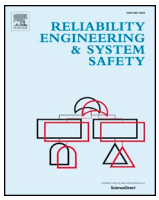
Reliability Engineering and System Safety 205 (2021) 107093

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress



Risk management for cyber-infrastructure protection: A bi-objective integer programming approach

Adam Schmidt[a], Laura A. Albert[*,a], Kaiyue Zheng[b]

[a] Department of Industrial and Systems Engineering, University of Wisconsin-Madison, 53706, United States
[b] Amazon, Seattle, WA 98121, United States

ARTICLE INFO

Keywords:
Cyber-security
Information and communication technology security
Bi-objective optimization
Supply chain security
Risk management
Risk threshold

ABSTRACT

Information and communication technology supply chains present risks that are complex and difficult for organizations to manage. The cost and benefit of proposed security controls must be assessed to best match an organizational risk tolerance and direct the use of security resources. In this paper, we present integer and stochastic optimization models for selecting a portfolio of security controls within an organizational budget. We consider two objectives: to maximize the risk reduction across all potential attacks and to maximize the number of attacks whose risk levels are lower than a risk threshold after security controls are applied. Deterministic and stochastic bi-objective budgeted difficulty-threshold control selection problems are formulated for selecting mitigating controls to reflect an organization's risk preference. In the stochastic problem, we consider uncertainty as to whether the selected controls can reduce the risks associated with attacks. We demonstrate through a computational study that the trade-off between the two objectives is important to consider for certain risk preferences and budgets. We demonstrate the value of the stochastic model when a relatively high number of attacks are desired to be secured past a risk threshold and show the deterministic solution provides near optimal solutions otherwise. We provide an analysis of model solutions.

1. Introduction

Reliance on information and communication systems is ubiquitous in both private and public sectors. The high complexity of and resources required to maintain information and communication technology (ICT), or "cyber", systems create risk that are difficult to manage. ICT systems are continuously at risk of attacks by individuals, organizations, and foreign governments [36]. Organizations wishing to secure their ICT infrastructure must continually make strategic supply chain security investments [3,30]. Estimates suggest that up to 80% of cyber breaches originate in the supply chain and are not due to attacks on local networks [40]. Supply chain security initiatives protect organizations against risk associated with their suppliers or purchasers, especially those with access to sensitive information of their organizations [40]. According to the National Institute of Standards and Technology (NIST), ICT supply chain risks include insertion of counterfeit equipment, unauthorized production, tampering, theft, insertion of malicious software and hardware, poor manufacturing and development practices in the ICT supply chain, and other methods [34]. Both adversarial and non-adversarial risks must be addressed by an organization [3] since ICT systems are also at risk to accidental data loss, accidental access to confidential data, weather related disasters, or flawed component design [34]. Risk to ICT systems can be reduced by selecting from proposed security controls to secure vulnerabilities while considering one's risk preference and the costs and benefits of the proposed controls [3,30]. Since it is impossible to eliminate all risk, an organization should aim to reduce risk to an acceptable level or threshold [3]. To do so, NIST recommends managing supply chain risks with a structured approach that uses well-defined goals and scope to represent threat scenarios.

Securing ICT infrastructure requires multiple levels of planning and multiple processes in place that span protection, detection, response, and recovery [16,30]. This paper is concerned with a planning problem to identify a cost-effective set of security controls to protect ICT infrastructure systems against risks. This planning problem addresses a long-term phased rollout to support supply chain risk management over a time horizon of, say, one year. We contribute optimization models based on integer and stochastic programming that serve as a tool to plan investment into ICT infrastructure security using the guidelines proposed by NIST. In these models, we consider two objectives. The first is to maximize the security against all potential attacks on the system. The second is to maximize the weighted number of attacks

* Corresponding author.
E-mail addresses: apschmidt2@wisc.edu (A. Schmidt), laura@engr.wisc.edu (L.A. Albert), kay.zheng@wisc.edu (K. Zheng).

https://doi.org/10.1016/j.ress.2020.107093
Received 1 October 2019; Received in revised form 11 May 2020; Accepted 22 June 2020
Available online 17 September 2020

secured past a risk threshold. The models we present can be used to select a portfolio of security controls subject to a budget constraint. With the bi-objective structure of our models, organizations can investigate the trade-off between increasing the total difficulty to carry-out the attacks and securing threats past a predefined risk-threshold while tailoring the firm's security portfolio to match its risk preference.

1.1. Background

ICT supply chains are complex networks with both physical and virtual vulnerabilities. They are a networks of end-users, policy makers, procurement specialists, systems integrators, network provider, and software/hardware vendors [25]. Disruptions, intentional or unintentional, within the ICT supply chain have the potential to significantly impact traditional supply chains and normal business operations due to organizational reliance on ICT systems. Threats can be intentional or unintentional, and both types can be grouped into three categories: information, systematic, and cyber threats [25]. Information threats include access to view or tamper with information within the supply chain, data theft, or data loss. Systematic threats result from limited visibility within the supply chain leading to malicious or noncompliant introduction of hardware or software. Cyber threats include traditional network attacks.

Supply chain risk management for ICT security has been a problem of national concern for nearly a decade [1], and in 2019 a national emergency was declared to secure the ICT critical infrastructure supply chain in the United States due to growing threats [44]. As ICT supply chain risks are impossible to remove completely, strategic decisions must be made by policy makers, including enterprise security and supply chain risk management teams [3,4]. Strategic decisions regarding supply chain risk management require the elicitation of policy maker risk attitudes. Risk attitudes have been learned and modeled using a variety of techniques such as reference lotteries [6], expected utility theory [6], cumulative prospect theory [6], multicriteria decision analysis [11], and loss exceedance curves [12].

Supply chain risk management is a sensitive topic, and organizations do not always share their practices with the public [4]. As a result, current trends in supply chain risk management are often difficult to identify. A 2015 report by SANS [40] describes best practices in combating cyber risks in the supply chain. They highlight the need to create a vendor management policy, ensure network isolation and segmentation, and internal audits for appropriate access and processes by employees [40]. NIST has published guidelines for managing ICT risk but does not recommend tools to decide which security investments to make [2]. NIST describes the need collaborate with suppliers and include key suppliers in resiliency decisions to manage ICT supply chain risk. For example, Microsoft has adopted a supply chain risk management approach founded on gaining visibility into their supply chain, assessing risk on supplier level by constructing a supplier risk profiling model, and then working with suppliers to improve security against vulnerabilities [31,32]. NIST reports a variety of approaches used by firms with respects organizational management, supplier management, and supplier risk based on a series of interviews with various sized organizations [4]. NIST reports that organizations rarely monitor for risk and have not adopted security measures that comprehensively protect supply chains, and additional guidance and methods are needed to support organizations, especially less mature organizations [4]. These documents suggest that the state of the art practice is, generally, to develop methods to describing their risk (e.g. supplier risk scores), guide their supply chain in building security, and adopt simple heuristic policies that address risk associated with globalized ICT supply chains (e.g., regulate items purchased from a "foreign adversary" [44]).

Protecting critical ICT infrastructure using a structured approach is an emerging area of research. Cost benefit analysis is a commonly proposed approach to selecting security controls [8,16,47], and more specific approaches, such as advanced component traceability, have been proposed [5]. For optimization-based approaches, we refer the reader to a review paper by Enayaty et al. [9] that surveys literature for research that study how to protect critical cyberinfrastructure, including ICT supply chains. They note that some papers discusses methods using a derivative of attack graphs for a structured approach to reduce critical infrastructure risk [9]; we describe this approach in detail as we employ it in this paper.

Attack graphs and attack trees have been widely used to model the steps required to carry out an attack in a structured way for applications extending beyond supply chains [19,27]. Attack graphs and trees characterize the possible attacks against a system and identify protections against such attacks. A stream of papers in the literature seeks to identify defenses uses the attack graph paradigm. Jha et al. [14] and Sheyner et al. [41] investigate how to identify attack graph structures and identify cost-effective methods to improve security of these graphs. Kordy and Widel [20] formulate a novel attack-defense tree (ADTree) problem. Information describing potential attacks and controls are extracted from an ADTree and an exact algorithm for selecting controls is presented. Kordy and Widel further their analysis with ADTrees to present a bottom-up evaluation method for quantifying the trees [21]. Fei et al. present quantifiable ADTrees, which can be used to determine the priority of defense nodes [10].

Recent research has examined how to identify a portfolio of security controls that reduce risk while employing attack graph models. The motivation to solve a portfolio problem to manage ICT supply chain risk is introduced and motivated by Kao et al. [15] and Edwards et al. [8]. They describe the need for a decision analytics framework and tools that consider the supply chain holistically to support supply chain risk management [8,15]. Within the framework, there is a identified need for optimization to select a subset of security controls for supply chain risk [8,15]. Working from this initial framework, Zheng et al. [51] formulate integer and stochastic programming models using budgeted maximal multiple coverage formulations applied to ICT supply chain security. They adapt network security attack graphs to model an attack as a group of steps through vulnerable components of an infrastructure, where a step of an attack is a node in the attack graph. They consider security controls that cover a step, or a node, of an attack. Each covered node of an attack increases security (reduces risk) against the attack at a marginally decreasing value as more nodes in the path are covered. The models are used to select the optimal set of controls such that the total cost does not exceed the organizational budget. The results yield a set of cost-effective security controls that improve the overall security the most on average, however, some of the some attack paths may not be protected. This may be unacceptable in certain circumstances. Zheng and Albert [50] address this issue by proposing alternative models that consider different robust methods to protect against worst case risks, including models that maximize the worst case coverage, minimize the worst case regret, and maximize the average coverage in the $(1-\alpha)$ worst cases (the conditional value at risk). In these models, the worst-case performance is evaluated over scenarios, and therefore, some attack paths may still be unprotected in the optimal solutions. Li et al. [24] present a multiobjective optimization framework to select security measures to maximize the security, system, and state benefit against the "most dangerous" attacks using the idea of a risk tolerance threshold.

Several papers account for adaptive adversaries by formulating defender-attacker models that capture worst-case performance. Nandi et al. [33] study how to interdict attack graphs by deploying security counter measures subject to a budget using a two-stage Stackelberg game model between a defender and an attacker. They formulate the problem as a min-max bi-level mixed-integer linear program and develop an exact algorithm for finding a subset of arcs to interdict on an attack graph. Letchford and Vorobeychik [23] solve a different Stackelberg game problem in which a defender chooses a set of controls that interdict adversarial attack actions while the attacker is capable of executing an optimal attack plan. Zheng and Albert [49] consider how to defend against attacks originating from multiple adversaries, each of

which finds a critical path through an attack surface. The goal is to delay the completion time of these attacks by selecting a cost-effective set of controls. They present new deterministic and stochastic Stackelberg game models to interdict multiple adversaries' attack projects, and they develop a Lagrangian heuristic that identifies near-optimal solutions efficiently.

1.2. Contribution

Previous research on budgeted maximal multiple coverage models [50,51] focuses on how to reduce overall risk under a budget constraint, but not how to systematically reduce risk to an acceptable level for as many potential attacks as possible (i.e., implementation of a risk threshold). Moreover, the previous research generally focuses on a single objective. This paper seeks to fill this knowledge gap by introducing a methodology for selecting a set of controls to reduce the risk to ICT systems posed by supply chains by formulating new bi-objective integer programming models to select investments into ICT supply chain security that consider competing goals. The first model we present in this paper is a budgeted difficulty-threshold control selection (BDTCS) bi-objective model. The first objective considered captures an organization's goal to minimize the risk to all attacks by maximizing the weighted difficulty of attack steps. However, the risk reduction may not be adequate for mitigating the risk associated with some of the potential attacks. To address this issue, we consider a second objective that captures an organization's goal to increase the difficulty of each attack until it is at an acceptable level of risk as defined by a risk threshold. The second objective maximizes the weighted number of attacks past a risk threshold. The objectives represent two aspects decision-makers may consider when selecting security controls to manage risk.

The second model we present, the expected-value budgeted difficulty-threshold control selection (EBDTCS) bi-objective model, is a stochastic variant of the first model. In this model, we consider uncertainty as to whether the security controls are effective in increasing the difficulty of a step in the attack tree. Analagously to the BDTCS problem, the objective is to maximize the expected weighted risk reduction and expected weighted number of attacks past a risk threshold. By considering stochasticity, we are able to understand the implications of uncertain information about vulnerabilities on system security and security control selection.

These models provide methods to identify security control portfolios that increase the security of an ICT supply chain while considering the trade-offs across two criteria. Computational studies are conducted to investigate insights and trade-offs presented by the bi-objective formulation. We investigate the deterministic model and then retrospectively compare the results to those of the stochastic programming model. These computational studies highlight benefits of using the models to reduce the risks of attacks on cyber security supply chains and provide insights for practice.

In Section 2, we define the problem and present our deterministic and stochastic formulations. In Section 3, we conduct computational studies to investigate the results of these models. Section 4 concludes the paper.

2. Problem description

In this section, we introduce two optimization models for selecting controls to support ICT supply chain risk management. In both models, we formalize a collection of possible supply chain attacks, P, that represent systematic risks to supply chains. Each attack $p \in P$ is defined by the steps required to complete it; these steps comprise a set of *vulnerability nodes, N*. The collection of steps that make up an attack, $N_p \subseteq N$, is an *attack path*, which is an abstraction of an *attack graph* [35,51]. In this paper, "attack" describes both adversarial and non-adversarial threats, although we do not explicitly model adaptive adversarial behavior.

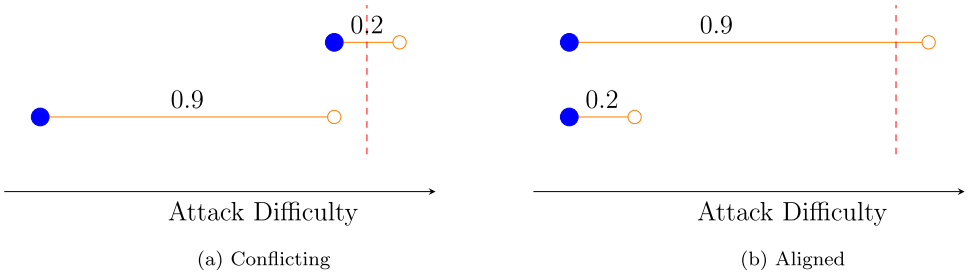
We consider a set of *security controls, M*, each of which has an investment of cost b_m, $m \in M$, and *covers* a subset of nodes if deployed. Coverage of a node increases the difficulty of an attack step, δ_n, from its current difficulty level, d_n, and thus provides a level of protection against the attack. The increase in difficulty, δ_n, does not depend on the security control covering it, since we assume different security controls exploit the same mechanism when covering the same node [51]. Nodes can be defined with a level of granularity such that the security controls either exploit the same mechanism at that node or change the characteristics of the node in such a similar way that the difference in difficulty change between the mitigations is practically insignificant [51]. The model can be trivially adapted to lift this assumption as follows. If two security controls apply different mechanisms to the same node, we can create a copy of the node with each security control separately covering one of the two node copies. Supply chain attack paths and their associated parameter values can be enumerated by subject matter experts (SMEs), generated automatically for well defined supply chains, or learned through risk assessment studies [8,12,41,46].

The models' objective functions present two ways to reduce the risk presented by these attack paths. The first objective is to maximize the total change in difficulty of all attack paths by covering nodes that make up the attack paths. It reflects the goal to reduce overall risk and increase expected difficulty of completing any of the attacks. This objective is called the *difficulty objective* (BDTCS) for the deterministic model or *expected-difficulty objective* (EBDTCS) for the stochastic programming model for the remainder of the paper. The second objective has the goal of ensuring that the risk associated with attack paths are at an acceptable level. We introduce a "consumer grade" or "military grade" risk threshold, T_p, for the difficulty of attack $p \in P$. Risk management literature supports the approach of a risk threshold to define an organization's acceptable level of risk [3,7]. The goal is to select security controls to increase the weighted number of attacks whose difficulties are past their thresholds. This objective is called the *threshold objective* (BDTCS) for the deterministic model or *expected-threshold objective* (EBDTCS) for the stochastic programming model for the remainder of the paper.

To an extent, the objectives are mutually beneficial as they both aim to increase the difficulty of the attacks. However, there is a trade-off for how this difficulty increase is realized. The difficulty objective incentivizes covering steps of paths most likely to increase in overall difficulty, while the threshold objective targets covering steps in the some paths to increase the path's difficulty past the threshold, and no further, within the given budget, B. It is possible for the solution to be the same irrespective of the importance placed on each objective function (i.e., objectives are *aligned*) in specific problems. However, in general this is not the case (i.e., objectives are *conflicting*).

We can explore the alignment or conflict between the objectives more closely through an example. Fig. 1 provides two simple examples, where the objectives are either conflicting (Fig. 1(a)) or aligned (Fig. 1(b)). Both examples consist of two attack paths composed of a single node and two security controls. Each blue circle (●) represents an equally weighted attack, and its placement on the x-axis represents its unmitigated difficulty level. Each attack has a single, unique security control that acts on it. Let the orange circles (○) represent the difficulty of the associated attack if the security control acting on it is selected, and the increase in difficulty is labeled by the number above the line (——), either 0.2 or 0.9. This value is also the contribution to the difficulty objective. The vertical red dashed lines (- - -) represent the security threshold for both attacks. If an attack is past the threshold, then the threshold objective increases by one; otherwise the threshold objective is zero. The budget allows us to select one of the two security controls in each example.

In Fig. 1(a), there are two potential solutions—selecting the first or second security control—and neither solution dominates the other in a multi-criteria optimization sense. The security control that protects against the first attack is selected when a high importance is placed on securing attacks past a threshold as it moves one attack past the



(a) Conflicting (b) Aligned

Fig. 1. Examples where the objective functions are (a) conflicting and (b) aligned. Both (a) and (b) consist of two attack paths, each composed of a single node and with a unique security control. Each blue circle (●) represents an equally weighted attack. Its placement on the x-axis represents its unmitigated difficulty level. The orange circles (○) represent the difficulty of the associated attack if secured against, and the increase in difficulty is labeled by the number above the line (——), either 0.2 or 0.9. This value is also the contribution to the difficulty objective. The vertical red dashed lines (- - -) represent the security threshold for both attacks. If an attack is past the threshold, then the threshold objective increases by one; otherwise the threshold objective is zero. The budget allows us to select one of the two security controls in each example. In (a), neither security control is dominated and the solution will change based on the importance of each objective function. In (b), the security control which protects against the top attack dominates the other and will always be the optimal solution. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

threshold. However, this security control only increases the overall difficulty of attacks by 0.2, and thus when the importance of securing attacks past a threshold is low, the security control that protects against the second attack is selected as it adds a difficulty of 0.9. The example in Fig. 1(b) has the same two potential solutions. The security control that protects against the first attack is always selected because it increases the difficulty by 0.9 and moves 1 attack past path the threshold. This solution dominates the solution of selecting the second security control. When the objectives are conflicting, as in Fig. 1(a), it is important to consider both objectives and strategically select the importance of each objective; this is generally the case in practice. The degree to which the objectives are conflicting or aligned is a property of the problem (e.g., mitigations, attacks, budget) not only the functions. We investigate the trade-off between the two objectives during the computational experiments in a later section.

2.1. Deterministic model

In this subsection, we present the deterministic model. Table 1 reports the sets, parameters, and variables to be used in the models. We begin by introducing the objective functions.

The difficulty objective function captures the weighted difficulty change of all attack paths where a_p is a weight on each path. We define the difficulty objective function as

Table 1
Notation used in the optimization models.

Sets	
P	set of all attacks
N	set of the supply chain vulnerability nodes
N_p	subset of vulnerability nodes in attack path $p \in P$ representing the steps required to carry out the attack, $N_p \subseteq N$
M	set of available controls
M_n	subset of controls that cover vulnerability node n, $n \in N$, $M_n \subseteq M$
Parameters	
d_n	unmitigated difficulty of node n that represents the difficulty associated with completing node n.
δ_n	increase in the difficulty of node n if a selected control impacts it, $n \in N$
a_p	weight of attack p representing the relative importance of protection against it, $p \in P$
b_m	implementation cost for control m, $m \in M$
B	total control budget of the organization
T_p	threshold value for attack p, $p \in P$
Decision variables	
$x_m =$	$\begin{cases} 1 & \text{control } m \text{ is selected, } m \in M \\ 0 & \text{otherwise} \end{cases}$
$z_n =$	$\begin{cases} 1 & \text{if node } n \text{ is covered by a selected control, } n \in N \\ 0 & \text{otherwise} \end{cases}$
$t_p =$	$\begin{cases} 1 & \text{if attack } p \text{ is past the threshold, } p \in P \\ 0 & \text{otherwise} \end{cases}$

$$f_D(z) = \sum_{p \in P} \sum_{n \in N_p} a_p \delta_n z_n$$

The goal is to maximize the weighted change in difficulty across all potential attacks, which captures overall risk reduction. We define the threshold objective function as

$$f_T(t) = \sum_{p \in P} a_p t_p$$

to capture the weighted number of paths whose difficulty levels are past their thresholds. The weight, a_p, captures the importance of securing against the attack $p \in P$. Organizations wishing to ensure protection against certain attacks should place high weights on those attacks. The weight could represent, for example, the expected consequences or expected financial cost of a successful attack, for example.

The weights can be obtained through subject matter expert solicitation [13], multicriteria decision analysis [11], and structured expert judgment methods [37].

The budgeted difficulty-threshold control selection (BDTCS) problem is formulated as a bi-objective, 0–1 integer program as follows:

BDTCS:

$$\max \quad \left[f_D(z) = \sum_{p \in P} \sum_{n \in N_p} a_p \delta_n z_n, f_T(t) = \sum_{p \in P} a_p t_p \right] \tag{1}$$

$$\text{s.t.} \quad z_n \leq \sum_{m \in M_n} x_m n \in N \tag{2}$$

$$t_p T_p \leq \sum_{n \in N_p} d_n + \delta_n z_n p \in P \tag{3}$$

$$t_p A + T_p \geq \sum_{n \in N_p} d_n + \delta_n z_n p \in P \tag{4}$$

$$\sum_{m \in M} b_m x_m \leq B \tag{5}$$

$$x_m, \ z_n, \ t_p \in \{0, 1\} m \in M, \ n \in N, \ p \in P \tag{6}$$

The objectives (1) are to maximize the weighted difficulty change of all the paths, $f_D(z)$, and maximize the weighted number of paths past their thresholds, $f_T(t)$. Constraint set (2) determines if a node is covered by at least one selected control. Constraint set (3) requires each attack's difficulty (starting difficulty plus any difficulty changes after implementing controls) to meet the threshold if the variable t_p equals to 1 (i.e., the attack is past the threshold). Constraint set (4) forces the threshold variable t_p to be 1 if the attack's difficulty is greater than the threshold, where A is a large value. Constraint (5) is the budget constraint on selected controls. Constraint set (6) restricts the x_m, z_n, and t_p variables to be binary.

A single unique solution is not guaranteed to maximize both

objective functions. Instead we use the ε-constraint method to approximate the complete Pareto set of this problem for a posteriori decision by decision makers. The ε-constraint method reformulates the problem to maximize a single objective function, (7), and bounds all other objective function values (OFVs) using additional constraints, (8), as defined below. The bounds on the objective functions are then varied and the model is solved to enumerate solutions in the Pareto optimal set. Our problem is subsequently reformulated as:

$$\max f_D(z_n) \tag{7}$$

$$\text{s.t.} \quad f_T(t_p) \geq \varepsilon \quad \text{Constraints (2)} - \text{(6)} \tag{8}$$

The ε-constraint method allows the budget constraint (5) to be considered as a third objective function to be minimized. This becomes particularly useful to organizations without a fixed budget at the time of planning who wish to investigate the implications of various budget levels. We explore this issue in Section 3.

2.2. Stochastic model

In the BDTCS problem, we assume a node's difficulty can be increased by a control with certainty. However, the effect of these controls may be uncertain in reality due to unknown characteristics of a given step of an attack or interactions between steps of a specific attack (e.g., a supplier fails to complete a security training). Modeling probability distributions, uncertainty intervals, and stochastic scenarios, as opposed to point estimates and expected values, is widely used to guide cybersecurity risk management decisions [12,37,39]. Modeling uncertainty with respect to the effect of the controls on the nodes' difficulty levels is particularly important for the threshold objective, since knowing that a set of controls may not move a path's difficulty level past its threshold with certainty may affect a decision-maker's willingness to invest in those controls. This motivates the need for a stochastic programming extension to the BDTCS problem. Moreover, Zheng et al. [51] note that stochastic programming models offer substantial value to decision makers over equivalent deterministic models. For these reasons, it is necessary to formulate and investigate a stochastic variant of the deterministic problem.

The computational results presented in Section 3 highlight the practical importance of using a stochastic model in certain situations for our problem.

In this section, we present an extension to the BDTCS problem that considers this uncertainty. We introduce a stochastic program that considers the ineffectiveness of all controls in the covering set, M_n, on node n, $n \in N$, when the node s in attack p, $p \in P$. This models the situation in which if one control in the set is ineffective again a node in a path, all controls in the covering set will also be ineffective since we assume controls exploit the same mechanism. When the control set is ineffective, the difficulty of the node cannot be increased. Under these uncertainties, there is a natural incentive to cover multiple nodes in the path. There are multiple alternative ways to represent uncertainty in our problem. Certain controls in the covering set could be ineffective against a node [51] or the change in difficulty could be a random variable [49]. We selected the method presented in this paper because it has not been researched in depth in a comparable situation, and it is believed to have a large impact on solutions with our threshold formulation.

We introduce a finite-scenario set $\xi = \{\xi^\omega, \omega \in \Omega = \{1, 2, \ldots, |\Omega|\}\}$, to represent the uncertainty in the model. Each scenario ξ^ω, $\omega \in \Omega$, occurs with probability ρ_ω with $\sum_{\omega\in\Omega} \rho_\omega = 1$. Each scenario is comprised of parameters ξ^ω_{pn} that take on the value 1 if node n, $n \in N$, is vulnerable to its control set in attack p, $p \in P$, and 0 otherwise. We consider the function $f_D(x, \xi)$ which represents the difficulty objective value, and $f_T(x, \xi)$ which represents the threshold objective value based on the controls selected, x, and the scenarios of the scenario set ξ. These functions take the same form as in the BDTCS problem except they consider the uncertainty within the model formulation. The intent is to study how to improve the expected security of the supply chain, $[\, \mathbf{E}_\xi[f_D(x, \xi)],\ \mathbf{E}_\xi[f_T(x, \xi)]\,]$, under analogous constraints as BDTCS in a future section.

The expected-threshold and expected-difficulty objective functions of the expected-value of the finite-scenario BDTCS problem (EBDTCS) respectively are:

$$\mathbf{E}_\xi[f_D(x, \xi)] = \sum_{\omega\in\Omega} p_\omega f_D(z^\omega, \xi^\omega)$$
$$\mathbf{E}_\xi[f_T(x, \xi)\ ] = \sum_{\omega\in\Omega} p_\omega f_T(t^\omega, \xi^\omega)$$

Notation and variables in EBDTCS are extended from the deterministic model in a straightforward manner. Variables indexed by superscript $\omega \in \Omega$ have the same meaning as those in Table 1 except that they are defined and valued for each scenario ξ^ω, $\omega \in \Omega$. The finite-scenario EBDTCS problem is presented as a stochastic bi-objective integer programming model below.

EBDTCS:

$$\max \quad [\, \mathbf{E}_\xi[f_D(x, \xi)] = \textstyle\sum_{\omega\in\Omega} \rho_\omega f_D(z^\omega, \xi^\omega),\ \mathbf{E}_\xi[f_T(x, \xi)\ ] = \sum_{\omega\in\Omega} \rho_\omega f_T(t^\omega, \xi^\omega)\,] \tag{9}$$

$$\text{s.t.} \quad z_n^\omega \leq \sum_{m\in M_n} x_m \quad n \in N, \omega \in \Omega \tag{10}$$

$$t_p^\omega T_p \leq \sum_{n\in N_p} d_n + \delta_n \xi^\omega_{pn} z_n^\omega \quad p \in P, \omega \in \Omega \tag{11}$$

$$t_p^\omega A + T_p \geq \sum_{n\in N_p} d_n + \delta_n \xi^\omega_{pn} z_n^\omega \quad p \in P, \omega \in \Omega \tag{12}$$

$$\sum_{m\in M} b_m x_m \leq B \tag{13}$$

$$x_m,\ z_n^\omega,\ t_p^\omega \in \{0, 1\} \quad m \in M, n \in N, p \in P, \omega \in \Omega \tag{14}$$

The objective (9) maximizes the expected weighted difficulty and the expected weighted number of paths past the threshold given the finite-scenario set. Constraint set (10) determines if a node for scenario ξ^ω, $\omega \in \Omega$ is covered by a selected control. Constraint set (11) requires the attack p to be past the threshold for scenario ξ^ω, $\omega \in \Omega$ to realize the gains in the threshold objective function. Constraint set (12) forces t_p^ω to be 1 if the path difficulty is greater than the threshold for scenario ξ^ω, $\omega \in \Omega$ where A is a large value. The knapsack budget constraint is represented in constraint (13) and constraints (14) are binary variable constraints.

Similar to the deterministic model, we can reformulate EBDTCS using the ε-constraint method to ensure all non-dominated solution can be found by varying ε.

$$\max \sum_{\omega\in\Omega} \rho_\omega f_D(z^\omega, \xi^\omega) \tag{15}$$

$$\text{s.t.} \sum_{\omega\in\Omega} \rho_\omega f_T(t^\omega, \xi^\omega) \geq \varepsilon \quad \text{Constraints (10)} - \text{(14)} \tag{16}$$

The ε-constraint method maximizes the value of the expected-difficulty function, Eq. (15), and constrains the value of the expected-threshold function in (16). Without loss of generality, functions $\mathbf{E}_\xi[f_D(x, \xi)]$ and $\mathbf{E}_\xi[f_T(x, \xi)]$ can be exchanged in (15) and (16).

The finite-scenario sets can be large in general. When this occurs, we can instead solve an approximate stochastic problem using sample average approximation [17]. In sample average approximation, we take a finite set of samples of ξ and approximate the expected value function with a sample average. We take this approach when solving the models

in the computational results section.

2.3. Solution methods

A multi-objective optimization can be approached using either priori, interactive, and posteriori methods [45]. We employ the posteriori method as it requires no initial decision maker input and provides a set of "optimal" solutions that either fully enumerates or approximates the optimal solution set. The set of "optimal" solutions provided by posteriori methods is called the non-dominated, Pareto optimal, efficient, or non-inferior solution set.

Algorithms can employ the ε-constraint method to approximate the Pareto frontier by solving for a given number of solutions [22,28] or for a complete enumeration of the Pareto set [29,48]. In this paper, we approximate the Pareto set using a fixed number of solution points. To determine the range of ε's considered in the BDTCS problem, we calculate the maximum and minimum attainable OFVs in a non-dominated solution set, while fixing the budget. To determine the minimal attainable threshold OFV, we solve the instance of the BDTCS problem with $\varepsilon = 0$ and retrospectively calculate the threshold OFV under this solution, setting $\varepsilon_{\min} = f_T(\cdot)$. We then solve the instance of the BDTCS problem to determine the maximum attainable threshold OFV, and let $\varepsilon_{\max}$ equal this maximal threshold OFV. The parameters $\varepsilon_{\min}$ and $\varepsilon_{\max}$ define the range for ε in the Pareto set. We use an analogous approach for the EBDTCS problem.

The most common posteriori solution approach in multi-objective optimization is to use the weighted sum of objectives method; however, for mixed-integer multi-objective optimization problems, this method is not guaranteed to provide the complete set of non-dominated solutions [28]. Heuristic algorithms, such as evolutionary and genetic algorithms, have been heavily studied as multi-objective optimization solution methods to approximate the Pareto frontier without any guarantees on Pareto efficiency [18,43]. Dynamic programming and lexicographic optimization are other methods that have been used to solve multi-objective optimization problems [26,38]. We focus on the ε-constraint method, since it can easily be used to control which solutions on the Pareto frontier are obtained from a problem instance and can provide the complete set of non-dominated solutions [29,48].

3. Computational results

In this section, we illustrate solutions to the deterministic and stochastic models for optimal control selection to protect against potential attacks in an ICT supply chain. This section contains three subsections: (1) case studies, (2) deterministic model results, and (3) stochastic model results. The results provide insight into the model solutions and the tradeoffs between the multiple objectives. The models were programmed and run with 64 bit Python 2.7.15 and were solved using Gurobi 8.1.0. The tests were run on an Intel® Core™ i5-7500U CPU at 3.40 GHz with 16 GB of RAM.

3.1. Case studies

To test and evaluate the proposed models, we formulate realistic data with varying sizes and values in conjunction with information provided by collaborators at Sandia National Laboratories and information provided in relevant literature. Real data are sensitive and not available for release, and therefore, collaborators at Sandia National Laboratories developed a model of attack path vulnerabilities suitable for this and similar optimization research that would be informative to their decision-making processes. We used their models to construct hypothetical yet realistic data sets for our computational experiment, and we note that our analysis using hypothetical data sets may have natural limitations for interpretability and reliability of insights.

A detailed explanation of the data used for the computational studies is provided below. A summary of the data regarding the size of six data sets used for the computational study are provided in Table 2. There are two small, two medium, and two large data sets.

Table 2
Data set sizes of the case studies.

Data set	$\lvert M\rvert$	$\lvert N\rvert$	$\lvert P\rvert$
s1	50	50	10
s2	100	100	20
m1	500	500	100
m2	500	500	75
ℓ1	1000	1000	200
ℓ2	2000	1000	200

We generate the sample data using a pseudo-random number generator and adopt the following logic. Controls are unlikely to impact a large number of nodes as they are generally specific in scope and impact. Thus, we specify an upper bound of three on the number of nodes a control can affect and a lower bound of one. Also, it is unlikely a single node will be impacted by many controls, therefore we specify the upper bound on the number of controls that can affect each node to be three with no lower bound. For each control, we randomly select a subset of nodes that the control covers under these restrictions.

Similarly, we set a lower bound of 5 and an upper bound of 10 on the number of nodes in a path. Based on the information received from our collaborators, there are relatively few access points for successful control of supply chain vulnerabilities. We randomly sample from the set of nodes N to generate a subset of nodes N_p in an attack path $p \in P$. We define a node in the path to be the *end node* that represents the final step of the attack and determines the attack consequence, $c_p, p \in P$. In this computational study, we let $a_p = c_p$. Paths can and are likely to share nodes. This allows multiple attacks to have the same "goal" and consequence but with different steps as is likely to be seen in practice. The consequence of each node is uniformly selected over a nominal range [0,1].

We randomly generate the current state difficulty of each node, d_n, $n \in N$, to be negatively correlated to the number of controls covering it. It is likely that nodes that can be covered by more controls are currently more vulnerable and are less difficult to complete. For each d_n, we generate a random number between [0,4] and divide it by the number of controls covering it; if no control covers the node, we divide by one instead of zero. To generate various changes in difficulties, $\delta_n, n \in N$, we uniformly generate a number in the range [0,4]. The difficulty and difficulty change can be determined by various risk analyses, including subject matter expert elicitations [12]. One approach would be to calculate the difficulty, d_n, and difficulty change, δ_n, of an attack from the conditional probability of success if the step is attempted [47]. The difficulty could then be the negative logarithm of the success probability of completing a step. In this case, [0,1] represents the multiplier by which the probability of success for a step decreases with added security.

We randomly generate control costs, $b_m, m \in M$, to be positively correlated to the difficulty change of the nodes impacted by that control; if a control has a large impact on many nodes, it is likely to cost more than a control that has a small impact on a few nodes. For each control, we generate a random value from the range [0,5] and multiply the value by the summation of δ_n values of its covered nodes. Letting $a_p = c_p$, the difficulty objective function becomes $f_D(z) = \sum_{p \in P} \sum_{n \in N_p} c_p \delta_n z_n$ and the threshold function becomes $f_T(t) = c_p t_p$. This captures the idea that attacks with higher consequence are more important to protected against. We define $T_p = 4c_p^2 + 15$ which is scaled to the data and provide a meaningful investigation. The quadratic form represents a need for more security, or higher difficulty, for high consequence attacks to be considered secure. We selected the scaling values so most paths are short of the threshold but a small number are already past before any controls are

implemented, as is likely to be seen in practice.

In the EBDTCS problem, for each scenario and attack path, we randomly determine if the control set M_n is effective in mitigating a node $n \in N$ in attack path $p \in P$ with the probability of 0.9: $Pr(\xi_{pn}^{\omega} = 1) = 0.9$; $Pr(\xi_{pn}^{\omega} = 0) = 0.1$, $\omega \in \Omega$. We let the probability of each scenario be $\rho_{\omega} = \frac{1}{|\Omega|}$ for $\omega \in \Omega$, where $|\Omega|$ is the cardinality of scenario set.

3.2. Deterministic results

In this subsection, we report the results of the deterministic bi-objective problem and perform a sensitivity analysis on select parameters of the BDTCS problem. Solution times associated with solving BDTCS problem instances presented in this paper are less than three minutes.

We start by investigating the competing objectives. With a fixed budget equal to 10% of the cost of all controls, we determine the OFVs for solutions on the BDTCS Pareto frontier. Table 3 provides Pareto optimal solution values to three different instances of the model for each data set. Each solution corresponds to a different ε value: $\varepsilon_{\min}$, ε_{mid}, and $\varepsilon_{\max}$. A solution to an instance with $\varepsilon_{\min}$ only has an explicit incentive to increase $f_D(z)$. A solution to an instance with $\varepsilon_{\max}$ only has an explicit incentive to increase $f_T(t)$. A solution to an instance with ε_{mid} represents one solution near the mid point on the Pareto frontier. From Table 3, we can see the trade-off between the optimal threshold objective, $f_T^*(\cdot)$, and the optimal difficulty objective, $f_D^*(\cdot)$, to different ε-valued instances of BDTCS is significant for each data set. As ε is increased for each data set, the optimal difficulty objective value decreases as the optimal threshold objective value is forced to increase.

Fig. 2 provides 15 solution OFVs on the Pareto frontier for data set $\ell 1$ with a budget set to 10% of the total control costs in the data set. The points marked with a box (□) correspond to the solutions from Table 3. Under a static budget, it is apparent from our computational analysis that not all optimal solutions on the Pareto frontier achieve the same goal. Maximizing just one of these objectives results in a significantly lower OFV of the other. These results indicate the importance of considering both model objectives as the trade-off is nontrivial. Organizations can use available information about the trade-off to match their control selection to their risk preference.

Further investigation suggests certain controls are more important than others. Solutions to data sets $s2$ and $\ell 1$ with B= 10% of total control costs can be seen in Tables 4 and 5, respectively. In Tables 4 and 5, each column represents the solution to the ε-valued BDTCS instance. Each control selected in at least one solution can be seen in the rows of the tables. Controls selected in the solution to a ε-valued instance are indicated with an "x" in the corresponding cell. In all of the data sets studied, there were subsets of controls selected in all studied ε-valued BDTCS instances considered. For example, Table 4 shows that 13 of the 36 controls are selected across all possible values of ε. This suggests there is a subset of controls that are the most critical to implement across all values of the organizational risk tolerance. The difference in OFVs tends to arise from tailored additions to this subset. The model solutions also indicate that nodes are rarely covered two or more times by the selected controls as there is no incentive in BDTCS to do so.

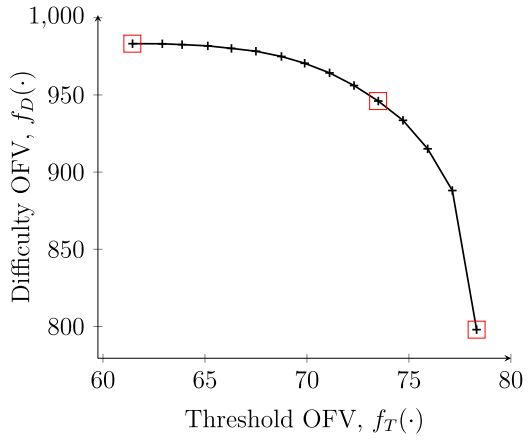


Fig. 2. The Pareto frontier of the difficulty OFVs and threshold OFVs to the BDTCS problem for data set $\ell 1$. Budget was set to 10% of total control costs. OFV's indicated by □ are solutions from Table 3 ($\varepsilon_{\min}$, ε_{mid}, $\varepsilon_{\max}$), and the controls selected in these solutions are provided in Fig. 5.

Next, we directly investigate the consequence, difficulty, and coverage of the attack paths under different Pareto optimal solutions. Fig. 3 provides a comparison of paths under different optimal solutions for data set $\ell 1$ with budget set to 10% of all control costs. The solutions provided in Fig. 3 correspond to the solution OFVs provided in Table 3 and solutions in Table 4. Each mark (+) is a path plotted by its consequence and difficulty. Fig. 3(a) shows the initial position of each attack path in comparison to the risk threshold (—) generated for the $\ell 1$ data set. Fig. 3(b)–(d) show the path consequence and difficulty levels after deploying security controls. The implementation of controls moves the difficulty level of the paths horizontally to the right. Fig. 3(b)–(d) shows difficulty of paths compared to the threshold under various ε values. When ε is low (ε = 61.44), as shown in Fig. 3(b), the solution is to select cost effective controls without consideration of the threshold. This results in some path difficulties much larger than the threshold and thus potentially ineffective use of the budget if an organization believes they do not need to protect against attacks past the threshold. In contrast, when ε is high (ε = 78.32), as shown in Fig. 3(d), the model selects controls that cover nodes in paths whose difficulty can be larger than the threshold. This leaves some paths uncovered and at a very low difficulty. The difficulty of most paths are just past the threshold with only a few path significantly past from its threshold. This trade-off again highlights the need to understand the trade-off between both objectives and to set meaningful thresholds.

Lastly, we investigate the impact of the budget on the tradeoff between the OFVs. Fig. 4 shows results of the trade-off between the two objective functions considering varying level of the budget for data set $\ell 1$. Fig. 4(a) shows the range of the difficulty OFV on the Pareto frontier. At a given budget, the difficulty OFV for any ε-valued instance is bounded by the OFV of each curve. The realized OFV is found within the shaded region and depends on ε. The difficulty OFV is maximal when $\varepsilon = \varepsilon_{\min}$ and minimal when $\varepsilon = \varepsilon_{\max}$. Fig. 4(b) analogously shows the range of the threshold OFV on the Pareto frontier with the OFV being optimal when $\varepsilon = \varepsilon_{\max}$ and minimal when $\varepsilon = \varepsilon_{\min}$. For a low budget, the selected controls are similar between a difficulty-driven model (small ε) and a threshold-driven model (large ε). Therefore the range for both OFVs is small, and any solution in this range will be near-optimal for both objectives. This highlights that, to an extent, these objectives are mutually beneficial and a subset of controls achieve near maximal function values for both objectives with low budget. However, as the budget increases, the range of both OFVs become larger because the model has the budget to tailor controls to the ε-valued instance. In this range, careful selection of the appropriate risk tolerance is critical. As the budget continues to grow, the range of both OFVs become smaller and the curves level off as most meaningful controls in M can be selected with a smaller budget and the objectives are mutually beneficial. This computational study of the deterministic BDTCS problem shows the importance of considering both the threshold objective and difficulty objective, as well as the budget, when approaching IT supply chain security problems. Therefore, defining an organizations risk preference and selecting appropriate security controls is a critical and non-trivial task.

3.3. Stochastic results

In this subsection, we investigate the results of the stochastic problem and retrospectively compare results of the deterministic solution

Table 3
Optimal OFV's, $f_T^*(\cdot)$ and $f_D^*(\cdot)$, for BDTCS instances with ε_{min}, ε_{mid}, and ε_{max} for each data set. Budget was set to 10% of total control costs for each data set.

Data set	ε_{min}			ε_{mid}			ε_{max}		
	ε_{min}	$f_T^*(\cdot)$	$f_D^*(\cdot)$	ε_{mid}	$f_T^*(\cdot)$	$f_D^*(\cdot)$	ε_{max}	$f_T^*(\cdot)$	$f_D^*(\cdot)$
$s1$	2.71	2.71	52.94	3.32	3.56	50.04	3.56	3.56	38.91
$s2$	3.89	3.89	88.16	5.34	5.47	81.29	5.92	5.92	69.61
$m1$	27.91	27.91	452.17	32.84	32.94	443.51	34.80	34.80	363.82
$m2$	19.46	19.46	332.68	23.03	23.03	328.09	24.45	24.45	296.89
$\ell 1$	61.44	61.44	983.19	73.49	73.50	946.02	78.32	78.32	797.95
$\ell 2$	56.37	56.37	632.64	66.89	66.93	596.51	71.10	71.10	431.85

to understand solution differences. We generate for each scenario ξ^{ω}, $\omega \in \Omega$ a sample set which captures the uncertainty that the control set M_n is effective in mitigating the node $n \in N$ in attack path $p \in P$. For each data set, we solve instances with $|\Omega| = 5,\ 25,\ 100$ while using the same base parameter values as the deterministic model, except for ε. We investigate these sizes of Ω to understand the impact of the number of uncertainty scenarios on solution results and interpretation. When comparing models, however, the BDTCS objective function value cannot be directly compared to values of the EBDTCS problem as the results are not to the same problem. Thus, we retrospectively evaluate control selections of the BDTCS problem and determine the expected-difficulty and expected-threshold OFVs under the same uncertainty scenarios considered in the EBDTCS problem. We can then directly compare the solutions. We define the value of the stochastic solution (VSS) as follows: when fixing a value of the expected-threshold objective value and budget, the VSS is the difference between the EBDTCS expected-difficulty OFV and BDTCS retrospective expected-difficulty OFV.

Solution times to EBDTCS rapidly increase even for moderately sized scenarios, as shown in Table 6. Investigation into these solution times indicate significantly larger solution times when $\varepsilon = \varepsilon_{max}$ for most data sets, some of which did not solve in 2 h. A smaller scenario set for the stochastic problem may be sufficient for some applications as even a small number of scenarios provide improved model accuracy compared to the deterministic solution.

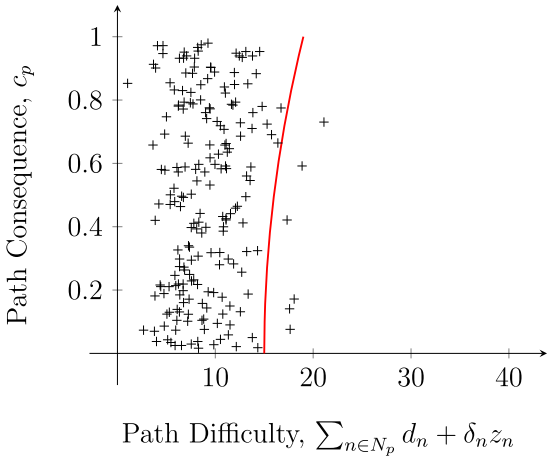
Fig. 5 plots the deterministic BDTCS solutions' retrospective OFVs (■) against the stochastic Pareto optimal frontier (—•—) for data set m2, $|\Omega| = 25$, and B= 10% of all control costs. Insights can be gained from this comparison. A BDTCS solution does not achieve a solution beyond the stochastic Pareto frontier, otherwise the same solution would be

Table 4
Controls selected in 15 solutions on the Pareto frontier for data set $s2$. An (x) indicates that the control was selected in the solution to the ε-valued instance. Controls not shown in the figure were not selected in any of the solutions. Budget was set to 10% of total control costs.

	ε														
Control	3.89	4.03	4.18	4.32	4.47	4.62	4.76	4.91	5.05	5.20	5.34	5.49	5.63	5.78	5.92
M1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M5	x	x	x	x	x	x				x					
M6	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M11		x	x	x	x	x				x					
M12										x					
M21	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M29											x		x	x	x
M30							x	x	x				x	x	
M32	x		x			x				x		x			
M33	x	x	x	x	x	x	x	x	x	x	x				
M37	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M42	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M43	x	x	x	x	x	x	x	x	x	x	x	x			
M46	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M49		x	x	x	x	x	x	x	x	x	x	x	x	x	x
M50		x		x	x		x	x	x	x					
M51	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M52	x														
M53	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M54															x
M57	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M58	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M61	x	x	x	x	x	x									
M63												x			
M65	x	x	x	x	x	x	x	x	x	x					
M67	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
M68	x	x	x	x	x	x	x	x	x		x	x	x	x	x
M73	x	x	x	x	x	x				x	x		x	x	
M74	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
M75	x	x	x	x	x	x	x	x	x	x					
M77	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
M80	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M82	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M83	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
M85												x			
M95	x	x	x	x	x	x					x				

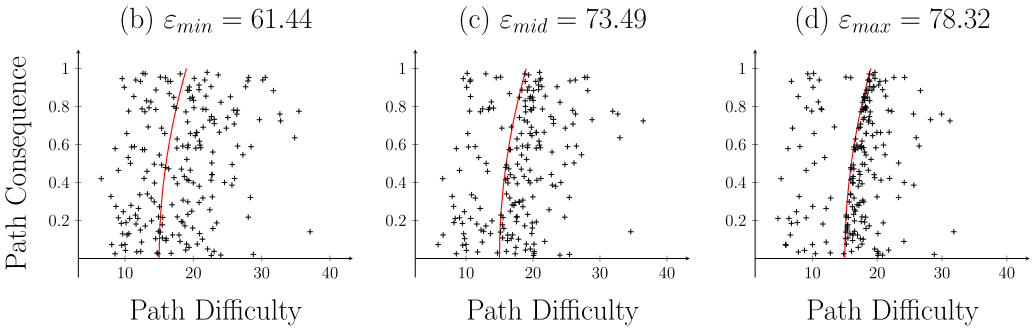
Table 5
Controls selected in three solutions corresponding to $\varepsilon_{\min}$, ε_{mid}, $\varepsilon_{\max}$ with data set $\ell 1$. An (x) indicates that the control was selected in the solution to the ε-valued instance. Controls not identified in the figure were not selected in any of the three solutions. Budget was set to 10% of total control costs.

	ε				ε				ε				ε				ε				ε			
Control	min	mid	max		Control	min	mid	max	Control	min	mid	max	Control	min	mid	max	Control	min	mid	max	Control	min	mid	max
M8	x				M185	x		x	M331	x	x	x	M526	x	x	x	M726	x		x	M835	x	x	x
M11	x	x	x		M188	x	x	x	M333	x	x	x	M531	x	x	x	M736	x	x		M837		x	x
M16	x				M189			x	M334	x	x		M532	x	x	x	M738	x			M839		x	x
M19	x	x			M193	x	x		M340	x	x	x	M535	x	x	x	M740	x	x	x	M842		x	x
M25	x	x	x		M195	x	x		M345	x	x		M542		x	x	M744			x	M845	x	x	x
M27	x	x	x		M196	x	x		M351			x	M550	x	x		M745	x	x	x	M846	x	x	x
M31	x	x			M198	x		x	M358		x	x	M553	x	x	x	M749	x	x	x	M847	x	x	
M32			x		M202	x	x	x	M369	x	x		M556	x	x		M750	x			M858	x	x	x
M36	x	x			M203	x	x		M370	x	x		M559	x	x	x	M752			x	M863		x	
M41			x		M204			x	M380	x	x	x	M562	x	x		M753	x	x	x	M864	x	x	x
M42		x	x		M208	x	x	x	M382	x	x		M569		x	x	M754	x	x	x	M867		x	x
M43	x	x			M210	x	x	x	M385	x	x		M570			x	M758	x	x	x	M868	x	x	
M44	x				M211		x		M389	x	x	x	M571	x	x	x	M761			x	M869	x	x	x
M46	x	x	x		M212			x	M390		x	x	M573	x			M762	x	x		M871	x	x	
M50	x	x	x		M213	x	x	x	M392	x	x	x	M577	x	x		M763	x			M878	x	x	x
M51		x	x		M214	x	x	x	M394			x	M582	x			M764	x	x		M880	x	x	x
M52			x		M218	x			M400		x		M593	x			M765		x	x	M882	x	x	
M55	x	x			M219	x		x	M403			x	M598	x	x	x	M766	x	x	x	M883	x	x	x
M66	x	x	x		M222	x	x	x	M406	x	x	x	M600	x		x	M768			x	M887	x	x	
M68	x	x			M223	x	x	x	M411		x	x	M601	x	x	x	M770			x	M892	x	x	x
M70	x	x	x		M226	x	x		M416	x	x		M609	x	x	x	M771	x			M893	x	x	x
M73	x	x	x		M230	x	x	x	M417	x	x	x	M611			x	M772	x	x	x	M896	x	x	x
M77		x	x		M232	x	x		M425	x	x	x	M618	x	x	x	M775	x	x		M897	x		
M84	x	x	x		M240	x	x		M428	x	x	x	M624			x	M778	x			M898	x	x	x
M86	x	x			M247	x			M433	x	x	x	M627	x			M779	x	x	x	M913	x	x	x
M89	x	x			M249	x	x		M435	x	x	x	M631	x	x		M782	x	x		M915	x	x	
M95	x	x	x		M258	x	x		M437	x	x	x	M632	x	x	x	M787		x	x	M925	x	x	x
M98	x				M259	x	x	x	M439	x	x	x	M638	x	x	x	M789			x	M930	x	x	x
M104	x	x	x		M263	x			M441			x	M644	x	x	x	M792	x			M931	x	x	x
M107	x	x	x		M272	x	x	x	M442	x	x		M646	x	x	x	M795	x	x		M938	x	x	
M110	x	x			M276	x	x	x	M444	x	x	x	M651	x	x		M796	x	x	x	M941	x	x	x
M113			x		M277	x	x	x	M445			x	M657	x	x		M797			x	M945			x
M120	x				M279	x	x	x	M446	x	x	x	M663		x	x	M798	x	x	x	M949	x	x	
M121	x	x	x		M282	x	x		M447	x	x	x	M666	x	x		M802	x			M961	x	x	x
M122	x	x			M283			x	M448		x	x	M670	x	x	x	M803	x	x		M969	x	x	x
M127	x	x	x		M289	x	x		M452	x	x		M671	x	x		M805	x	x	x	M970			x
M134	x	x	x		M292	x		x	M460	x			M682	x	x	x	M807	x	x	x	M972	x		
M136	x	x			M293	x	x		M467			x	M689	x			M810	x	x	x	M977	x	x	x
M145			x		M294	x	x	x	M470	x	x	x	M690	x	x		M811	x	x	x	M987	x	x	
M148	x				M297	x	x	x	M472	x	x		M691	x	x	x	M814	x	x	x	M990	x	x	
M161	x	x	x		M300	x	x	x	M475	x	x	x	M694	x	x	x	M818			x	M994	x	x	x
M165	x	x	x		M303		x	x	M483	x	x		M702	x	x	x	M819			x	M995	x		x
M167	x	x	x		M305	x	x	x	M484	x	x	x	M703	x	x	x	M820	x	x		M997	x		
M173	x	x			M306	x	x		M486	x	x		M711		x	x	M823			x	M998	x	x	
M176			x		M308	x	x	x	M488	x		x	M713			x	M825		x	x				
M177	x	x	x		M312		x	x	M492	x	x		M716			x	M827	x	x					
M181	x	x	x		M318	x	x		M507	x	x		M718	x	x		M828	x	x	x				
M183			x		M323	x	x		M517	x	x	x	M720	x	x	x	M829	x	x					

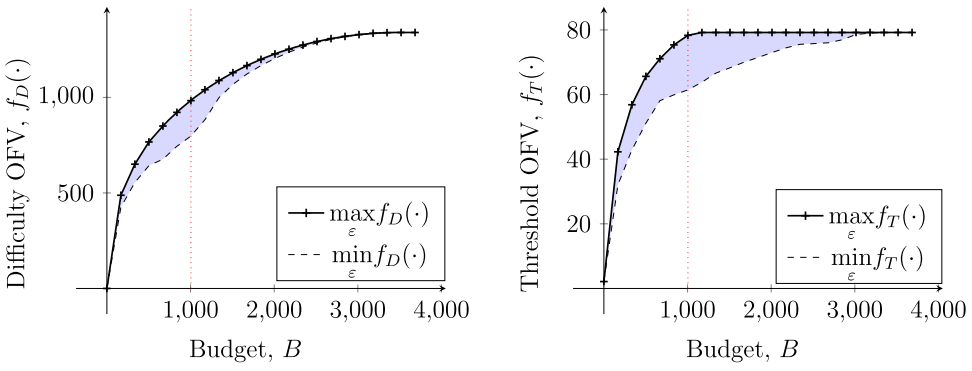


(a) Difficulty and consequence of paths in relation to the risk threshold before any controls are implemented.

Fig. 3. Characteristics of attack paths under various solutions to the BDTCS. Attack paths are plotted (+) in relation to the risk threshold (——) before selecting any controls, (a), and three optimal solutions to the BDTCS problem, (b)–(d), in dataset ℓ1. Budget was set to 10% of total control costs.



chosen by the stochastic problem. The vertical differences between the BDTCS solution and the optimal EBDTCS solution in Fig. 5 represents the value of the stochastic solution. Empirically, the VSS is negligible for a low expected-threshold OFV, and there is an ε value for the EBDTCS problem above which it becomes valuable to consider the uncertainty in the model. This occurs because the threshold function is a binary outcome; the model either realizes all value from a path being past the threshold or it gets no added benefit. Empirically, the high-ε BDTCS instances incentivize control selection resulting in paths just past the threshold and little further—see Fig. 3(d). Thus a loss of coverage on one node can significantly change the path's relation to the threshold. These empirical results suggest that stochasticity is important to consider in threshold driven solutions. The deterministic model suffices in difficulty driven situations and can be solved in significantly less computational time. Further investigation into the solutions highlights the extent of similarity between controls selected in the BDTCS and EBDTCS solutions. Over all sample data sets and $|\Omega|$, we observe that the deterministic and stochastic models select many of the same controls, with an average fraction of 0.914 of the controls selected in the EBDTCS solution also selected comparable BDTCS solutions. We provide a comparison of BDTCS and EBDTCS solutions for three different data sets in Table 7(a)–(c). Table 7(a) presents comparisons for data set s1 with $|\Omega| = 100$. Table 7(b) presents comparisons for data set m2 with $|\Omega| = 100$. Table 7(c) presents comparisons for data set ℓ1 with $|\Omega| = 100$. Each row in each table presents a BDTCS solution and a corresponding EBDTCS solution. To find corresponding solutions, we took an optimal solution to an BDTCS problem instance, computed the retrospective expected-threshold OFV, *Retro-*$f_t(\cdot)$, and solved an instance of EBDTCS with $\varepsilon = $ *Retro*$-f_t(\cdot)$. This allows us to directly compare BDTCS and EBDTCS solutions. Rows with a low reference number in Table 7(a)–(c) correspond to BDTCS instances with relatively low ε values. Rows with a high reference number in Table 7(a)–(c)



(a) Difficulty OFV versus Budget

(b) Threshold OFV versus Budget

Fig. 4. Range of each OFV on the Pareto frontier for a variable budget with data set ℓ1. The (·····) represents a budget of 10% of the all control costs. Subfigure (a) represents the range of the optimal difficulty OFVs considering all possible ε values for different budgets. The line (—+—) indicates the maximum difficulty OFV across all ε values for a given budget; (- - -) represents the minimum. Subfigure (b) represents the range of the optimal threshold OFVs considering all possible ε values for different budgets. The line (—+—) indicates the maximum threshold OFV across all ε values for a given budget; (- - -) represents the minimum.

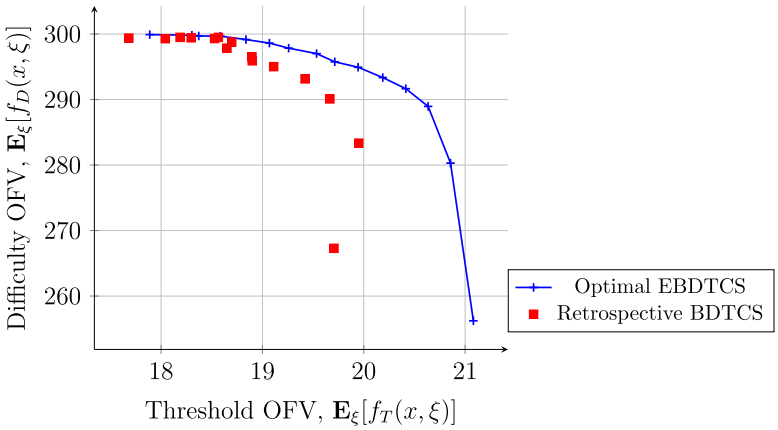


Fig. 5. A retrospective comparison of optimal BDTCS solutions and the EBDTCS Pareto frontier when the is uncertainty is considered. Solutions are for data set *m*2, $|\Omega| = 25$. Budget was set to 10% of total control costs.

Table 6
Sum of EBDTCS solution times for 15 instances with ε ranging from $\varepsilon_{\min}$ to $\varepsilon_{\max}$ in equal increments estimating the Pareto frontier for each data set and scenario size. A maximum solution time of 7200 s was set for each instance solution. Budget was set to 10% of total control costs.

Data Set	Total solution time (s)		
	$\|\Omega\| = 5$	$\|\Omega\| = 25$	$\|\Omega\| = 100$
*s*1	0.3	1.6	12.4
*s*2	2.3	33.1	132.2
*m*1	19.5	1,042.3	2,646.1*
*m*2	11.7	91.8	1,083.9
ℓ1	73.6	433.2*	11,171.5
ℓ2	11.0	277.7	3,999.9

* Indicates only 14 solutions; solution time $\geq$ 7200 s when $\varepsilon = \varepsilon_{\max}$.

correspond to BDTCS instances with relatively high ε values. We observe that solutions to BDTCS and EBDTCS instances with low reference numbers select a larger fraction of the same controls than solutions with high reference numbers and have a smaller value of the stochastic solution (VSS). Across all data sets and $|\Omega|$ we find that 0.982 of the EBDTCS controls are also selected in the BDTCS problem in the lowest reference number solutions. This fraction decreases to 0.677 in the highest reference number solutions, on average.

This computational study investigates and highlights the benefit of the stochastic solution compared the deterministic solution and quantifies the differences between the BDTCS and EBDTCS solutions. In cases where high importance is placed on securing attacks to an acceptable level of risk, organizations can and should use the stochastic model to make better decisions when they can identify and quantify the gaps in their knowledge.

4. Discussion and conclusion

In this paper, we present integer and stochastic programming models to select an optimal portfolio of security controls to reduce the risk of an information and communication technology supply chain. We do so by presenting a novel bi-objective programming model for selecting mitigating controls to simultaneously maximize the number of attacks past a risk-threshold and maximize the total difficulty increase among all potential attacks. We introduce a stochastic programming model variant to capture uncertainty in a control's effectiveness. We investigate the deterministic and stochastic model solutions. We find that under most budgets, it is necessary to set meaningful thresholds to match the control portfolio to an organization's risk tolerance. We also find the stochasticity of the problem should be considered in cases where high importance is placed on securing attacks past a threshold; otherwise, the deterministic model suffices to provide near-optimal solutions to the stochastic problem.

The models we present in this paper provide a structured approach to composing a portfolio of security controls that is effective with respect to cost and multiple objectives related to risk reduction. The analysis of the computational examples indicates that finding the right balance across multiple objectives is non-trivial, demonstrating the importance of including multiple criteria and the value of the optimization-based approach over approaches used in practice that rely on simple heuristics [44] and ranking based on a cost/benefit analysis [12,42]. Moreover, the analysis indicates that uncertainty regarding mitigation effectiveness is important for guiding portfolio composition decisions and should be included in the risk management processes if applicable.

This approach can be used as one method in a suite of supply chain risk management tools to aid managerial decision-making [16]. Implementing the approach presented in this paper may create several challenges for organizations; we highlight two. First, organizations must develop or have a framework to identify vulnerabilities, attack paths, security controls, difficulty measures, and weights using, for example, "red-teams," subject matter expert solicitations, structured risk assessments, or a combination of these methods. Second, many organizations currently lack visibility into aspects of their supply chains leading to missing or inaccurate data [8,16]; this a key reason to employ the proposed stochastic programming approach. Organizations have also worked with suppliers to increase visibility of their supply chain, which could remove the uncertainties and allow for use of the deterministic model [2,31,32].

Information technology supply chain risk is complex and requires significant mathematical modeling to reduce that risk. As the current information and communication technology literature is limited, there is opportunity for future investigation into additional models for reduction of information and communication technology supply chain risk. Some of these opportunities are as follows. First, non-linear optimization can capture controls which increase path difficulties and decrease the consequence of attacks. Second, multi-period models can aid organizations with yearly budgets and controls costs which may be able to be split among budget periods. Third, game-theoretic models can be used to defend against strategic, adversarial attacks and to consider coordination among multiple attackers.

CRediT authorship contribution statement

Adam Schmidt: Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing, Data curation. **Laura A. Albert:** Conceptualization, Validation, Writing - original draft, Writing - review & editing, Supervision. **Kaiyue Zheng:**

Table 7
Comparison of controls selected in comparable BDTCS and EBDTCS solutions. The number of shared controls is the number of controls found in both comparable BDTCS and EBDTCS solutions. The VSS is the value of the stochastic solution for each pair of comparable solutions.

	BDTCS				EBDTCS					
Ref. Num.	ε	Retro- $f_T(\cdot)$	Retro- $f_D(\cdot)$	Num. controls selected	ε	$\mathbf{E}_\xi[f_T(x,\xi)]$	$\mathbf{E}_\xi[f_D(x,\xi)]$	Num. controls selected	Num. shared controls	VSS
(a) Data set s2 with $\|\Omega\| = 100$										
1	3.89	3.34	80.00	27	3.34	3.34	80.00	27	27	0.00
2	4.03	3.49	78.60	28	3.49	3.49	78.60	28	28	0.00
3	4.18	3.49	78.60	28	3.49	3.49	78.60	28	27	0.00
4	4.32	3.49	78.60	28	3.49	3.49	78.60	28	28	0.00
5	4.47	3.49	78.60	28	3.49	3.49	78.60	28	28	0.00
6	4.62	3.49	78.60	28	3.49	3.49	78.60	28	27	0.00
7	4.76	3.68	74.86	24	3.68	3.70	78.14	23	21	3.28
8	4.91	3.68	74.86	24	3.68	3.70	78.14	23	21	3.28
9	5.05	3.68	74.86	24	3.68	3.70	78.14	23	21	3.28
10	5.20	3.70	73.65	27	3.70	3.72	78.11	25	23	4.46
11	5.34	4.03	73.58	23	4.03	4.07	75.91	26	20	2.33
12	5.49	3.91	65.31	22	3.91	3.95	76.35	24	18	11.04
13	5.63	4.21	64.97	21	4.21	4.23	70.92	25	17	5.95
14	5.78	4.21	64.97	21	4.21	4.23	70.92	25	17	5.95
15	5.92	4.36	62.92	17	4.36	4.39	66.15	22	14	3.23
(b) Data set m2 with $\|\Omega\| = 100$										
1	19.46	17.27	299.17	104	17.27	17.45	299.20	100	99	0.03
2	19.82	17.54	299.16	104	17.54	17.54	299.16	104	104	0.00
3	20.17	17.83	298.94	103	17.83	17.84	298.99	102	100	0.05
4	20.53	17.93	298.81	103	17.93	18.08	298.95	103	100	0.14
5	20.89	18.02	298.78	103	18.02	18.08	298.95	103	99	0.17
6	21.24	18.10	298.69	104	18.10	18.21	298.79	103	101	0.10
7	21.60	18.26	298.15	102	18.26	18.30	298.59	102	100	0.44
8	21.96	18.36	297.37	103	18.36	18.38	298.28	106	101	0.91
9	22.31	18.69	296.22	102	18.69	18.70	297.58	103	94	1.36
10	22.67	18.67	295.56	102	18.67	18.70	297.58	103	96	2.02
11	23.03	18.90	294.93	103	18.90	18.92	297.01	101	95	2.08
12	23.39	19.02	292.57	101	19.02	19.03	296.56	98	91	3.99
13	23.74	19.32	289.48	99	19.32	19.33	295.30	102	91	5.82
14	24.10	19.47	282.26	99	19.47	19.48	294.74	100	88	12.48
15	24.46	19.16	266.51	100	19.16	19.16	295.65	100	80	29.14
(c) Data set ℓ1 with $\|\Omega\| = 100$										
1	61.45	54.76	884.00	227	54.76	54.80	885.12	226	219	1.12
2	62.65	55.81	884.21	227	55.81	55.88	884.97	226	220	0.76
3	63.86	56.46	884.04	227	56.46	56.47	884.55	226	219	0.51
4	65.06	57.41	883.10	229	57.41	57.42	883.32	227	223	0.22
5	66.27	58.11	882.05	231	58.11	58.11	882.22	231	227	0.17
6	67.47	59.05	879.81	230	59.05	59.05	880.41	231	225	0.60
7	68.68	59.69	876.44	227	59.69	59.70	878.08	230	219	1.64
8	69.88	60.38	872.32	227	60.38	60.39	874.73	227	216	2.41
9	71.09	61.06	866.89	223	61.06	61.06	870.03	226	218	3.14
10	72.29	61.03	860.00	219	61.03	61.04	870.22	227	211	10.22
11	73.50	61.37	851.76	217	61.37	61.37	867.83	224	207	16.07
12	74.70	61.26	840.40	215	61.26	61.26	868.77	225	201	28.37
13	75.91	61.61	823.88	207	61.61	61.62	865.68	221	189	41.80
14	77.12	61.66	798.13	204	61.66	61.66	865.31	220	180	67.18
15	78.32	60.00	718.87	183	60.00	60.01	876.68	229	138	157.81

Conceptualization, Methodology, Validation, Writing - original draft.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

Acknowledgments

This work was funded by the National Science Foundation Awards 1422768 and 1912166. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation. This methodology in this paper is motivated by conversations with federal decision makers at Sandia National Laboratory about planning applications for mitigating risks in Federal IT infrastructure. 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, Dr. Forough Enayaty-Ahangar for reviewing an earlier version of this manuscript, and the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ress.2020.107093.

References

[1] Boyens J, Paulsen C, Bartol N, Shankles SA, Moorthy R. Notional Supply Chain Risk Management Practices for Federal Information Systems. Tech. Rep.. Gaithersburg, MD: National Institute of Standards and Technology; 2012. https://doi.org/10.6028/NIST.IR.7622.

[2] Boyens J, Paulsen C, Bartol N, Winkler K, Gimbi J. Key Practices in Cyber Supply Chain Risk Management: Observations from Industry. Tech. Rep.. Gaithersburg, MD: National Institute of Standards and Technology; 2020.

[3] Boyens J, Paulsen C, Moorthy R, Bartol N. Supply chain risk management practices for federal information systems and organizations. Special Publication. Gaithersburg, MD: National Institute of Standards and Technology; 2015.

[4] Boyens JM, Paulsen C, Bartol N, Winkler K, Gimbi J. Case Studies in Cyber Supply Chain Risk Management: Summary of Findings and Recommendations. Tech. Rep.. Gaithersburg, MD: National Institute of Standards and Technology; 2020.

[5] DiMase D, Collier ZA, Carlson J, Gray Jr RB, Linkov I. Traceability and risk analysis strategies for addressing counterfeit electronics in supply chains for complex systems. Risk Anal 2016;36(10):1834–43.

[6] Donkers B, Melenberg B, Van Soest A. Estimating risk attitudes using lotteries: alarge sample approach. J Risk Uncertain 2001;22(2):165–95. https://doi.org/10.1023/A:1011109625844.

[7] Duane M, Brandenburg R, Gruber M. When the Going Gets Tough, the Tough Get Going: Overcoming the Cyber Risk Appetite Challenge. Tech. Rep.. Oliver Wyman; 2018.

[8] Edwards NJ, Kao GK, Hamlet JR, Bailon J. Supply Chain Decision Analytics: Application and Case Study for Critical Infrastructure Security. Tech. Rep.. Albuquerque, NM, USA 87123: Sandia National Laboratories; 2015.

[9] Enayaty-Ahangar F, Albert LA, DuBois E. A survey of optimization models and methods for cyberinfrastructure. IISE Trans 2020. https://doi.org/10.1080/24725854.2020.1781306. Under Review

[10] Fei YJN, Jiang W. A quantifiable attack-defense trees model for apt attack. 2018 IEEE 3rd advanced information technology, electronic and automation control conference (IAEAC). 2018. p. 2303–6.

[11] Ganin AA, Quach P, Panwar M, Collier ZA, Keisler JM, Marchese D, et al. Multicriteria decision framework for cybersecurity risk assessment and management. Risk Anal 2020;40(183–198):1834–43.

[12] Hubbard DW, Seiersen R. How to measure anything in cybersecurity risk. Hoboken, New Jersey: John Wiley & Sons; 2016.

[13] Insua DR, Vieira AC, Rubio JA, Pieters W, Labunets K, Rasines DG. An adversarial risk analysis framework for cybersecurity. Risk Anal 2019. https://doi.org/10.1111/risa.13331.

[14] Jha S, Sheyner O, Wing JM. Two formal analyses of attack graphs. Proceedings 15th IEEE computer security foundations workshop. CSFW-15. 2002.

[15] Kao G, Lin H, Eames B, Haas J, Fisher A, Michalski J, et al. Supply Chain Lifecycle Decision Analytics. Tech. Rep.. Albuquerque, NM, USA 87123: Sandia National Laboratories; 2014.

[16] Kao GK, Hamlet J, Helinski R, Shakamuri M, Lin HW, Michalski JT. Supply Chain Security Decision Analytics: Macro Analysis. Tech. Rep.. Albuquerque, NM, USA 87123: Sandia National Lab; 2015.

[17] Kleywegt AJ, Shapiro A, Homem-de Mello T. The sample average approximation method for stochastic discrete optimization. SIAM J Optim 2002;12(2):479–502.

[18] Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: a tutorial. Reliab Eng Syst Saf 2006;91(9):992–1007. https://doi.org/10.1016/j.ress.2005.11.018.

[19] Kordy B, Mauw S, Radomirović S, Schweitzer P. Foundations of attack–defense trees. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 80–95.

[20] Kordy B., Widel W.. How well can I secure my system? Research Institute of Computer Science and Random Systems http://people.irisa.fr/Barbara.Kordy/papers/iFM17.pdf; 2017. [Online; accessed 14-Jan-2019].

[21] Kordy B, Wideł W. On quantitative analysis of attack–defense trees with repeated labels. International Conference on Principles of Security and Trust. Springer, Cham; 2018. p. 325–46.

[22] Laumanns M, Thiele L, Zitzler E. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. Eur J Oper Res 2006;169(3):932–42. https://doi.org/10.1016/j.ejor.2004.08.029.

[23] Letchford J, Vorobeychik Y. Optimal interdiction of attack plans. Proceedings of the 12th international conference on autonomous agents and multiagent systems, Saint Paul, MN. 2013.

[24] Li X, Zhou C, Tian Y-C, Qin Y. A dynamic decision-making approach for intrusion response in industrial control systems. IEEE Trans Ind Inf 2019;15(5):2544–54. https://doi.org/10.1109/TII.2018.2866445.

[25] Lu T, Yao P, Guo X, Zhang X, Yang H. A systematic study for ICT supply chain security. J Logist Inform Serv Sci 2015;2(1):28–41.

[26] Marler RT, Arora JS. Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 2004;26(6):369–95. https://doi.org/10.1007/s00158-003-0368-6.

[27] Mauw S, Oostdijk M. Foundations of attack trees. Proceedings of the international conference on information security and cryptology. Springer; 2005. p. 186–98.

[28] Mavrotas G. Effective implementation of the epsilon-constraint method in multi-objective mathematical programming problems. Appl Math Comput 2009;213(2):455–65. https://doi.org/10.1016/j.amc.2009.03.037.

[29] Mavrotas G, Florios K. An improved version of the augmented epsilon-constraint method (AUGMECON2) for finding the exact Pareto set in multi-objective integer programming problems. Appl Math Comput 2013;219(18):9652–69. https://doi.org/10.1016/j.amc.2013.03.002.

[30] McCrory FM, Kao GK, Blair DS. Supply Chain Risk Management: The Challenge in a Digital World. Tech. Rep.. Albuquerque, NM, USA 87123: Sandia National Laboratories; 2015.

[31] Microsoft Corporation. Securing the Supply Chain with Risk-Based Assessments. Tech. Rep.. One Microsoft Way Redmond, WA 98052-6399 USA: Microsoft; 2017.

[32] Microsoft Corporation. Guarding against supply chain attacks–Part 1: The big picture. https://www.microsoft.com/security/blog/2019/10/16/guarding-against-supply-chain-attacks-part-1-big-picture/; 2019. [Online: accessed 23-Apr-2020].

[33] Nandi AK, Medal HR, Vadlamani S. Interdicting attack graphs to protect organizations from cyber attacks: a bi-level defender–attacker model. Comput Oper Res 2016;75:118–31.

[34] National Institute of Standards and Technology. Guide for Conducting Risk Assessments. Tech. Rep.. 2012. Gaithersburg, MD 20899-8930

[35] Phillips C, Swiler LP. A graph-based system for network-vulnerability analysis. Proceedings of the 1998 workshop on new security paradigms, NSPW '98. New York, NY, USA: ACM; 1998. https://doi.org/10.1145/310889.310919http://doi.acm.org/10.1145/310889.310919.

[36] President's Commission on Enhancing National Cybersecurity. Report on Securing and Growing the Digital Economy. Tech. Rep.. 2016. Washington, D.C.

[37] Redondo A., Torres-Barrán A., Insua D.R., Domingo J.. Assessing Supply Chain Cyber Risks2019;*Preprint*; URL http://arxiv.org/abs/1911.11652.

[38] Rong A, Figueira JR. Dynamic programming algorithms for the bi-objective integer knapsack problem. Eur J Oper Res 2014;236(1):85–99. https://doi.org/10.1016/j.ejor.2013.11.032.

[39] Scala NM, Reilly AC, Goethals PL, Cukier M. Risk and the five hard problems of cybersecurity. Risk Anal 2019;39(10):2119–26. https://doi.org/10.1111/risa.13309.

[40] Shackleford D. Combatting Cyber Risks in the Supply Chain. Tech. Rep.. Bethesda, MD, USA: SANS; 2015.

[41] Sheyner O, Haines J, Jha S, Lippmann R, Wing JM. Automated generation and analysis of attack graphs. Proceedings 2002 IEEE symposium on security and privacy. 2002. p. 273–84. https://doi.org/10.1109/SECPRI.2002.1004377.

[42] Storch T. Toward a Trusted Supply Chain: A Risk Based Approach to Managing Software Integrity. Tech. Rep.. Redmond, WA, USA: Microsoft; 2017.

[43] Tang L, Wang H, Li G, Xu F. Adaptive heuristic search algorithm for discrete variables based multi-objective optimization. Struct Multidiscip Optim 2013;48(4):821–36. https://doi.org/10.1007/s00158-013-0932-7.

[44] The White House. Securing the Information and Communications Technology and Services Supply Chain, Office of the Press Secretary, Washington, D.C.; 2019. Executive Order No. 13873.

[45] Ulungu EL, Teghem J. Multi-objective combinatorial optimization problems: a survey. J Multi-Criteria Decis Anal 1994;3(2):83–104.

[46] Vigo R, Nielson F, Nielson HR. Automated generation of attack trees. 2014 IEEE 27th computer security foundations symposium. 2014. p. 337–50. https://doi.org/10.1109/CSF.2014.31.

[47] Wyss GD, Hinton JP, Dunphy-Guzman K, Clem J, Darby J, Silva C, et al. Risk-based cost-benefit analysis for security assessment problems. Vulnerability Uncertain Risk 2011:738–45. https://doi.org/10.1061/41170(400)90.

[48] Zhang W, Reimann M. A simple augmented epsilon-constraint method for multi-objective mathematical integer programming problems. Eur J Oper Res 2014;234(1):15–24. https://doi.org/10.1016/j.ejor.2013.09.001.

[49] Zheng K, Albert LA. Interdiction models for delaying adversarial attacks against critical information technology infrastructure. Naval Res Logist (NRL) 2019;66(5):411–29. https://doi.org/10.1002/nav.21859.

[50] Zheng K, Albert LA. A robust approach for mitigating risks in cyber supply chains. Risk Anal 2019;39(9):2076–92. https://doi.org/10.1111/risa.13269.

[51] Zheng K, Albert LA, Luedtke JR, Towle E. A budgeted maximum multiple coverage model for cybersecurity planning and management. IISE Trans 2019;51(12):1303–17. https://doi.org/10.1080/24725854.2019.1584832.