%	0.00%
50	50	20	3000	99.89%	8.3	447.4	>3600	0.00%	na
50	50	20	4000	99.85%	8.7	640.4	>3600	0.00%	na
50	50	20	5000	99.83%	13.2	677.7	>3600	0.00%	na
100	100	50	1000	100.00%	17.1	181.4	1467.0	0.00%	0.00%
100	100	50	2000	100.00%	41.5	410.2	>3600	0.00%	na
100	100	50	3000	100.00%	79.1	686.3	>3600	0.00%	na
100	100	50	4000	100.00%	104.8	1119.0	>3600	0.00%	na
100	100	50	5000	100.00%	136.1	1308.9	>3600	0.00%	na
200	200	100	1000	99.69%	202.0	643.3	>3600	0.00%	na
200	200	100	2000	99.50%	477.9	1271.1	>3600	0.00%	na
200	200	100	3000	99.43%	738.5	2131.3	>3600	0.00%	na
200	200	100	4000	99.40%	941.2	3127.7	>3600	0.00%	na
200	200	100	5000	99.39%	1166.7	>3600	>3600	0.11%	na

Table 4: Comparison of Algorithm 1, directly solving DEF, and Benders branch-and-cut algorithmand on EBMMC instances

5.3 Group cardinality model solutions

Finally, we test the performance of Algorithm 3 and the exact methods on kEMMCG and kMMCG. By Theorem 4, Algorithm 3 achieves a 1/2-approximation ratio for these problems.

We first compare the solutions obtained from Algorithm 3 to BEN and DEF on kEMMCG instances from the main dataset. We use varying values of ℓ and k. Given the number of groups ℓ, we randomly sample from the set of mitigations M without replacement to generate a partition of mitigations $M = \bigcup_{i=1}^{\ell} M_i$. We make each partition group as close in size as possible. We use $k = \ell/2$ and $k \approx \ell/4$, because the cardinality constraint is redundant when $k \geq \ell$. For all instances, we consider $K = 1000$ scenarios. The deterministic variant, kMMCG, was easily solved by Algorithm 3 and BNC across all of our test instances. Hence, we do not report results for kMMCG on the main dataset.

Table 5 presents the approximation ratio of Algorithm 3, the solve times of Algorithm 3, BEN,

and DEF, and the relative MIP optimality gaps achieved after the one-hour time limit for BEN and DEF. For all instances for which we were able to obtain an optimal solution, the solutions obtained by Algorithm 3 were within 0.6% of optimal, and provided solutions for all instances in less than 1300 seconds. In terms of exact methods, the MIP solver was only able to solve the DEF for three of the 18 test instances. BEN was able to solve 12 of the instances, and the maximum ending MIP gap was 4.12%. Thus, BEN is again clearly a much better option for obtaining an exact optimal solution than DEF, but it is also clear that Algorithm 3 may be necessary for obtaining good solutions to the hardest instances.

					Approx. ratio	Solve time (s)			MIP gap	
$\|M\|$	$\|N\|$	$\|S\|$	ℓ	k	Alg. 3	Alg. 3	BEN	DEF	BEN	DEF
100	100	50	10	3	100.00%	1.3	70.3	1469.1	0.00%	0.00%
100	100	50	10	5	100.00%	2.2	123.9	2833.5	0.00%	0.00%
100	100	50	50	13	100.00%	14.5	944.6	>3600	0.00%	na
100	100	50	50	25	99.43%	40.8	>3600	>3600	1.20%	na
100	100	50	100	25	100.00%	47.9	785.0	>3600	0.00%	na
100	100	50	100	50	100.00%	136.5	637.9	>3600	0.00%	na
200	200	100	10	3	100.00%	4.8	78.9	2593.5	0.00%	0.00%
200	200	100	10	5	100.00%	8.0	98.2	>3600	0.00%	na
200	200	100	50	13	100.00%	46.0	276.1	>3600	0.00%	na
200	200	100	50	25	*	114.0	>3600	>3600	1.10%	na
200	200	100	100	25	*	137.2	>3600	>3600	1.67%	na
200	200	100	100	50	*	357.0	>3600	>3600	2.68%	na
500	500	200	10	3	100.00%	25.9	135.7	>3600	0.00%	na
500	500	200	10	5	100.00%	42.8	228.2	>3600	0.00%	na
500	500	200	50	13	100.00%	160.8	539.4	>3600	0.00%	na
500	500	200	50	25	99.59%	352.3	2045.0	>3600	0.00%	na
500	500	200	100	25	99.62%	485.6	>3600	>3600	0.07%	na
500	500	200	100	50	*	1232.7	>3600	>3600	4.12%	na

Table 5: Comparison of Algorithm 3, BEN, and DEF on kEMMCG instances with varying ℓ and k.

Table 6 reports the performance of Algorithm 3 and BNC on the large OR-Library instances. We generate partitions of M in the aforementioned manner. Algorithm 3 was again able to obtain near-optimal solutions to all instances in under a minute, while BNC could only solve instances with the smallest values of ℓ. On instances for which we obtained an optimal solution with BNC, Algorithm 3's solutions were within 2.4% of optimal.

			Approx. ratio	Solve time (s)	
Instance	ℓ	k	Alg. 3	Alg. 3	BNC
scpnrg1	10	3	99.76%	0.4	27.9
	10	5	97.60%	0.6	42.6
	50	13	99.58%	2.8	>3600
	50	25	*	6.7	>3600
	100	25	*	8.1	>3600
	100	50	*	26.5	>3600
scpnrg2	10	3	99.83%	0.5	22.6
	10	5	98.41%	0.8	42.3
	50	13	99.35%	3.5	>3600
	50	25	*	8.5	>3600
	100	25	*	10.4	>3600
	100	50	*	26.6	>3600
scpnrg3	10	3	100.00%	0.4	21.0
	10	5	99.95%	0.8	40.3
	50	13	98.75%	3.5	>3600
	50	25	*	7.7	>3600
	100	25	*	10.4	>3600
	100	50	*	25.9	>3600
scpnrg4	10	3	100.00%	0.4	23.5
	10	5	99.02%	0.8	57.2
	50	13	99.08%	3.5	>3600
	50	25	*	8.3	>3600
	100	25	*	9.8	>3600
	100	50	*	25.5	>3600
scpnrg5	10	3	98.63%	0.5	20.0
	10	5	98.76%	0.8	60.2
	50	13	99.19%	3.6	>3600
	50	25	*	8.5	>3600
	100	25	*	10.0	>3600
	100	50	*	26.6	>3600

Table 6: Comparison of Algorithm 3 and MIP solver on kMMCG OR-Library instances with varying ℓ and k.

6 Conclusion

In this paper, we address the problem of prioritizing and deploying security mitigations in a layered defense. We do so by proposing budgeted maximum multiple coverage problems to characterize investment in cybersecurity mitigations with the objective of maximizing multiple coverage, where

the coverage is represented by a submodular set function. We introduce model variations that consider the possibility of coverage failure, the solutions of which lead to a more robust investment plan. Motivated by the decision maker's different requirements for selecting security mitigations, we consider model variants based on cardinality constraints and group cardinality constraints.

We demonstrate that our problems can be formulated as submodular maximization problems subject to linear or matroid constraints. We develop approximation algorithms with a polynomial number of objective function evaluations for each of the problem variants and prove guaranteed performance ratios for these algorithms. An optimal $(1-1/e)$-approximation ratio is demonstrated for solving EBMMC, BMMC, kMMC, and kEMMC. An algorithm yielding a $1/2$-approximation ratio is identified for general monotone submodular maximization subject to cardinality constraints and group cardinality constraints, of which kMMCG and kEMMCG are instances. The greedy algorithms can identify a nested set of non-dominated solutions at no additional computational expense.

In the computational study, we observe that greedy algorithms identify near-optimal solutions to large problem instances. In general, these approximation algorithms solved instances faster than directly solving the corresponding MIP formulation (for deterministic formulations) and a Benders branch-and-cut implementation (for stochastic formulations). The objective values of these solutions are within 2.6% of the optimal objective values across all the instances for which we were able to obtain the true optimal solution. This suggests that simple greedy algorithms are a practical option for solving large-scale maximum coverage variants. We also found that for the expected-value variations of the problem, the proposed Benders branch-and-cut algorithm was able to solve the majority of test instances within an hour, providing a solution option for modelers wishing to find a provably optimal solution to this variation.

Our models can be used by decision makers to select mitigations with respect to an overall budget or cardinality requirement, as well as multiple choice limitations. In reality, there could be additional selection requirements. These can be easily adapted into our models and algorithms by removing or adding certain constraints. However, it is not clear if a polynomial-time $(1-1/e)$ algorithm exists for solving submodular maximization problems subject to multiple general knapsack or linear constraints. Work is in progress to develop approximation algorithms with guaranteed performance ratios for other problem variants with special structures.

Acknowledgements

This work was funded by the National Science Foundation Award 1422768. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation. The authors would like to thank Dr. Gio Kao at Sandia National Laboratory for his guidance and feedback on the research results reported in this paper. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

[1] U. S. G. A. Office, "National security-related agencies need to better address risks," tech. rep., GAO-12-361, Washington, D.C., 2012.

[2] J. C. Clapper, "Worldwide Threat Assessment to the Senate Select Committee on Intelligence," tech. rep., Office of the Director of National Intelligence, Washington, D.C., 2012.

[3] U. S. G. A. Office, "National strategy, roles, and responsibilities need to be better defined and more effectively implemented," tech. rep., GAO-13-187, Washington, D.C., 2013.

[4] U. S. G. A. Office, "Cybersecurity actions needed to address challenges facing federal systems," tech. rep., GAO-15-573T, Washington, D.C., 2015.

[5] T. W. House, "Improving critical infrastructure cybersecurity," Executive Order EO-13636, Office of the Press Secretary, Washington, D.C., 2013.

[6] T. W. House, "Critical infrastructure security and resilience," Presidential Policy Directive PPD-21, Office of the Press Secretary, Washington, D.C., 2013.

[7] President's Commission on Enhancing National Cybersecurity, "Report on securing and growing the digital economy," tech. rep., Washington, D.C., 2016.

[8] T. W. House, "Cybersecurity Policy," National Security Presidential Directive NSPD-54, Washington, D.C., 2008.

[9] D. S. Blair, F. M. McCrory, and G. K. Kao, "Supply chain risk management: The challenge in a digital world," Tech. Rep. SAND2015-3667C, Sandia National Laboratories, Albuquerque, NM, 2015.

[10] J. Boyens, C. Paulsen, R. Moorthy, N. Bartol, and S. A. Shankles, "Supply chain risk management practices for federal information systems and organizations," *NIST Special Publication*, vol. 800, no. 161, p. 1, 2014.

[11] G. K. Kao, J. Hamlet, R. Helinski, M. Shakamuri, H. W. Lin, and J. T. Michalski, "Supply chain security decision analytics: Macro analysis.," Tech. Rep. SAND2015-4070C, Sandia National Laboratories, Albuquerque, NM, 2015.

[12] N. Edwards, G. Kao, J. Hamlet, J. Bailon, and S. Liptak, "Supply chain decision analytics: Application and case study for critical infrastructure security," Tech. Rep. SAND2015-2587C, Sandia National Laboratories, Albuquerque, NM, 2016.

[13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 420–429, ACM, 2007.

[14] A. Afful-Dadzie and T. T. Allen, "Data-driven cyber-vulnerability maintenance policies," *Journal of Quality Technology*, vol. 46, no. 3, pp. 234–250, 2014.

[15] Y. Zhuo and S. Solak, "Measuring and optimizing cybersecurity investments: A quantitative portfolio approach," in *Proceedings of the IIE Annual Conference*, Institute of Industrial and Systems Engineers (IISE), 2014.

[16] A. Nagurney, L. S. Nagurney, and S. Shukla, "A supply chain game theory framework for cybersecurity investments under network vulnerability," in *Computation, Cryptography, and Network Security*, pp. 381–398, Springer, 2015.

[17] A. Nagurney and S. Shukla, "Multifirm models of cybersecurity investment competition vs. cooperation and network vulnerability," *European Journal of Operational Research*, vol. 260, no. 2, pp. 588–600, 2017.

[18] R. Church and C. ReVelle, "The maximal covering location problem," in *Papers of the Regional Science Association*, vol. 32, pp. 101–118, Springer, 1974.

[19] J. B. Goldberg, "Operations research models for the deployment of emergency services vehicles," *EMS Management Journal*, vol. 1, no. 1, pp. 20–39, 2004.

[20] L. Brotcorne, G. Laporte, and F. Semet, "Ambulance location and relocation models," *European Journal of Operational Research*, vol. 147, no. 3, pp. 451–463, 2003.

[21] S. H. Jacobson, J. L. Virta, J. M. Bowman, J. E. Kobza, and J. J. Nestor, "Modeling aviation baggage screening security systems: a case study," *IIE Transactions*, vol. 35, no. 3, pp. 259–269, 2003.

[22] S. H. Jacobson, L. A. McLay, J. E. Kobza, and J. M. Bowman, "Modeling and analyzing multiple station baggage screening security system performance," *Naval Research Logistics*, vol. 52, no. 1, pp. 30–45, 2005.

[23] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley & Sons, 2nd ed., 2013.

[24] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[25] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[26] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Research Letters*, vol. 32, no. 1, pp. 41–43, 2004.

[27] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—II," in *Polyhedral Combinatorics*, pp. 73–87, Springer, 1978.

[28] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pp. 67–74, ACM, 2008.

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

[30] A. A. Ageev and M. I. Sviridenko, "Pipage rounding: A new method of constructing algorithms with proven performance guarantee," *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 307–328, 2004.

[31] A. Badanidiyuru and J. Vondrák, "Fast algorithms for maximizing submodular functions," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1497–1514, Society for Industrial and Applied Mathematics, 2014.

[32] N. Buchbinder, M. Feldman, and R. Schwartz, "Comparing apples and oranges: Query trade-off in submodular maximization," *Mathematics of Operations Research*, vol. 42, no. 2, pp. 308–329, 2016.

[33] A. Kulik, H. Shachnai, and T. Tamir, "Approximations for monotone and nonmonotone submodular maximization with knapsack constraints," *Mathematics of Operations Research*, vol. 38, no. 4, pp. 729–739, 2013.

[34] B. Schneier, "Attack trees," *Dr. Dobb's Journal*, vol. 24, no. 12, pp. 21–29, 1999.

[35] S. Mauw and M. Oostdijk, "Foundations of attack trees," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 186–198, Springer, 2005.

[36] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 273–284, IEEE, 2002.

[37] A. Shostack, *Threat Modeling: Designing for Security.* Wiley Publishing, 2014.

[38] T. Storch, "Toward a trusted supply chain: a risk based approach to managing software integrity," tech. rep., Microsoft Corporation, 2014.

[39] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 479–502, 2002.

[40] C. Chekuri and A. Kumar, "Maximum coverage problem with group budget constraints and applications," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 72–83, Springer, 2004.

[41] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[42] A. Koç, D. P. Morton, E. Popova, S. M. Hess, E. Kee, and D. Richards, "Prioritizing project selection," *The Engineering Economist*, vol. 54, no. 4, pp. 267–297, 2009.

[43] A. Koç and D. P. Morton, "Prioritization via stochastic optimization," *Management Science*, vol. 61, no. 3, pp. 586–603, 2014.

[44] A. M. Geoffrion, "Generalized benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 1972.

[45] M. Bodur, S. Dash, O. Günlük, and J. Luedtke, "Strengthened benders cuts for stochastic integer programs with continuous recourse," *INFORMS Journal on Computing*, vol. 29, no. 1, pp. 77–91, 2016.

[46] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," in *Proceedings of the 43rd Hawaii International Conference on System Sciences*, IEEE, 2010.

[47] J. E. Beasley, "A Lagrangian heuristic for set-covering problems," *Naval Research Logistics*, vol. 37, no. 1, pp. 151–164, 1990.

[48] A. Caprara, P. Toth, and M. Fischetti, "Algorithms for the set covering problem," *Annals of Operations Research*, vol. 98, no. 1-4, pp. 353–371, 2000.