Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Shortest Path Interdiction Problem with Randomized Interdiction Strategies: Complexity and Algorithms

Tim Holzmann, J. Cole Smith

informs
http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Vol. 69, No. 1, January–February 2021, pp. 82–99
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Contextual Areas

The Shortest Path Interdiction Problem with Randomized Interdiction Strategies: Complexity and Algorithms

Tim Holzmann,[a] J. Cole Smith[b]

[a] Department of Operational Sciences, Air Force Institute of Technology, Wright Patterson AFB, Dayton, Ohio 45433; [b] Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, New York 13244

Contact: timothy.holzmann@afit.edu, https://orcid.org/0000-0002-0073-0682 (TH); colesmit@syr.edu, https://orcid.org/0000-0001-5106-6964 (JCS)

Received: June 6, 2019
Accepted: April 1, 2020
Published Online in Articles in Advance: December 25, 2020

Subject Classifications: 90C30: mathematical programming: nonlinear programming; 60W25: algorithms in computer science: approximation algorithms; 91A43: game theory: games involving graphs
Area of Review: Military and Homeland Security

https://doi.org/10.1287/opre.2020.2023

Abstract. Shortest-path interdiction problems involve a leader and a follower playing a zero-sum game over a directed network. The leader interdicts a set of arcs, and arc costs increase as a function of the number of times they are interdicted. The follower observes the leader's actions and selects a shortest path in response. The leader's optimal interdiction strategy maximizes the follower's minimum-cost path. In classic formulations of these problems, the leader's interdiction actions are deterministic. In this paper the leader selects a policy of randomized interdiction actions, and the follower only knows the probability of where interdictions are deployed on the network. The follower identifies a path having the minimum expected cost, whereas the leader seeks to maximize the follower's objective. When the arc costs are affine functions of the number of times they are interdicted, this problem is solvable by linear programming. However, when the cost functions are convex or when they are concave, we show that the expected costs are Schur concave or convex, respectively. This property allows us to prove that the problem becomes strongly NP-hard in the nonaffine case. We also propose several algorithms for this problem. Some are for special cases, and one is a general algorithm. We examine the efficacy of our algorithms on a test bed of randomly generated instances by comparing our algorithms to a standard solver.

Funding: This work was supported by the Office of Naval Research [Grant N00014-17-1-2421].

Keyword: networks/graphs • nonlinear programming • military • Schur convexity

1. Introduction and Motivation

Classic shortest-path interdiction problems (SPIPs; Israeli and Wood 2002) have a leading actor that selects a deterministic interdiction strategy and a following actor that observes the leader's strategy and selects a shortest path in response. The optimal interdiction strategy maximizes the follower's minimum-cost path. By contrast, we examine the SPIP variant in which the leader can select a *randomized* interdiction strategy. The follower cannot observe the true interdiction locations before selecting a shortest path, but does know the probability distribution that governs the leader's interdiction strategy. The follower's goal is to select a path having minimum *expected* cost, given the leader's (stochastic) interdiction strategy.

Our variation of the SPIP is motivated by real-life applications where randomized interdiction strategies can be gainfully employed. This situation arises frequently in security applications, for which true interdiction resources are dispersed among decoys. For instance, detection devices such as remote cameras may be placed at border crossings, only some of which are monitored due to personnel limits (see Kutz and Cooney 2007). The evader may reasonably observe the camera locations and obtain data to ascertain evasion probabilities in the presence of cameras, but the evader cannot know if the cameras are actually being monitored. Here the interdictor can randomize its monitoring activity and hide the day-to-day assignment of interdiction resources (i.e., agents monitoring the cameras) from the evader. (Note the equivalence between maximizing a minimum-cost path and minimizing a maximum-evasion-probability path for an evader (see, e.g., Pan and Morton 2008).)

In addition to these randomized strategies, we also assume that the interdictor may deploy multiple interdiction actions at a single location. By returning to the case of border security, the follower's cost may increase as a concave function of interdiction actions if redundant monitors at a location yield diminishing returns in detection probability. Alternatively, the cost function may be convex if monitoring agents work in teams and are more effective as a group than they are as individuals. Lunday and Sherali (2012) provide a thorough discussion of such synergies among interdiction resources, leading to nonlinear interdiction models such as the one we study here.

In the remainder of the introduction we present our problem formulation and give an example in Section 1.1. Section 1.2 reviews relevant literature, and Section 1.3 gives notational preliminaries and describes previous results regarding the complexity of our problem. In Section 1.4 we describe our contributions and provide an overview of the paper.

1.1. Problem Description and Formulation

In a preliminary work, Holzmann and Smith (2019) introduce the shortest path interdiction problem with randomized strategies (SPIP-RS) as a max-min Stackelberg game. The leader and follower play over a directed network given by $G = (\mathcal{N}, \mathcal{A})$, with node set $\mathcal{N} = \{1, \ldots, n\}$ and arc set $\mathcal{A} = \{a_1, \ldots, a_m\}$. The leader performs exactly $b \in \mathbb{Z}_+$ interdiction actions on the arcs, possibly interdicting some arcs multiple times. The cost for the follower to use arc $a \in \mathcal{A}$, if that arc has been interdicted τ times, is given by $c_a(\tau)$. Defining $\mathcal{H} = \{1, \ldots, b\}$, we assume that $c_a : \{0\} \cup \mathcal{H} \to \mathbb{R}_+$ is nondecreasing.

In the SPIP-RS the leader adopts a randomized interdiction strategy. We define Bernoulli random variables χ_{ha} that equal 1 if interdiction h is deployed on arc a and 0 otherwise, $\forall h \in \mathcal{H}$ and $a \in \mathcal{A}$. Since each interdiction action is deployed on exactly one arc, these random variables are not independent. The leader's decision space is $\mathfrak{X} = \{X \in [0,1]^{b \times m} : \sum_{a \in \mathcal{A}} x_{ha} = 1, \forall h \in \mathcal{H}\}$, where the components of X are chosen so that $\mathcal{P}(\chi_{ha} = 1) = x_{ha}$. The follower observes X, knows the cost functions for each arc, and seeks a path from node $s \in \mathcal{N}$ to node $t \in \mathcal{N}$ with the minimum expected cost.

Denote the columns of decision matrix X by $\mathbf{x}_a$ for $a \in \mathcal{A}$. Throughout, we refer to $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$ as the expected cost of arc $a \in \mathcal{A}$, given vector $\mathbf{x}_a$, where $\tau = \sum_{h \in \mathcal{H}} \chi_{ha}$ acts as a random variable corresponding to the number of times arc a is interdicted. The leader selects a stochastic interdiction strategy, $X \in \mathfrak{X}$, that maximizes the follower's minimum expected cost. We formulate the following mathematical optimization model for this problem by taking the dual of the follower's shortest-path problem and associating dual variables d_i with flow-balance constraints for each $i \in \mathcal{N}$ (see Fulkerson and Harding 1977):

$$\max_{(X,\mathbf{d}) \in (\mathfrak{X}, [0,\infty)^n)} \quad d_s \tag{1}$$

$$\text{s.t.} \quad d_i - d_j \le \mathrm{E}\big(c_{ij}(\tau)|\mathbf{x}_{ij}\big) \quad \forall (i,j) \in \mathcal{A}, \tag{2}$$

$$d_t = 0. \tag{3}$$

Note that d_i represents the shortest expected distance from node i to node t.

Example 1. Consider the network displayed in Figure 1 with cost functions given in Table 1. Our example consists of two cases, both using a budget of $b = 2$ interdictions. In the first case we adopt an optimal deterministic interdiction strategy. In the second case we randomize the interdiction strategy, which yields a larger objective for the SPIP-RS than the Case 1 solution.

Case 1. Strategy X^1 is an optimal deterministic interdiction strategy. Arcs (s, a) and (s, t) are each interdicted once with probability one (i.e., $1 = \mathcal{P}(\chi_{1,(s,t)} = 1 |\mathbf{x}^1_{(s,t)}) = \mathcal{P}(\chi_{2,(s,a)} = 1|\mathbf{x}^1_{(s,a)})$), giving the expected costs in Table 2. The follower opts to use arc $s \to t$ at an expected cost of 4.8. We have

$$\begin{array}{cc} & \begin{array}{ccccc} (s,a) & (s,b) & (s,t) & (a,t) & (b,t) \end{array} \\ X^1 = & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{array}$$

Case 2. The stochastic interdiction strategy X^2, given below, applies each interdiction to each arc with some positive probability. To clarify the model computations, we compute the expected cost for arc (s, a) below. We have

$$\begin{array}{cc} & \begin{array}{ccccc} (s,a) & (s,b) & (s,t) & (a,t) & (b,t) \end{array} \\ X^2 = & \begin{bmatrix} 0.3 & 0.05 & 0.5 & 0.1 & 0.05 \\ 0.35 & 0.05 & 0.5 & 0.05 & 0.05 \end{bmatrix}, \end{array}$$

$$\mathcal{P}\left(\chi_{1,(s,a)} = 1|\mathbf{x}^2_{(s,a)}\right) = X^2_{1,(s,a)} = 0.3,$$

$$\mathcal{P}\left(\chi_{2,(s,a)} = 1|\mathbf{x}^2_{(s,a)}\right) = X^2_{2,(s,a)} = 0.35$$

$$\Longrightarrow \mathcal{P}\left(\tau = t|\mathbf{x}^2_{(s,a)}\right) = \begin{cases} 45.5\% & t = 0 \\ 44\% & t = 1 \\ 10.5\% & t = 2 \end{cases}$$

$$\begin{aligned} \Longrightarrow E\big(c_{(s,a)}(\tau)|\mathbf{x}_{(s,a)}\big) &= 0.455 c_{(s,a)}(0) \\ &\quad + 0.44 c_{(s,a)}(1) \\ &\quad + 0.105 c_{(s,a)}(2) \\ &= 3.2025. \end{aligned}$$

We continue in this fashion to compute the expected costs for each path through the network, as seen in Table 3. The follower now chooses path $s \to a \to t$ with expected cost 5.4285.

Figure 1. Example 1 Graph

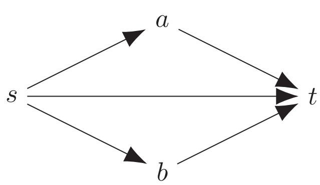


Table 1. Arc Costs for Figure 1

Arc (i,j)	$c_{ij}(0)$	$c_{ij}(1)$	$c_{ij}(2)$
(s,a)	2.5	3.5	5
(s,b)	3.3	4.9	5.7
(s,t)	3.9	4.8	11.1
(a,t)	2	3.5	5.2
(b,t)	2.4	3.2	5.6

Table 2. Expected Path Costs Given X^1

Path	Expected cost
$s \to a \to t$	$3.5 + 2 = 5.5$
$s \to b \to t$	$3.3 + 2.4 = 5.7$
$s \to t$	4.8

Table 3. Expected Path Costs Given X^2

Path	Expected cost
$s \to a \to t$	$= 5.4285$
$s \to b \to t$	$= 6.5228$
$s \to t$	$= 6.15$

1.2. Literature Review

Smith et al. (2013) present an introductory summary on network interdiction problems and review of the relevant literature. Fulkerson and Harding (1977) provide a linear programming (LP) algorithm to solve the SPIP with fractional interdictions allowed, whereas Ball et al. (1989) prove the NP-hardness of SPIP with interdictions constrained to be binary. The seminal work of Wood (1993) provides a general expanded treatment of network interdiction problems, including the SPIP, whereas Israeli and Wood (2002) and Lozano and Smith (2017) examine cutting-plane algorithms for the SPIP.

Multiple works examine stochastic variations of the SPIP. The case where the SPIP interdiction action may or may not be successful is considered in Cormican et al. (1998) and Pan and Morton (2008). Janjarassuk and Linderoth (2008) consider stochastic maximum flow interdiction problems in which interdiction action effectiveness is uncertain. Hemmecke et al. (2003) present a formulation where the network topology is uncertain, and Held et al. (2005) propose solving this variant with a decomposition algorithm. The uncertain topology scenario may be generalized to problems where network costs are uncertain for both parties. In contrast, Bayrak and Bailey (2008), Salmerón (2012), and Sullivan et al. (2014) consider problems where the costs are known only to the follower, and Held and Woodruff (2005), Lunday and Sherali (2012), and Borrero et al. (2015) assume that the costs are known only to the leader. For each of these works, the interdiction strategy is deterministic, unlike the randomized interdiction strategy that we examine for SPIP-RS.

There are comparatively few studies that consider randomized interdiction strategies similar to the SPIP-RS (apart from the brief introduction of SPIP-RS without a complexity discussion or algorithms in Holzmann and Smith (2019)). Washburn and Wood (1995) explore simultaneous games in which the interdictor chooses to monitor a single arc at the same time the evader selects its path. A Nash equilibrium strategy exists for this game, consisting of a mixed-strategy solution. Solutions to the SPIP-RS can be envisioned as mixed-strategy solutions; however, a SPIP-RS interdiction solution X (in general) captures an exponential number of interdiction actions. Furthermore, the game we consider is a Stackelberg (leader-follower) game in which the follower selects a minimum-expected-cost path, avoiding the necessity of stating a randomized path strategy.

Three more notable papers on stochastic interdiction actions focus on the application of deterring fare evasion on public transit networks. Correa et al. (2017) model multiple followers, denoted by their source-sink pairs, (s_i, t_i), for $i = \{1, \ldots, k\}$. The leader seeks to maximize the sum of tolls and evasion fines from all i. They prove the NP-hardness of the leader's problem and provide nonlinear formulations under various assumptions. Finally, they propose a local search algorithm to solve these formulations and provide computational results, comparing the resulting objectives and computational times against an exact branch-and-bound solution. Borndörfer et al. (2013) take a game-theoretic approach, developing linear programs to find Nash and Stackelberg equilibrium solutions. Finally, Bahamondes et al. (2017) consider an adaptive follower that updates interdiction probabilities after crossing each arc, based on whether an interdiction occurred on the arc just travelled, and adjusts the remaining path accordingly. They show that this interdiction problem is NP-hard and solve it using metaheuristics. The SPIP-RS is distinguished from these in that it has a single follower with known source and sink nodes, and we do not assume an affine cost structure.

1.3. Preliminaries

In this paper we examine the complexity of SPIP-RS under different assumptions regarding the cost functions. In many classic SPIP formulations and in the more recent works of, Borndörfer et al. (2013), Bahamondes et al. (2017), and Correa et al. (2017) the cost functions are affine. If this property holds, then $\mathrm{E}(c_a(\tau)|\mathbf{x}_a) = c_a(\sum_{h\in\mathcal{H}} x_{ha})$ for all $a \in \mathcal{A}$, and model (1)–(3) is a linear program. Thus, SPIP-RS is polynomially solvable in this case (as shown by Fulkerson and Harding 1977 and Bahamondes et al. 2017). For the more general case in which the cost functions are nonlinear, we use the following definitions for convexity and concavity of discrete cost functions: c_a is a *discrete-convex function* if $c_a(t) - 2c_a(t+1) + c_a(t+2) \geq 0$ (or ≤ 0) for all $t \in \{0, \ldots, b-2\}$, and we define discrete-concave functions analogously. Since we only use discrete cost functions, for brevity we will simply refer to these properties as convexity or concavity.

An important feature of our problem (which we later use in our proofs) is how it relates to Schur convexity (and Schur concavity; Marshall and Olkin 1979). We summarize that relationship here.

Definition 1 (Majorized). Let $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{b+1}$, and let $v_{[k]}$ denote the kth largest element of a vector $\mathbf{v}$. We say that $\mathbf{x}$ is *majorized* by $\mathbf{x}'$ ($\mathbf{x} \prec \mathbf{x}'$) if $\forall h \in \mathcal{H}$, $\sum_{k=0}^{h} x_{[k]} \leq \sum_{k=0}^{h} x'_{[k]}$, with $\sum_{k=0}^{b} x_{[k]} = \sum_{k=0}^{b} x'_{[k]}$.

Definition 2 (Schur Convexity/Concavity). Let $f : \mathbb{R}^{b+1} \to \mathbb{R}$. We say that f is *Schur convex if* $\mathbf{x} \prec \mathbf{x}' \Rightarrow f(\mathbf{x}) \leq f(\mathbf{x}')$ and *Schur concave* if $\mathbf{x} \prec \mathbf{x}' \Rightarrow f(\mathbf{x}) \geq f(\mathbf{x}')$.

Lemma 1. *Let $a \in \mathcal{A}$. If c_a is concave, then $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$ is Schur convex. If c_a is convex, then $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$ is Schur concave.*

Proof. We prove the first statement, with the second statement holding by a parallel argument. We first introduce the Schur-Ostrowski criterion (Ostrowski 1952), which applies to continuously differentiable and symmetric functions (where the condition is equivalent to requiring that $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$ is unaffected by permuting the elements of $\mathbf{x}_a$, which clearly holds true). The criterion states that c_a is Schur convex on the interval $(0,1)^{b+1}$ if and only if the following condition holds for every $0 \leq j < k \leq b$:

$$\left(\frac{\partial \mathrm{E}(c_a(\tau)|\mathbf{x}_a)}{\partial x_{ja}} - \frac{\partial \mathrm{E}(c_a(\tau)|\mathbf{x}_a)}{\partial x_{ka}}\right)(x_{ja} - x_{ka}) \geq 0. \tag{4}$$

We show that (4) holds for $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$. Let $\mathrm{E}(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = \alpha, \chi_{ka} = \beta)$ represent the value of $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$ conditioned on the outcome that $\chi_{ja} = \alpha$ and $\chi_{ka} = \beta$. Note that

$$\begin{aligned}\mathrm{E}(c_a(\tau)|\mathbf{x}_a) = {} & x_{ja}x_{ka}\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 1\big) \\ & + x_{ja}(1 - x_{ka})\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 0\big) \\ & + \big(1 - x_{ja}\big)x_{ka}\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 1\big) \\ & + \big(1 - x_{ja}\big)(1 - x_{ka})\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 0\big).\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{\partial \mathrm{E}(c_a(\tau)|\mathbf{x}_a)}{\partial x_{ja}} = {} & x_{ka}\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 1\big) \\ & + (1 - x_{ka})\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 0\big) \\ & - x_{ka}\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 1\big) \\ & - (1 - x_{ka})\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 0\big), \text{and} \\ \frac{\partial \mathrm{E}(c_a(\tau)|\mathbf{x}_a)}{\partial x_{ka}} = {} & x_{ja}\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 1\big) \\ & - x_{ja}\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 0\big) \\ & + \big(1 - x_{ja}\big)\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 1\big) \\ & - \big(1 - x_{ja}\big)\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 0\big).\end{aligned}$$

By the symmetry of the interdiction actions, $\mathrm{E}(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 0) = \mathrm{E}(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 1)$, and so the left-hand side of (4) becomes

$$\begin{aligned}&\Big(\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 1\big) - 2\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 1, \chi_{ka} = 0\big) \\ &\quad + \mathrm{E}\big(c_a(\tau)|\mathbf{x}_a, \chi_{ja} = 0, \chi_{ka} = 0\big)\Big)\big(x_{ka} - x_{ja}\big)\big(x_{ja} - x_{ka}\big).\end{aligned} \tag{5}$$

When c_a is concave, the first term is nonpositive, whereas the product of the last two terms must be nonpositive. Hence, the Schur-Ostrowski criterion holds, which completes the proof. □

We will apply Schur convexity and concavity principles to establish important structural properties of optimal solutions.

Next, to perform the complexity analysis in this paper, we will examine the decision problem associated with the SPIP-RS.

DSPIP-RS.
Input: Given the following inputs:
- digraph $G = (\mathcal{N}, \mathcal{A})$ with source $s \in \mathcal{N}$ and destination $t \in \mathcal{N}$;
- number of interdictions $b \in \mathbb{Z}_+$;
- discrete, nondecreasing arc cost functions c_a, for each $a \in \mathcal{A}$;
- threshold, $z^* \geq 0$;

Question: Does there exist $(X, \mathbf{d}) \in \mathfrak{X} \times \mathbb{R}^n$ feasible for (2)–(3) with $d_s \geq z^*$?

Holzmann and Smith (2019) present an efficient method for computing expected arc costs using dynamic programming. For completeness, we summarize their approach here.

Let $h \in \{0\} \cup \mathcal{H}$, and define $\Pi^h(\mathbf{x}_a, t) = \mathcal{P}(\sum_{\hat{h}=1}^{h} \chi_{\hat{h}a} = t|\mathbf{x}_a)$ as the probability that arc $a \in \mathcal{A}$ has been interdicted exactly t times among interdictions $\{1, \ldots, h\}$, given $\mathbf{x}_a$. Define $\Pi^0(\mathbf{x}_a, 0) = 1$. Then we have the recursive formula:

$$\Pi^h(\mathbf{x}_a, t) = x_{ha}\Pi^{h-1}(\mathbf{x}_a, t-1) + (1 - x_{ha})\Pi^{h-1}(\mathbf{x}_a, t). \tag{6}$$

Using (6) we can compute $\Pi^h(\mathbf{x}_a, 0), \ldots, \Pi^h(\mathbf{x}_a, b)$, in increasing order of $h = 0, \ldots, b$ with $\mathcal{O}(b^2)$ operations. The final expected cost of arc $a \in \mathcal{A}$ is computed using $\mathcal{O}(b)$ additional operations via the standard expectation formula:

$$\mathrm{E}(c_a(\tau)|\mathbf{x}_a) = \sum_{\tau=0}^{b} c_a(\tau)\Pi^b(\mathbf{x}_a, \tau). \tag{7}$$

Lemma 2. *DSPIP-RS belongs to NP.*

Proof. Given a certificate $(X, \mathbf{d})$ we can compute the expected costs, $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$, $\forall a \in \mathcal{A}$, in polynomial time. Then, a polynomial-time verification step can verify the objective and feasibility of $(X, \mathbf{d})$. □

1.4. Contribution and Overview

The SPIP-RS is a novel problem in the field of interdiction analysis, with existing research mostly focusing on the case of affine cost functions. Building on the preliminary work of Holzmann and Smith (2019), this paper contributes complexity analysis and algorithms for the SPIP-RS when the cost functions are nonlinear. In Section 2 we show that DSPIP-RS is NP-complete when all costs functions are convex, and we provide a spatial branch-and-bound approach to solving the SPIP-RS. In Section 3 we examine the case in which all cost functions are concave. We prove that DSPIP-RS is NP-hard in this case as well, and we present an efficient approximation algorithm. We then provide a sample average approximation approach for the SPIP-RS under general cost functions in Section 4. Section 5 describes the methods and results of our computational study to examine the performance of our algorithms. Finally, we conclude with future research directions in Section 6.

2. The Convex Case

In this section we focus on the convex SPIP-RS, that is, the SPIP-RS in which c_a is convex for all $a \in \mathcal{A}$. We show that the convex SPIP-RS is NP-hard in Section 2.1 and provide a tailored algorithm to solve this problem in Section 2.2.

2.1. Complexity

We begin this section by establishing a property of optimal solutions to the convex SPIP-RS.

Lemma 3. *Let c_a be convex for each $a \in \mathcal{A}$. There exists a vector $\mathbf{t} \in \mathbb{R}^m$ with $\mathbf{t} \geq \mathbf{0}$ and $\sum_{a=1}^{m} t_a = b$ such that*

$$X^*(\mathbf{t}) = \begin{bmatrix} t_1/b & t_2/b & \cdots & t_m/b \\ \vdots & \vdots & \ddots & \vdots \\ t_1/b & t_2/b & \cdots & t_m/b \end{bmatrix} \tag{8}$$

is part of an optimal solution to (1)–(3).

Proof. Consider an optimal solution $(X, \mathbf{d})$, and define $t_a = \sum_{h \in \mathcal{H}} x_{ha}$, $\forall a \in \mathcal{A}$. We show that $(X^*(\mathbf{t}), \mathbf{d}^*)$ is an alternative optimal solution to (1)–(3), where each $d_a^* = \mathrm{E}(c_a(\tau)|\mathbf{x}_a^*(\mathbf{t}))$. Let $a \in \mathcal{A}$. By the convexity of c_a and Lemma 1, we have that $\mathrm{E}(c_a(\tau)|\mathbf{x}_a))$ is Schur concave. Observe that either $\mathbf{x}_a^*(\mathbf{t}) = \mathbf{x}_a$ or $\mathbf{x}_a^*(\mathbf{t}) \prec \mathbf{x}_a$, which implies by Schur concavity that $\mathrm{E}(c_a(\tau)|\mathbf{x}_a^*(\mathbf{t})) \geq \mathrm{E}(c_a(\tau)|\mathbf{x}_a)$. Repeating this over every arc $a \in \mathcal{A}$, (2) implies that $d_a^* \geq d_a$ for all $a \in \mathcal{A}$, and in particular, $d_s^* \geq d_s$. □

Lemma 3 implies that if an optimal expected number of interdictions, t_a, is known for each arc $a \in \mathcal{A}$, then $X^*(\mathbf{t})$ gives an optimal variable assignment for the SPIP-RS, with the expected cost of arc a being $\sum_{\tau=0}^{b} c_a(\tau)\binom{b}{\tau}(t_a/b)^\tau(1 - t_a/b)^{b-\tau}$. Thus, model (1)–(3) simplifies to

$$\max \quad d_s \tag{9}$$

$$\text{s.t.} \quad d_i - d_j \leq \sum_{\tau=0}^{b} c_{ij}(\tau)\binom{b}{\tau}(t_{ij}/b)^\tau(1 - t_{ij}/b)^{b-\tau}, \quad \forall(i,j) \in \mathcal{A}, \tag{10}$$

$$d_t = 0, \tag{11}$$

$$\sum_{(i,j)\in\mathcal{A}} t_{ij} = b, \tag{12}$$

$$\mathbf{t} \geq \mathbf{0}. \tag{13}$$

We can now prove the hardness of DSPIP-RS for the convex case.

Theorem 1. *DSPIP-RS is NP-complete, even when c_a is convex for all $a \in \mathcal{A}$.*

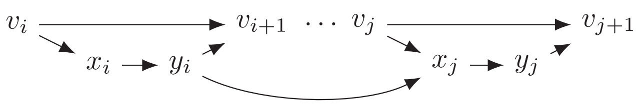
Proof. We employ a reduction from vertex cover to DSPIP-RS modified for (9)–(13), that is, the question becomes "$\exists(\mathbf{t}, \mathbf{d}) \in \mathbb{R}^m \times \mathbb{R}^n$ feasible to (10)–(13) such that $d_s \geq z^*$?" The vertex cover problem is strongly NP-complete (Garey and Johnson 1979) and is stated as follows.

Vertex Cover.
Input: Given the following inputs:
- an undirected graph $G = (V, E)$;
- threshold $r \in \{1, \ldots, |V|\}$;

Question: Does there exist a vertex cover for G of size at most r, that is, $\exists U \subseteq V$ such that $|U| \leq r$ and $U \cap \{i, j\} \neq \emptyset$ for all $(i, j) \in E$?

Given a vertex cover instance, let $|V| = n$. Our graph for the DSPIP-RS instance, $G' = (\mathcal{N}', \mathcal{A}')$, is composed of n "gadgets" composed in a chain, with shortcuts between the gadgets corresponding to edges in E. These gadgets are similar to those used by Bar-Noy et al. (1995) for their complexity analysis of the most vital arcs problem. Gadget i consists of nodes $\{v_i, x_i, y_i, v_{i+1}\}$ and arcs $\{(v_i, x_i), (x_i, y_i), (y_i, v_{i+1}), (v_i, v_{i+1})\}$. Also there are "shortcut" arcs between

Figure 2. Gadgets Used in Proof of Theorem 1



gadgets: $\{(y_i,x_j):(i,j)\in E; i<j\}$. Figure 2 illustrates gadgets for two nodes, i and j. The follower's source node is $s=v_1$, and the destination is $t=v_{n+1}$. We set the number of interdictions

$$b=\left\lfloor \log_{r/r+0.5}\left(\frac{1}{4}\right)\right\rfloor=\left\lfloor\frac{\ln(1/4)}{\ln(r/(r+0.5))}\right\rfloor,$$

and we assign cost functions for $i\in V$ and $(i,j)\in E$ with $i<j$ according to Table 4. Finally, our cost threshold for the DSPIP-RS is $z^*=n$.

As we will show, G' is constructed so that if we have a vertex cover U of cardinality $|U|=k$, then we can deploy each interdiction action on the arcs (x_i,y_i) with probability $1/k$ for each $i\in U$. In this way, any s–t path using a shortcut arc will be too expensive. As a result the optimal path uses only arcs (v_i,v_{i+1}), with total cost of n. Similarly, given that the shortest path cost is n, the shortcut arcs must be expensive enough to guarantee that, for any arc $(i,j)\in E$, at least one of the arcs (x_i,y_i) or (x_j,y_j) is interdicted with probability at least $1/r\sqrt[b]{1/4}$. Selecting $U:=\{i\in V: t_{(x_i,y_i)}\geq b\sqrt[b]{1/4}/r)\}$, we obtain a cover, which, by the functional relationship of b and r, must have cardinality $|U|\leq r$.

Observe that, for any $a\in\mathcal{A}'$, c_a is nondecreasing and convex. It is evident that G' can be constructed in polynomial time. Also, all parameters other than $c_{(x_i,y_i)}(b)=2r^b$ are clearly polynomial in size; moreover, since $b\leq 3r+1$ for $r\geq 0$, the encoding size for these costs are also polynomial as well. We now show that the vertex cover instance has a solution if and only if there exists a feasible solution $(\mathbf{t},\mathbf{d})$ to (9)–(13) having $d_s\geq n$.

Assume that $U\subseteq V$ is a vertex cover for G of size $|U|=k\leq r$. For each $i\in U$, we set $t_{(x_i,y_i)}=b/k$, and we set $t_a=0$ for all other $a\in\mathcal{A}'$ (so that $\mathbf{t}$ satisfies (12) and (13)). Lastly, we set the components of $\mathbf{d}$ as follows:

- $d_{v_i}=n-i+1$ for $i\in\{1,\ldots,n+1\}$;
- $d_{x_i}=n-i+1$ for $i\in\{1,\ldots,n\}\cap U$ and $d_{x_i}=n-i$ for $i\in\{1,\ldots,n\}\setminus U$;
- $d_{y_i}=n-i-1$ for $i\in\{1,\ldots,n\}\cap U$ and $d_{y_i}=n-i$ for $i\in\{1,\ldots,n\}\setminus U$.

We note that (10) is satisfied since $d_t=d_{v_{n+1}}=n-(n+1)+1=0$. Next, we show that the solution $(\mathbf{t},\mathbf{d})$ satisfies (10) by examining the left-hand side (LHS) and right-hand side (RHS) of (10) for every arc in a gadget of $\mathcal{A}'$.

Arc (v_i,v_{i+1}) *for* $i\in\{1,\ldots,n\}$. The LHS of (10) for this arc is $d_{v_i}-d_{v_{i+1}}=(n-i+1)-(n-i)=1$, whereas the RHS is 1 regardless of $\mathbf{t}$. Thus, constraint (10) is satisfied.

Arc (v_i,x_i) *for* $i\in\{1,\ldots,n\}$. Again, $d_{v_i}=n-i+1$. If $i\in U$, then $d_{x_i}=n-i+1$; otherwise, $d_{x_i}=n-i$. The LHS of (10) is therefore either 0 or 1. The RHS is 1 regardless of $\mathbf{t}$, thus satisfying (10).

Arc (x_i,y_i) *for* $i\in\{1,\ldots,n\}$. There are two cases to consider.

Case 1*:* $i\in U$, implying that $t_{(x_i,y_i)}=b/r$. The LHS of (10) is $d_{x_i}-d_{y_i}=(n-i+1)-(n-i-1)=2$, whereas the RHS evaluates to $2(r/k)^b\geq 2$.

Case 2*:* $i\notin U$, implying that $t_{(x_i,y_i)}=0$. The LHS of (10) is $d_{x_i}-d_{y_i}=(n-i)-(n-i)=0$, whereas the RHS is 0 because $t_{(x_i,y_i)}=0$.

In either case, constraint (10) is satisfied.

Arc (y_i,v_{i+1}) *for* $i\in\{1,\ldots,n\}$. We set d_{y_i} to either $n-i-1$ or $n-i$, and $d_{v_{i+1}}=n-i$. The LHS of (10) is either 0 or -1, whereas the RHS is 0 regardless of $\mathbf{t}$, thus satisfying (10).

Arc (y_i,x_j) *for* $(i,j)\in E$. Because $(i,j)\in E$ and U is a cover, then either i or j (or both) belong to U. If both i and j belong to U, then the LHS of (10) is $d_{y_i}-d_{x_j}=(n-i-1)-(n-j+1)=j-i-2$. If only i belongs to U, then the LHS is $(n-i-1)-(n-j)=j-i-1$, and if only j belongs to U, then the LHS is $(n-i)-(n-j+1)=j-i-1$. The maximum value that the LHS can take is $j-i-1$; furthermore, regardless of $\mathbf{t}$, the RHS of (10) for this arc is $j-i-1$, so the constraint is satisfied.

Since we have demonstrated that $(\mathbf{t},\mathbf{d})$ satisfies (10)–(13), the first direction of our proof is completed by showing $d_s=d_{v_1}=n-1+1=n$, implying that the transformed DSPIP-RS instance also has a solution.

Conversely, assume there exists a solution $(\mathbf{t},\mathbf{d})$ feasible for (10)–(13) with $d_s\geq n$. We construct $U=\{i\in V: t_{(x_i,y_i)}\geq(b\sqrt[b]{1/4})/r\}$. To show that U is a cover, assume by contradiction that $(i,j)\in E$ is not covered by U; that is, $t_{(x_i,y_i)},t_{(x_j,y_j)}<(b\sqrt[b]{1/4})/r$. We show that there exists a path from s to t that constrains $d_{v_s}<n$,

Table 4. Costs for Arcs in Figure 2

Arc (a)	$c_a(0)=\cdots=c_a(b-1)$	$c_a(b)$
(v_i,v_{i+1})	1	1
(v_i,x_i)	1	1
(x_i,y_i)	0	$2r^b$
(y_i,v_{i+1})	0	0
(y_i,x_j)	$j-i-1$	$j-i-1$

which leads to the desired contradiction. From repeated applications of (10) we have

$$\begin{aligned}
d_{v_{j+1}} &\le d_{v_{j+2}} + 1 \le d_{v_{j+3}} + 2 \le \cdots \le d_{v_{n+1}} \\
&\quad + n - j = n - j \\
\Rightarrow d_{y_j} &\le d_{v_{j+1}} + \sum_{\tau=0}^{b} c_{(y_j, v_{j+1})}(\tau) \binom{b}{\tau} \left(t_{(y_j, v_{j+1})}/b\right)^{\tau} \\
&= (n-j) + 0 = n - j \\
\Rightarrow d_{x_j} &\le d_{y_j} + \sum_{\tau=0}^{b} c_{(x_j, y_j)}(\tau) \binom{b}{\tau} \left(t_{(x_j, y_j)}/b\right)^{\tau} \\
&= d_{y_j} + c_{(x_j, y_j)}(b) \left(t_{(x_j, y_j)}/b\right)^{b} \\
&< (n-j) + (2r^b) \left(\frac{b\sqrt[b]{1/4}}{br}\right)^{b} = n - j + 1/2 \\
\Rightarrow d_{y_i} &\le d_{x_j} + \sum_{\tau=0}^{b} c_{(y_i, x_j)}(\tau) \binom{b}{\tau} \left(t_{(y_i, x_j)}/b\right)^{\tau} \\
&= d_{x_j} + c_{(y_i, x_j)}(0) \binom{b}{0} (0/b)^{0} \\
&< (n - j + 1/2) + (j - i - 1) = n - i - 1/2 \\
\Rightarrow d_{x_i} &\le d_{y_i} + \sum_{\tau=0}^{b} c_{(x_i, y_i)}(\tau) \binom{b}{\tau} \left(t_{(x_i, y_i)}/b\right)^{\tau} \\
&= d_{y_i} + c_{(x_i, y_i)}(b) \binom{b}{b} \left(t_{(x_i, y_i)}/b\right)^{b} \\
&< (n - i - 1/2) + (2r^b) \left(\frac{b\sqrt[b]{1/4}}{br}\right)^{b} = n - i \\
\Rightarrow d_{v_i} &\le d_{x_i} + \sum_{\tau=0}^{b} c_{(v_i, x_i)}(\tau) \binom{b}{\tau} \left(t_{(v_i, x_i)}/b\right)^{\tau} \\
&= d_{x_i} + c_{(v_i, x_i)}(0) \binom{b}{0} (0/b)^{0} \\
&< (n - i) + 1 = n - i + 1 \\
\Rightarrow d_s = d_{v_1} &\le d_{v_2} + 1 \le \cdots \le d_{v_i} + (i - 1) \\
&< (n - i + 1) + (i - 1) = n,
\end{aligned}$$

which contradicts our assumption that $d_s \ge n$. Thus, we conclude that U is a cover. It remains to show that $|U| \le r$. Note from (11) that $|U|$ is no more than b divided by the threshold interdiction level required to place a node in U, that is,

$$|U| \le \frac{b}{(b\sqrt[b]{1/4})/r}. \tag{14}$$

Moreover, since $|U|$ is an integer, we can take the floor of the RHS of (14), yielding with a slight rearrangement $|U| \le \lfloor r/(\sqrt[b]{1/4}) \rfloor$. Recalling $b = \lfloor \log_{r/r+0.5}(1/4) \rfloor \Rightarrow r/(r+0.5) \le \sqrt[b]{1/4}$, we then obtain

$$|U| \le \left\lfloor \frac{r}{\sqrt[b]{1/4}} \right\rfloor \le \lfloor (r + 0.5) \rfloor = r.$$

Because DSPIP-RS belongs to NP as shown in Lemma 2, this completes the proof. □

2.2. Spatial Branch-and-Bound Algorithm for the Convex Case

Our proposed algorithm for the convex SPIP-RS leverages the following result regarding the RHS of (10).

Proposition 1. *If c_a is convex, then the function $f(t_a) = \sum_{\tau=0}^{b} c_{ij}(\tau)\binom{b}{\tau}(t_a/b)^{\tau}(1 - t_a/b)^{b-\tau}$ from (10) is convex.*

Proof. Let $t_a \in [0, b]$, and for notational ease, let $\hat{t}_a = \frac{t_a}{b}$. Define $\mathbf{x}_a = [\hat{t}_a, \ldots, \hat{t}_a]$. Note that the first two derivatives of $f(\hat{t}_a)$ are

$$\begin{aligned}
f'(\hat{t}_a) &= \sum_{\tau=0}^{b} c_a(\tau) \binom{b}{\tau} \Big((\tau)(\hat{t}_a)^{\tau-1}(1-\hat{t}_a)^{b-\tau} - (b-\tau)(\hat{t}_a)^{\tau} \\
&\qquad \times (1-\hat{t}_a)^{b-\tau-1} \Big) \\
\Rightarrow f''(\hat{t}_a) &= \sum_{\tau=0}^{b} c_a(\tau) \binom{b}{\tau} \Big((\tau)(\tau-1)(\hat{t}_a)^{\tau-2}(1-\hat{t}_a)^{b-\tau} \\
&\qquad - 2(\tau)(b-\tau)(\hat{t}_a)^{\tau-1}(1-\hat{t}_a)^{b-\tau-1} \\
&\qquad + (b-\tau)(b-\tau-1)(\hat{t}_a)^{\tau} \\
&\qquad \times \left(1-\hat{t}_a\right)^{b-\tau-2} \Big).
\end{aligned}$$

Now regrouping the summed form for $f''(\hat{t}_a)$, we have

$$\begin{aligned}
f''(\hat{t}_a) &= \sum_{\tau=0}^{b-2} (\hat{t}_a)^{\tau} (1 - \hat{t}_a)^{b-\tau-2} \\
&\quad \times \left(c_a(\tau+2) \left[(\tau+2)(\tau+1) \binom{b}{\tau+2} \right] \right. \\
&\quad - 2c_a(\tau+1) \left[(\tau+1)(b-\tau-1) \binom{b}{\tau+1} \right] \\
&\quad \left. + c_a(\tau) \left[(b-\tau)(b-\tau-1) \binom{b}{\tau} \right] \right).
\end{aligned}$$

Observing that

$$\begin{aligned}
(\tau+1)(\tau+2)\binom{b}{\tau+2} &= (\tau+1)(b-\tau-1)\binom{b}{\tau+1} \\
&= (b-\tau)(b-\tau-1)\binom{b}{\tau},
\end{aligned}$$

we can simplify $f''(\hat{t}_a)$ as

$$\begin{aligned}
f''(\hat{t}_a) &= \sum_{\tau=0}^{b-2} (c_a(\tau+2) - 2c_a(\tau+1) + c_a(\tau)) \\
&\quad \times \left(\frac{b!}{\tau!(b-\tau-2)!} \right) (\hat{t}_a)^{\tau} (1 - \hat{t}_a)^{b-\tau-2} \ge 0,
\end{aligned}$$

where the inequality holds by the convexity of c_a and the fact that $0 \le \hat{t}_a \le 1$. □

By Proposition 1, formulation (9)–(14) is a nonconvex problem. Appendix A provides a spatial branch-and-bound (Falk and Soland 1969) algorithm to solve the SPIP-RS. It solves over the convex hull of (10)–(14) and then repeatedly branches and tightens each subproblem relaxation to find a solution whose objective value is arbitrarily close to the true optimal objective.

3. The Concave Case

In the previous section, we optimized over aggregate variables $t_1, \ldots, t_m$, where $t_a = \sum_{h=1}^{b} x_{ha}$ represents the expected number of interdiction actions taken on arc a. For the case in which all c_a are convex, finding this optimal set of $\mathbf{t}$-values is NP-hard; however, once this is done, an optimal solution, $X^*(\mathbf{t}) \in \mathfrak{X}$, is easily generated by dividing t_a equally among the b rows. By contrast, when the c_a functions are all concave, the problem of finding a preferred vector $\mathbf{t}$ is polynomially solvable, whereas the problem of mapping a solution $\mathbf{t}$ to an optimal matrix $X \in \mathfrak{X}$ becomes NP-hard. We prove this result in Section 3.1, and then in Section 3.2, we provide a polynomial-time $(1 - 1/\mathrm{e})$-approximation algorithm.

3.1. Complexity Proof

Similar to Section 2.1 we begin by proving a column property of the ideal solution to the concave SPIP-RS.

Lemma 6. *Consider a concave function c_a for some $a \in \mathcal{A}$, and consider an interdiction solution vector $\mathbf{x}_a$ for arc a that contains two fractional components h' and h'' such that $0 < x_{h'a} \le x_{h''a} < 1$. Define $\boldsymbol{\delta} \in \mathbb{R}^b$ to be a vector of all zeros except for $\overline{\delta}_{h'} = -\epsilon$ and $\overline{\delta}_{h''} = \epsilon$, where $\epsilon = \min\{x_{h'a}, 1 - x_{h''a}\}$. Then $\mathrm{E}(c_a(\tau)|\mathbf{x}_a + \boldsymbol{\delta}) \ge \mathrm{E}(c_a(\tau)|\mathbf{x}_a)$.*

Proof. Because c_a is concave, $\mathrm{E}(c_a(\tau)|\mathbf{x}_a)$ is Schur convex. Note that $\mathbf{x}_a + \boldsymbol{\delta} \succ \mathbf{x}_a$. The result then follows by Definition 2. □

Lemma 6 implies that given a vector of preferred column sums, $\mathbf{t}^*$, we would ideally select some $X(\mathbf{t}^*) \in \mathfrak{X}$ such that each column contains at most one fractional component. The challenge would be to allocate fractional components of the columns to rows in a way such that the sum of components in every row equals one (recalling the condition of $\mathfrak{X}$ that requires $\sum_{a \in \mathcal{A}} x_{ha} = 1$, $\forall h \in \mathcal{H}$). However, this allocation is not always feasible, and there may exist a uniquely optimal solution in which some columns have multiple fractional values. This observation motivates our proof of the following theorem.

Theorem 7. *DSPIP-RS is NP-complete, even when c_a is concave for all $a \in \mathcal{A}$.*

Proof. We prove the result with a reduction from 3 partition (3PART), which is NP-complete in the strong sense (Garey and Johnson 1979), and is stated as follows.

> **3PART.**
> **Input:** Given a set of positive integers, $S = \{s_1, \ldots, s_m\}$ where
> - $m = 3q$ for some $q \in \mathbb{Z}_+$;
> - $\sum_{a=1}^{m} s_a = qB$ for some $B \in \mathbb{Z}_+$;
> - $B/4 < s_a < B/2$ for each $s_a \in S$;
>
> **Question:** Does there exist a partitioning of S into three-element sets, $S_1, \ldots, S_q$, such that the sum of elements in each set is B?

Given a 3PART instance we construct a reduction to DSPIP-RS as follows. Our directed graph $G = (\mathcal{N}, \mathcal{A})$ has two vertices, $\mathcal{N} = \{s, t\}$, and m parallel arcs, $\mathcal{A} = \{a_1, \ldots, a_m\}$. A similar transformation to a network having no parallel arcs is easy to obtain by splitting arcs; we use a multigraph here for ease of presentation. The leader's interdiction budget is q. Each arc a has the concave cost function

$$c_a(t) = \begin{cases} 0 & t = 0, \\ B/s_a & t = 1, \ldots, b. \end{cases}$$

Finally, the threshold is $z^* = 1$. Observing that this reduction is polynomial time, we proceed to show that our 3PART instance has a solution if and only if the transformed DSPIP-RS instance has a solution.

Let $S_1, \ldots, S_q$ be a true certificate for 3PART. For each $h = 1, \ldots, q$, examine each $s_a \in S$, and set $x_{ha} = s_a/B$ if $s_a \in S_h$ and $x_{ha} = 0$ otherwise. Then for each $h = \mathcal{H}$, we have $\sum_{a \in \mathcal{A}} x_{ha} = \sum_{s_a \in S_h} s_a/B = B/B = 1$, so that $X \in \mathfrak{X}$. Further, since $S_1, \ldots, S_q$ is a partition, there is a unique $S_h \ni s_a$; hence, each column of X has a unique nonzero component $x_{ha} = s_a/B$. Thus, $\mathrm{E}(c_a(\sigma)|\mathbf{x}_a) = (1 - s_a/B)c_a(0) + (s_a/B)c_a(1) = 1$ for each $a \in \mathcal{A}$, and $(X, [1, 0])$ is a true certificate for DSPIP-RS.

Conversely, assume that the DSPIP-RS instance has a true certificate, $(X, [1, 0])$. We first prove that $\sum_{h \in \mathcal{H}} x_{ha} = s_a/B$ for each $a \in \mathcal{A}$.

The ≥ direction: Assume by contradiction that $\sum_{h \in \mathcal{H}} x_{ha} < s_a/B$ for some arc $a \in \mathcal{A}$. If $\mathbf{x}_a$ has a single positive component, $x_{h^*a} > 0$, then from (9)–(14):

$$\begin{aligned} \mathrm{E}(c_a(\tau)|\mathbf{x}_a) &= c_a(0)\left(1 - x_{h^*a}\right) + c_a(1) x_{h^*a} \\ &< 0(1 - s_a/B) + B/s_a(s_a/B) = 1. \end{aligned} \tag{15}$$

Alternatively, if $\mathbf{x}_a$ has at least two fractional components, then Lemma 4 bounds $\mathrm{E}(c_a(\tau_a)|\mathbf{x}_a)$ by the result in (15), which is less than 1. In either case $\mathrm{E}(c_a(\tau_a)|\mathbf{x}_a) < 1$, contradicting our certificate.

The ≤ direction: This follows directly from the three linear inequalities:

$$\sum_{h \in \mathcal{H}} \sum_{a \in \mathcal{A}} x_{ha} = \sum_{a=1}^{m} s_a/B; \quad \sum_{h \in \mathcal{H}} x_{ha} \ge s_a/B; \text{ and } x_{ha} \ge 0.$$

Next, noting that $\sum_{h\in\mathcal{H}} x_{ha} = s_a/B$ for all arcs $a \in \mathcal{A}$, we have that $\mathrm{E}(c_a(\tau_a)|\mathbf{x}_a) = 1$ if and only if each column $\mathbf{x}_a$ has exactly one positive component. This statement holds by noting that $c_a(0) - 2c_a(1) + c_a(2) < 0$ and by adjusting the proof of Lemma 1 to obtain a strict inequality in (5). Since $\mathrm{E}(c_a(\tau_a)|\mathbf{x}_a) = 1$ by our certificate, we can set $S_h = \{s_a : a \in \mathcal{A}, x_{ha} > 0\}$ for each $h \in \mathcal{H}$ to obtain a partition of S. Moreover, for each $h \in \mathcal{H}$,

$$X \in \mathfrak{X} \Longrightarrow \sum_{a\in\mathcal{A}:x_{ha}>0} s_a/B = 1 \Longrightarrow \sum_{s_a\in S_h} s_a = B.$$

Finally, $|S_h| = 3$ for each $h = 1,\dots,b$ because $B/4 < s_a < B/2$. Thus, $S_1,\dots,S_b$ is indeed a true certificate for 3PART. Thus, the concave DSPIP-RS is NP-hard, and by Lemma 2 the DSPIP-RS is NP-complete. □

3.2. Polynomial-Time Approximation Algorithm for Concave SPIP-RS

We present a polynomial-time approximation algorithm for the concave SPIP-RS. This algorithm first identifies preferred column sums $\mathbf{t}^* \geq \mathbf{0} : \sum_{i=1}^m t_i^* = b$, and then uses that vector to construct a heuristic solution $X^{\mathrm{H}}(\mathbf{t}) \in \mathfrak{X}$. We will show that our heuristic objective is within a factor of at least $1 - 1/e \approx 63.2\%$ of the optimal objective.

We begin our heuristic by finding a set of preferred column sums, $\mathbf{t}^*$, by solving the linear program:

$$\begin{aligned} \max \quad & d_s && (16)\\ \text{s.t.} \quad & d_i - d_j \leq \big(t_{ij} - \tau\big)\big(c_{ij}(\tau+1) - c_{ij}(\tau)\big) + c_{ij}(\tau) && (17)\\ & \qquad \forall (i,j) \in \mathcal{A},\ \tau \in \{0,\dots,b-1\}, && (18)\\ & d_t = 0, && (19)\\ & \sum_{a\in\mathcal{A}} t_a = b, && (20)\\ & t_a \geq 0 \qquad \forall a \in \mathcal{A}. && (21) \end{aligned}$$

Observe that (16) and (18) are equivalent to (1) and (3), respectively, and the right-hand side of constraint (2) is replaced in (17) by the linear functions defining facets of the piecewise-linear function c_{ij}. From the definition of (discrete) concavity, it is straightforward to show that these functions correspond to upper bound values for $\mathrm{E}(c_{ij}(\tau)|\mathbf{x}_a)$. Furthermore, the columnwise interdiction policy defined by $\mathbf{t}$ subject to constraints (19)–(20) relaxes the row-sum constraints of $X \in \mathfrak{X}$ by aggregating them. Thus, (16)–(21) is a relaxation of (1)–(3).

Given $(\mathbf{t}^*, \mathbf{d}^*$ optimal to (16)–(21), we construct a heuristic solution that allocates integer portions of each t_a^* to individual rows and then spreads the remaining fractional portions evenly across the remaining rows. Let $b' = \sum_{a=1}^m \lfloor t_a^* \rfloor$ and $b'' = b - b'$. The integer portion of the heuristic solution generated from $\mathbf{t}^*$ is given by the $b' \times m$ submatrix:

$$X^{\mathrm{Int}}(\mathbf{t}^*) = \begin{array}{r} \\ \text{Interdiction } 1 \\ \vdots \\ \text{Interdiction } \lfloor t_1 \rfloor \\ \text{Interdiction } \lfloor t_1 \rfloor + 1 \\ \vdots \\ \text{Interdiction } \lfloor t_1 \rfloor + \lfloor t_2 \rfloor \\ \vdots \\ \text{Interdiction } b' \end{array} \begin{array}{c} \begin{array}{cccc} \text{Arc } 1 & \text{Arc } 2 & \cdots & \text{Arc } m \end{array} \\ \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \end{array}$$

By letting $\widehat{t_a} = t_a - \lfloor t_a \rfloor$ for each $a \in \mathcal{A}$, the full heuristic solution is given by

$$X^{\mathrm{H}}(\mathbf{t}^*) = \begin{array}{r} 1 \\ \vdots \\ b' \\ b'+1 \\ \vdots \\ b \end{array} \left[\begin{array}{ccc} & X^{\mathrm{Int}}(\mathbf{t}^*) & \\ \hline \widehat{t_1^*}/b'' & \cdots & \widehat{t_m^*}/b'' \\ \vdots & \ddots & \vdots \\ \widehat{t_1^*}/b'' & \cdots & \widehat{t_m^*}/b'' \end{array} \right].$$

The complexity of computing $X^{\mathrm{H}}(\mathbf{t}^*)$ is proportional to the complexity of solving the linear program (16)–(21), which is polynomial in the size of the SPIP-RS instance.

To establish the desired approximation result, we now define two more solutions:

$$X^{\mathrm{LB}}(\mathbf{t}^*) = \begin{array}{r} 1 \\ \vdots \\ b' \\ b'+1 \\ \vdots \\ b \end{array} \left[\begin{array}{c} X^{\mathrm{Int}}(\mathbf{t}^*) \\ \hline \mathbf{0} \end{array} \right], \quad \text{and}$$

$$X^{\mathrm{UB}}(\mathbf{t}^*) = \begin{array}{c} 1 \\ \vdots \\ b' \\ b'+1 \\ b'+2 \\ \vdots \\ b \end{array} \left[\begin{array}{c} X^{\mathrm{Int}}(\mathbf{t}^*) \\ \hline \widehat{\mathbf{t}^*}^{T} \\ \hline \mathbf{0} \end{array}\right].$$

Although neither $X^{\mathrm{LB}}(\mathbf{t})$ nor $X^{\mathrm{UB}}(\mathbf{t})$ necessarily belong to $\mathfrak{X}$, since they need not satisfy the row sum constraints, we use their corresponding objectives to bound the optimal objective. Defining $z(X)$ to be the objective function value of (1)–(3) given a fixed interdiction matrix X, we get the following objective bounds.

Proposition 2. *Let X^* be an optimal solution to (1)–(3) for some graph G with concave cost functions c_a $\forall a \in A$, and let $\mathbf{t}^*$ be an optimal solution to (16)–(21). Then $z(X^{\mathrm{LB}}(\mathbf{t}^*)) \le z(X^{\mathrm{H}}(\mathbf{t}^*)) \le z(X^*) \le z(X^{\mathrm{UB}}(\mathbf{t}^*))$.*

Proof. We have $z(X^{\mathrm{LB}}(\mathbf{t}^*)) \le z(X^{\mathrm{H}}(\mathbf{t}^*))$ since $X^{\mathrm{LB}}(\mathbf{t}^*) \le X^{\mathrm{H}}(\mathbf{t}^*)$ and all cost functions are nondecreasing. The next inequality $z(X^{\mathrm{H}}(\mathbf{t}^*)) \le z(X^*)$ holds since $X^{\mathrm{H}}(\mathbf{t}^*) \in \mathfrak{X}$ and X^* is optimal. The final inequality is a result from Lemma 4 and the optimality of $\mathbf{t}^*$ for (16)–(21), which is a relaxation of (1)–(3). □

We now state an approximation result that guarantees how well the heuristic solution performs relative to optimality. We measure the performance of $z(X^{\mathrm{H}}(\mathbf{t}^*)) - z(X^{\mathrm{LB}}(\mathbf{t}^*))$ relative to $z(X^*) - z(X^{\mathrm{LB}}(\mathbf{t}^*))$. Note that if $\mathbf{t}^* \in [0,1)^m$, then $X^{\mathrm{LB}}(\mathbf{t}^*) = [0]^{b\times m}$ (the zero matrix), and otherwise $X^{\mathrm{LB}}(\mathbf{t}^*)$ has at least one positive element. Thus, using $z(X^{\mathrm{LB}}(\mathbf{t}^*))$ as a baseline gives a result that is at least as strong as comparing $z(X^{\mathrm{H}}(\mathbf{t}^*)) - z([0])$ to $z(X^*) - z([0])$ (i.e., a baseline of the shortest-path objective with no interdiction action), which, in turn, is stronger than comparing $z(X^{\mathrm{H}}(\mathbf{t}^*))$ to $z(X^*)$ (i.e., a baseline of zero).

Theorem 3. *The objective $z(X^{\mathrm{H}}(\mathbf{t}^*)) - z(X^{\mathrm{LB}}(\mathbf{t}^*)) > (1 - 1/e)(z(X^*) - z(X^{\mathrm{LB}}(\mathbf{t}^*)))$, and this bound is tight.*

Proof. If $\mathbf{t}^* \in \mathbb{Z}^m$, then $X^{\mathrm{LB}}(\mathbf{t}^*) = X^{\mathrm{UB}}(\mathbf{t}^*)$, $X^{\mathrm{H}}(\mathbf{t}^*)$ is optimal from Proposition 2, and the result follows. Assume then $\mathbf{t}^* \notin \mathbb{Z}^m$, and by implication, $b'' = b - b' \ge 1$. It suffices to prove that $\mathrm{E}(c_a(\tau)|\mathbf{x}_a^{\mathrm{H}}) - \mathrm{E}(c_a(\tau)\,|\mathbf{x}_a^{\mathrm{LB}}) > (1 - 1/e)(\mathrm{E}(c_a(\tau)|\mathbf{x}_a^{\mathrm{UB}}) - \mathrm{E}(c_a(\tau)|\mathbf{x}_a^{\mathrm{LB}}))$, $\forall a \in \mathcal{A}$, where $\mathbf{x}_a^{\mathrm{H}}$, $\mathbf{x}_a^{\mathrm{LB}}$, and $\mathbf{x}_a^{\mathrm{UB}}$ denote the a^{th} column of $X^{\mathrm{H}}(\mathbf{t}^*)$, $X^{\mathrm{LB}}(\mathbf{t}^*)$, and $X^{\mathrm{UB}}(\mathbf{t}^*)$, respectively. This result guarantees that $z(X^{\mathrm{H}}(\mathbf{t}^*)) - z(X^{\mathrm{LB}}(\mathbf{t}^*)) > (1-1/e)(z(X^{\mathrm{UB}}(\mathbf{t}^*)) - z(X^{\mathrm{LB}}(\mathbf{t}^*)))$, and the theorem follows by Proposition 2. Let $a \in \mathcal{A}$. We note that

$$\begin{aligned}
\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a^{\mathrm{LB}}\big) &= c_a\big(\lfloor t_a^* \rfloor\big),\\
\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a^{\mathrm{UB}}\big) &= c_a\big(\lfloor t_a^* \rfloor\big) + \widehat{t}_a\big(c_a\big(\lfloor t_a^* \rfloor + 1\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big), \text{ and}\\
\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a^{\mathrm{H}}\big) &= c_a\big(\lfloor t_a^* \rfloor\big) + \sum_{\tau=1}^{b''}\big(c\big(\tau + \lfloor t_a^* \rfloor\big)\\
&\quad - c_a\big(\lfloor t_a^* \rfloor\big)\big)\binom{b''}{\tau}\big(\widehat{t}_a/b''\big)^{\tau}\big(1 - \widehat{t}_a/b''\big)^{b''-\tau}.
\end{aligned}$$

Using these values, we proceed with our algebraic reduction:

$$\begin{aligned}
\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a^{\mathrm{H}}\big) - \mathrm{E}\big(c_a(\tau)|\mathbf{x}_a^{\mathrm{LB}}\big) &= \Bigg(\sum_{\tau=1}^{b''}\big(c\big(\tau + \lfloor t_a^* \rfloor\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\binom{b''}{\tau}\\
&\quad \times \big(\widehat{t}_a/b''\big)^{\tau}\big(1 - \widehat{t}_a/b''\big)^{b''-\tau}\\
&\quad + c_a\big(\lfloor t_a^* \rfloor\big)\Bigg) - \big(c_a\big(\lfloor t_a^* \rfloor\big)\big)\\
&\ge \big(c_a\big(1 + \lfloor t_a^* \rfloor\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\sum_{\tau=1}^{b''}\binom{b''}{\tau}\left(\frac{\widehat{t}_a}{b''}\right)^{\tau}\left(1 - \frac{\widehat{t}_a}{b''}\right)^{b''-\tau} && (22)\\
&= \big(c_a\big(1 + \lfloor t_a^* \rfloor\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\left(1 - \left(1 - \frac{\widehat{t}_a}{b''}\right)^{b''}\right)\\
&> \big(c_a\big(1 + \lfloor t_a^* \rfloor\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\left(1 - \lim_{b''\to\infty}\left(1 - \frac{\widehat{t}_a}{b''}\right)^{b''}\right) && (23)\\
&= \big(c_a\big(1 + \lfloor t_a^* \rfloor\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\left(1 - \frac{1}{e^{\widehat{t}_a}}\right)\\
&\ge \big(c_a\big(1 + \lfloor t_a^* \rfloor\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\left(1 - \frac{1}{e}\right)\big(\widehat{t}_a\big) && (24)\\
&= (1 - 1/e)\big(\widehat{t}_a\big(c_a\big(\lfloor t_a^* \rfloor + 1\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\\
&\quad + c_a\big(\lfloor t_a^* \rfloor\big) - c_a\big(\lfloor t_a^* \rfloor\big)\big)\\
&= (1 - 1/e)\big(\mathrm{E}\big(c_a(\tau)|\mathbf{x}_a^{\mathrm{UB}}\big) - \mathrm{E}\big(c_a(\tau)|\mathbf{x}_a^{\mathrm{LB}}\big)\big). && (25)
\end{aligned}$$

The inequality in (22) holds by noting that, in the summation, $c_a(\tau + \lfloor t_a^* \rfloor) \ge c_a(1 + \lfloor t_a^* \rfloor)$ since c_a is nondecreasing. The inequality in (23) holds because $(1 - \widehat{t}_a/b'')^{b''}$ is increasing in $b'' \le b < \infty$. (This fact and the inequality in (24) are both proven using calculus arguments that we omit for brevity.) The inequalities in (22) and (25) becomes tight when $c_a(1 + \lfloor t_a^* \rfloor) = c_a(\tau + \lfloor t_a^* \rfloor)$ for $\tau > 1$; inequality (23) holds in the limit as b'' becomes arbitrarily large; and inequality (24) becomes tight either when $\widehat{t}_a = 0$ or in the limit as $\widehat{t}_a$ tends toward 1. Therefore, the ratio $1 - 1/e$ is tight. □

4. Sample Average Approximation for General SPIP-RS

In this section we briefly describe a sample average approximation (SAA) based formulation to find near-optimal solutions to the SPIP-RS. The method solves stochastic programs of the form $\min \mathrm{E}_{\Omega} g(\mathbf{x}, \omega)$, where the random variable $\omega \in \Omega$ is independent of $\mathbf{x}$. SAA uses Monte Carlo simulation to draw an independent and identically distributed set of scenarios, $\widehat{\Omega} \subset \Omega$. The sampling must be independent of the decision variables. The method then approximates an optimal solution by minimizing the sample average: $\sum_{\omega \in \widehat{\Omega}} g(\mathbf{x}, \omega)/|\widehat{\Omega}|$. Shapiro et al. (2009) show that the standard error for the SAA solution is given by $\mathcal{O}(|\widehat{\Omega}|^{-1/2})$.

The SPIP-RS exhibits decision-dependent uncertainty (see Jonsbraten et al. 1998, Goel and Grossmann 2006), since the random variables represent interdiction locations, which depend on the decision variables X. To provide an independent scenario generation, therefore, we will generate scenario vectors using a uniform 0–1 distribution, $\boldsymbol{\omega}_s \in (0,1)^b$ for $s \in \mathcal{S}$, where $\mathcal{S}$ is an indexing set of scenarios. Then for each interdiction $h \in \mathcal{H}$ in each scenario $s \in \mathcal{S}$, we will assign interdiction h to arc a if and only if

$$\sum_{a'=1}^{a-1} x_{ha'} \le \omega_{hs} < \sum_{a'=1}^{a} x_{ha'} \tag{26}$$

To accommodate the strict inequality in (26), we define $\Psi_h = \cup_{s \in \mathcal{S}} \{\omega_{hs}\} \cup \{0,1\}$ for all $h \in \mathcal{H}$, and we set $2\epsilon_h$ to be the smallest difference between any pair of values in Ψ_h, for each $h \in \mathcal{H}$. We ensure in our random vector generation process that each value in Ψ_h is distinct (by resampling if necessary).

To formulate our SAA-based formulation as a mixed-integer linear program, we introduce (with a slight abuse of notation) auxiliary binary variables χ_{has} that equal 1 if and only if interdiction $h \in \mathcal{H}$ is applied to arc $a \in \mathcal{A}$ in scenario $s \in \mathcal{S}$, and we introduce binary auxiliary variables $I_{\tau as}$ that equal 1 if and only if exactly $\tau \in \mathcal{H} \cup \{0\}$ interdictions occur on arc $a \in \mathcal{A}$ in scenario $s \in \mathcal{S}$. Our SAA mixed-integer linear program is

$$\max_{X \in \mathfrak{X}} \ d_s \tag{27}$$

$$\text{s.t.} \quad \chi_{has} \le 1 + \sum_{a'=1}^{a} x_{ha'} - \omega_{hs} - \epsilon_h \qquad \forall (h,a,s) \in \mathcal{H} \times \mathcal{A} \times \mathcal{S}, \tag{28}$$

$$\chi_{has} \le \omega_{hs} + \sum_{a'=a}^{m} x_{ha'} \qquad \forall (h,a,s) \in \mathcal{H} \times \mathcal{A} \times \mathcal{S}, \tag{29}$$

$$\sum_{\tau \in \{0\} \cup \mathcal{H}} I_{\tau as} = 1 \qquad \forall (a,s) \in \mathcal{A} \times \mathcal{S}, \tag{30}$$

$$\sum_{h \in \mathcal{H}} \chi_{has} = \sum_{\tau \in \{0\} \cup \mathcal{H}} \tau I_{\tau as} \qquad \forall (a,s) \in \mathcal{A} \times \mathcal{S}, \tag{31}$$

$$d_i - d_j \le \frac{1}{|\mathcal{S}|} \left(\sum_{s \in \mathcal{S}} \sum_{\tau \in \{0\} \cup \mathcal{H}} c_a(\tau) I_{\tau as} \right) \qquad \forall a = (i,j) \in \mathcal{A}, \tag{32}$$

$$d_t = 0, \tag{33}$$

$$\chi_{has} \in \{0,1\} \quad \forall (h,a,s) \in \mathcal{H} \times \mathcal{A} \times \mathcal{S}, \tag{34}$$

$$I_{\tau as} \in \{0,1\} \quad \forall (\tau,a,s) \in \{0\} \cup \mathcal{H} \times \mathcal{A} \times \mathcal{S}. \tag{35}$$

Objective (27) is the same as (1). Constraints (28) and (29) define the χ-variables so that $\chi_{has} = 1$ only if $\sum_{a'=1}^{a-1} x_{ha'} \le \omega_{hs} < \sum_{a'=1}^{a} x_{ha'}$. Note that the maximization objective induces $\chi_{has} = 1$ whenever $\sum_{a'=1}^{a-1} x_{ha'} \le \omega_{hs} < \sum_{a'=1}^{a} x_{ha'}$. Constraints (30) and (31) force $I_{\tau as}$ to their desired values. Constraints (32) correspond to (2), replacing the true expected arc cost with the sample average arc cost on the RHS, and constraint (33) is equivalent to (3). Finally, constraints (34)–(35) state integrality constraints on the χ- and I-variables. However, we can relax the integrality constraints on the I-variables corresponding to any discrete-concave function c_a (see Holzmann 2019 for details).

5. Computational Study

Our computational study focuses on the following questions. Each question is addressed in a separate subsection.

Question 1: What strategy is most efficient in solving general SPIP-RS instances of varying sizes?

Question 2: How well does the branch-and-bound algorithm described in Section 2.2 perform on convex instances?

Question 3: How well does the approximation heuristic of Section 3.2 perform on concave instances?

For this study, we coded each algorithm in Python v. 2.7. All runs were performed on Clemson University's Palmetto Cluster; each instance was run on a separate Intel Xeon E5-2670 (2.50 GHz) core. Collectively, the cores shared 60 GB of memory. For each run, we recorded computational time, termination status, and solution quality information. The running time for each algorithm was capped at two hours (wall-time), at which point the algorithm was terminated and partial results were collected.

Our test bed consisted of 75 randomly generated instances: 15 each having 10, 20, 30, 50, and 80 nodes. For all instances, we used an interdiction budget of $b = 4$. For each instance, we used igraph's (Csardi and Nepusz 2006) Erdős-Renyí module (Erdős and Rényi 1959) to randomly generate arcs with 30% arc density (i.e., 27, 114, 261, 735, and 1,896 arcs, respectively). The 15 instances for each graph size were divided into five convex instances, five concave instances, and five instances having general (i.e., unspecified) cost functions. We randomly generated convex and concave arc cost functions using $b+1$ random integers in $\{0,\ldots,10b\}$, sorted in increasing (for convex) or decreasing (for concave) order. Denoting the sorted integers for arc a by $\{\delta_0,\delta_1,\ldots,\delta_b\}$, we assigned

$$c_a(\tau) = \begin{cases} \delta_0 & \tau = 0, \\ c_a(\tau-1)+\delta_\tau & \tau \in \mathcal{H}. \end{cases}$$

For the instances with general cost functions, we simply sorted the random numbers in increasing order and assigned the function values $c_a(\tau)=\delta_\tau$, $\forall \tau \in \mathcal{H}\cup\{0\}$.

For baseline comparison, we used Artelys Knitro v. 11.1 (Byrd et al. 2006) as a general-purpose nonlinear solver (using the software's Python application programming interface). Knitro primarily uses ascent-based methods; hence, its results are not guaranteed to be optimal. We allowed Knitro to automatically determine both gradient and Hessian information.

5.1. SAA and Knitro Comparison

First we examine the performance of the SAA algorithm of Section 4. We compared Gurobi v. 7.5 solving (27)–(35) and Knitro solving (1)–(3). We used $|\mathcal{S}| = 50$ scenarios.

Initially, we compared these algorithms on the 10-node instances. Despite the small instance sizes, the SAA formulation could only be solved to optimality in 3 of the 15 instances within the time limit (see Table 5). The presence of $\mathcal{O}(bm|\mathcal{S}|)$ binary variables appears to make formulation (27)–(35) exceedingly difficult to solve. In contrast, Knitro was able to meet convergence criteria on all 10-node instances within 20 seconds. However, without solution quality information from the SAA model we could not assess the quality of solutions provided by Knitro.

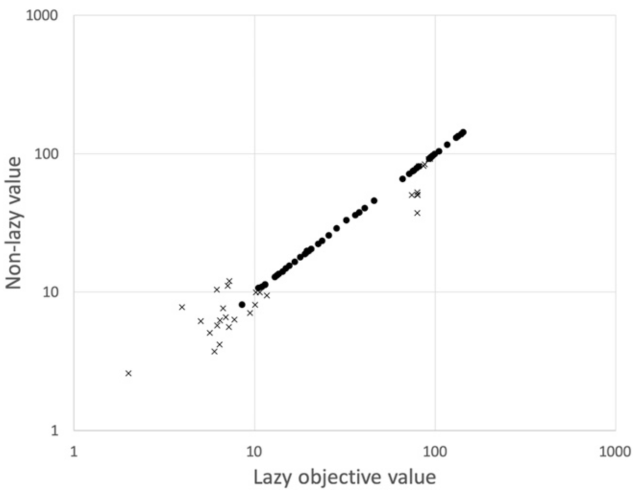
To obtain better results, therefore, we implemented a lazy approach to constructing our formulations, motivated by similar strategies for the SPIP (Israeli and Wood 2002, Lozano and Smith 2017). This approach optimizes SPIP-RS instances over a subgraph of the original graph, expanding the subgraph only as needed (see Appendix B for details). We repeated the previous computational experiment and report the results in Table 5. The first column lists the instance name. The next two columns report the objective of the best solution obtained by the end of each original algorithm at the time it terminated, whereas the fourth and fifth columns provide the computational times required by those two algorithms. The sixth, seventh, ninth, and tenth columns provide analogous results for the lazy approach. The eighth column reports values for z_{UB}, which is the best (i.e., minimal) upper bound on the SAA objective, as reported by Gurobi, among all of the lazy iterations. In two instances, Convex_$n10$_$i1$ and General_$n10$_$i4$, Gurobi failed to converge on even the first lazy iteration, which implies that the z_{UB} values corresponding to those instances may be excessively large. With the lazy approach, SAA was able to solve one additional instance within two hours, but still failed to meet the convergence criteria for the remaining 11 instances. In contrast, Knitro returned a result in less than two seconds in most cases when using the lazy approach. Furthermore, since Gurobi was able to converge in the first few iterations of the lazy approach on most

Table 5. Comparison of SAA (with $|\widehat{\Omega}| = 50$ Scenarios) and Knitro on SPIP-RS Instances with 10 Nodes and 27 Arcs

	Original algorithms				"Lazy" approach				
	Objective		Time (sec)		Objective			Time (sec)	
Instance	*SAA*	*Knitro*	*SAA*	*Knitro*	*SAA*	*Knitro*	z_{UB}	*SAA*	*Knitro*
Convex_$n10$_$i1$	87.74	94.00	>7200	8.26	6.00	94.00	140.14	>7200	0.24
Convex_$n10$_$i2$	35.69	40.72	>7200	9.91	13.00	40.72	57.38	>7200	0.82
Convex_$n10$_$i3$	74.77	76.00	3,501	13.29	74.04	76.00	76.56	0.56	0.60
Convex_$n10$_$i4$	38.59	45.81	>7200	11.64	8.00	45.81	126.86	>7200	1.11
Convex_$n10$_$i5$	138.14	143.00	>7200	7.92	142.49	143.00	144.68	63.92	0.18
Concave_$n10$_$i1$	75.00	75.00	0.51	13.58	75.00	75.00	75.00	0.03	0.18
Concave_$n10$_$i2$	77.92	79.00	0.35	11.03	79.00	79.00	79.00	0.03	0.15
Concave_$n10$_$i3$	90.22	97.00	>7200	15.39	69.00	97.00	129.62	>7200	2.31
Concave_$n10$_$i4$	122.51	130.75	>7200	19.69	103.00	130.74	143.76	>7200	0.47
Concave_$n10$_$i5$	128.77	134.04	>7200	12.77	113.19	134.04	151.11	>7200	2.32
General_$n10$_$i1$	27.96	28.91	>7200	11.85	10.00	28.42	33.00	>7200	2.56
General_$n10$_$i2$	15.12	16.66	>7200	6.11	4.00	16.66	27.24	>7200	0.76
General_$n10$_$i3$	21.16	23.60	>7200	6.20	17.00	23.60	25.02	>7200	0.34
General_$n10$_$i4$	31.49	37.87	>7200	10.12	14.00	37.89	76.94	>7200	6.18
General_$n10$_$i5$	30.57	33.23	>7200	8.00	23.00	32.24	38.76	>7200	0.42

Operations Research, 2021, vol. 69, no. 1, pp. 82–99, © 2020 INFORMS

Figure 3. Effect of the Lazy Approach on the Objective



Note. The "X" markers indicate that the nonlazy approach timed out before converging.

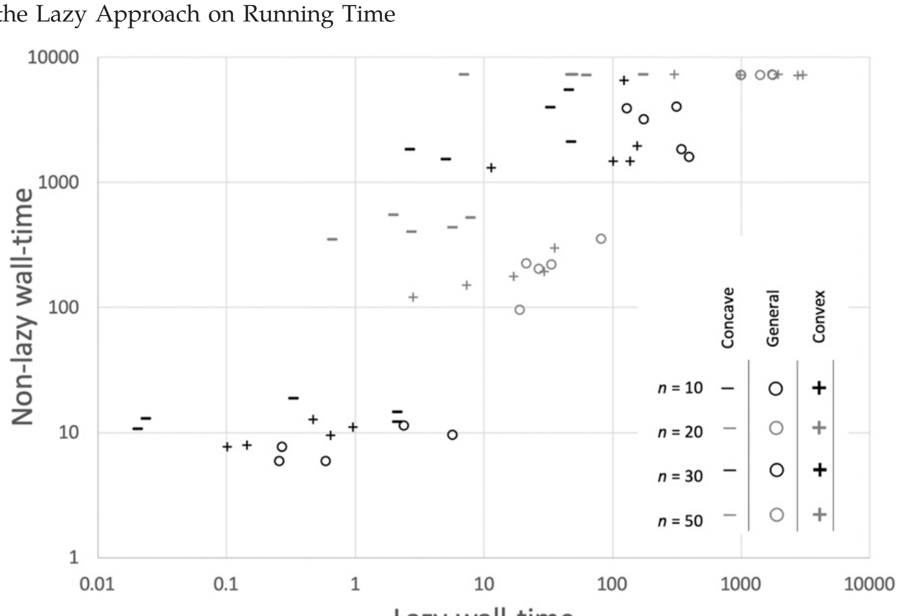
instances, that allowed us to obtain valid upper bounds and provides additional confidence that Knitro's solutions are near-optimal.

Based on the results from our smallest instances, we only used the Knitro solver for the remaining 60 instances, and we examined its performance with and without the lazy approach on all 75 instances. Our study confirms that, with respect to objective quality, the lazy approach and nonlazy approaches both yielded approximately equal objectives (see Figure 3). The notable exception was that the lazy approach appeared to become trapped in locally optimal solutions for our 80-node instances with general cost functions, since it met convergence criteria with an objective that was significantly worse than the nonlazy variant's objective. These are the points above the diagonal in Figure 3. With regard to computational time (see Figure 4), the lazy approach consistently terminated in a similar timeframe or faster than the nonlazy approach. For the larger instances with 50 and 80 nodes the nonlazy variant timed out on all 30 instances, whereas the lazy approach successfully converged for all 50-node instances within two hours, and 12 of 15 instances having 80 nodes. (For ease of display, the results from the 80-node instances are not depicted in Figure 4.) Finally, in Figure 5 and Table 6, we display the lazy variant's running time on all 75 instances and provide summary statistics, respectively. The algorithm exhibits exponential growth in solver times, as expected, with a slight tapering effect due to the fact that the lazy approach ignores a significant number of arcs in the network.

5.2. Branch-and-Bound Performance on Convex Problems

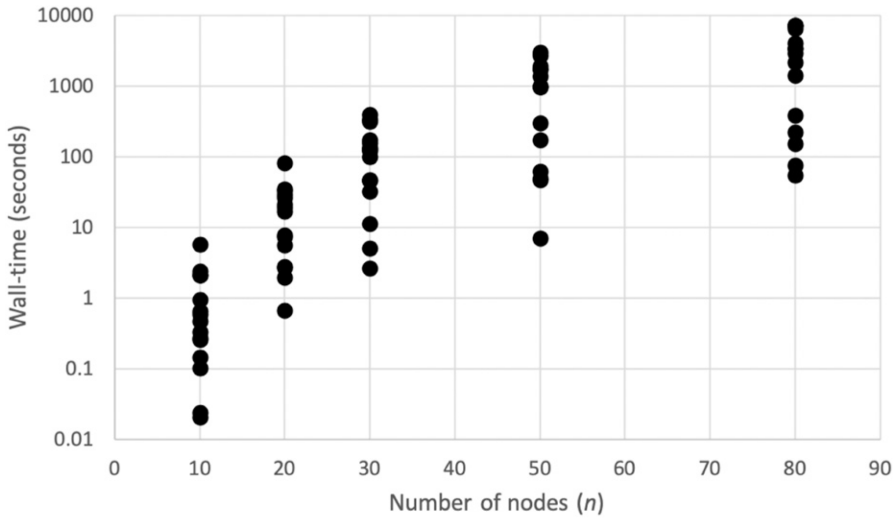
We examined the performance of the branch-and-bound algorithm from Section 2.2 on our convex test bed instances, comparing the computational time and solution quality to that provided by Knitro. We examined the performance of the branch-and-bound algorithm both with and without the lazy approach described in Section 5.1. For our optimality tolerance we used $\epsilon = 0.01$. The results are shown in Table 7. In the columns showing the number of nodes in the branch-and-bound tree, for the lazy approach we include both the total number of nodes over all

Figure 4. Effect of the Lazy Approach on Running Time



Note. Instances with $n = 80$ nodes are excluded.

Figure 5. Running Time (in Seconds) for Knitro (Using the Lazy Approach) by Number of Graph Nodes (n)



iterations and the number of nodes in the iteration having the largest tree.

Both the lazy and nonlazy variants were able to solve all of the 10- and 20-node instances to the specified optimality tolerance within two hours. In most of these instances, the lazy approach was faster: Usually it took on the order of half the time as the nonlazy approach. The notable exception is the Convex_n20_i4 instance, where the lazy approach took about 20% longer than the nonlazy approach. This occurred because the subgraphs in the last seven lazy iterations each had only one arc more than the previous iteration's subgraph; thus, these iterations each took similar (significant) effort to solve. Both the lazy and nonlazy approaches solved three of the five 30-node instances in under two hours, and with one exception, both failed to finish in two hours on all 50- and 80-node instances. The exception was instance Convex_n50_i4, where the lazy approach finished but the nonlazy timed out. In every case where both branch-and-bound variants timed out, the nonlazy approach identified a better incumbent solution than the lazy variant.

In most instances, the quality of the solutions from Knitro and the branch-and-bound algorithms are identical. This again validates the strength of Knitro's ascent approach, since it found an optimal solution much faster than the exact method. However, the strength of the branch-and-bound algorithm is seen in the Convex_n20_i3 instance, where the exact branch-and-bound algorithm provides a significantly better solution. Thus, for small- and medium-sized instances, which are complex enough such that Knitro occasionally returns a significantly suboptimal solution, the branch-and-bound solver can tractably find a true optimal solution.

5.3. Quality of the Approximation Heuristic on Concave Problems

The polynomial-time heuristic algorithm described in Section 3.2 terminated on all instances in our test set in less than one second. Recall from Section 3.2 that the heuristic must provide an objective within $1-1/e \approx 63.2\%$ of the true optimum. We compare in Table 8 the quality of the solutions from the approximation algorithm to those obtained by Knitro on each of the concave instances in our test bed. For every instance, the approximation heuristic is within approximately 95% of the Knitro result, and in one instance (Concave_n20_i1) the approximation heuristic found a better solution than Knitro.

Given the quality of these solutions, we also examined whether warm-starting Knitro with an initial solution from the approximation algorithm improves

Table 6. Summary Statistics for Knitro with the Lazy Approach

# of nodes	Running time		
	Min	Mean	Max
10	0.01	1.05	5.06
20	0.62	21.81	64.70
30	2.79	143.71	473.60
50	8.48	1,061.18	2,587.63
80	54.14	3,237.02	>7200

Table 7. Branch-and-Bound Study Results

	Running time			Objective value			Spatial B-&-B tree size		
	Knitro	B-&-B		Knitro	B-&-B		NL	Lazy	
Instance	Lazy	NL	Lazy	Lazy	NL	Lazy		Total	Largest
Convex_n10_i1	0.14	0.08	0.02	94.00	94.00	94.00	4	7	4
Convex_n10_i2	0.64	0.21	0.08	41.72	41.72	41.72	19	26	19
Convex_n10_i3	0.47	0.05	<0.005	76.00	76.00	76.00	1	1	1
Convex_n10_i4	0.96	0.31	0.14	45.81	45.81	45.81	31	39	28
Convex_n10_i5	0.10	5.05	<0.005	143.00	143.00	143.00	1	1	1
Convex_n20_i1	29.28	33.68	14.54	20.54	20.53	20.53	1,039	2,357	1,004
Convex_n20_i2	35.41	33.05	8.24	25.76	25.76	25.76	948	1,462	822
Convex_n20_i3	2.80	5.64	0.29	26.01	55.84	55.84	22	63	27
Convex_n20_i4	7.32	984.02	1,173.18	11.39	13.37	13.37	38,311	143,019	37,522
Convex_n20_i5	17.06	538.42	328.57	10.94	10.95	10.95	16,824	41,688	13,960
Convex_n30_i1	155.55	>7200	>7200	12.90	12.58	8.99	75,419	182,820	145,952
Convex_n30_i2	100.88	>7200	>7200	13.59	13.34	9.00	74,352	264,589	114,923
Convex_n30_i3	122.81	505.48	209.49	17.89	17.89	17.89	7,468	24,418	8,443
Convex_n30_i4	136.31	1,341.32	1,273.20	13.23	13.53	13.53	18,136	108,242	36,341
Convex_n30_i5	11.44	100.05	61.94	14.86	14.86	14.86	1,524	8,945	3,115
Convex_n50_i1	2,758.40	>7200	>7200	6.22	5.77	2.00	31,443	255,469	81,608
Convex_n50_i2	973.36	>7200	>7200	10.06	10.04	8.00	31,178	189,538	109,524
Convex_n50_i3	3,022.38	>7200	>7200	7.73	7.68	5.00	31,535	210,228	107,418
Convex_n50_i4	1,930.19	>7200	>7200	6.95	6.48	2.75	30,998	210,189	145,877
Convex_n50_i5	301.10	>7200	2,117.56	10.69	10.68	10.69	31,664	130,567	43,234
Convex_n80_i1	4,093.03	>7200	>7200	7.21	7.18	4.00	11,363	219,805	82,761
Convex_n80_i2	3,411.52	>7200	>7200	6.40	6.11	2.00	11,339	158,842	103,030
Convex_n80_i3	6,529.53	>7200	>7200	5.04	5.70	3.00	11,416	192,366	89,935
Convex_n80_i4	>7200	>7200	>7200	2.00	4.79	2.00	11,331	282,376	72,163
Convex_n80_i5	>7200	>7200	>7200	6.00	6.41	3.00	11,204	231,004	89,807

Note. "NL" indicates the nonlazy approach, in contrast to the lazy one described in Section 5.1.

Table 8. Comparing Objectives from Knitro and the Approximation Algorithm of Section 3.2 on Convex Instances

Instance	Objective		Percentage
	Knitro	*Approx*	
Concave_n10_i1	75.00	75.00	100.0%
Concave_n10_i2	79.00	79.00	100.0%
Concave_n10_i3	97.00	92.04	94.9%
Concave_n10_i4	130.74	130.74	100.0%
Concave_n10_i5	134.04	134.04	100.0%
Concave_n20_i1	80.79	80.96	100.2%
Concave_n20_i2	139.00	136.52	98.2%
Concave_n20_i3	99.44	96.82	97.5%
Concave_n20_i4	95.27	92.82	97.4%
Concave_n20_i5	66.10	66.10	100.0%
Concave_n30_i1	116.32	115.04	98.9%
Concave_n30_i2	92.44	90.17	97.5%
Concave_n30_i3	104.71	104.32	99.6%
Concave_n30_i4	71.66	71.66	100.0%
Concave_n30_i5	94.27	90.85	96.4%
Concave_n50_i1	94.12	89.78	95.4%
Concave_n50_i2	86.25	85.83	99.5%
Concave_n50_i3	81.24	79.89	98.3%
Concave_n50_i4	87.55	84.19	96.2%
Concave_n50_i5	91.97	90.18	98.1%
Concave_n80_i1	79.61	76.43	96.0%
Concave_n80_i1	80.23	76.34	95.1%
Concave_n80_i1	79.62	77.78	97.7%
Concave_n80_i1	74.44	73.72	99.0%
Concave_n80_i1	79.61	78.11	98.1%

Note. The percentage column is given by dividing the approximation objective by the Knitro objective.

the computational performance of Knitro. However, preliminary computational experiments showed no benefit to providing this warm start.

6. Conclusion and Future Work

In this paper we examine the SPIP-RS in which multiple interdiction actions impact arc cost functions in a potentially nonlinear fashion. When these cost functions are all linear, classical analysis reduces this problem to a linear program. When these cost functions are nonlinear, the problem can be formulated as a continuous nonconvex optimization problem. However, the SPIP-RS becomes NP-hard in the strong sense in this case, even if the cost functions are all concave or if they are all convex. We then provide algorithms to solve special cases of the SPIP-RS. For the convex SPIP-RS, we prescribe a spatial branch-and-bound algorithm that returns a solution that is within a specified absolute optimality tolerance. For the concave SPIP-RS, we provide a polynomial-time $(1-1/e)$-approximation algorithm, whose empirical performance indicates that the optimality gap is far tighter than the worst-case gap. Finally, we present an SAA-based linear mixed-integer programming model that can be solved to provide an approximate solution to the SPIP-RS. However, computational results show that simply employing a nonlinear optimization solver on the original nonlinear model for the SPIP-RS tends to provide better results with far less computational effort than solving the SAA-based formulation using a mixed-integer programming solver.

Further research on the SPIP-RS might characterize problem complexity and approximability results under special cases of the problem. These cases may regard assumptions on the number of nonlinear arc cost functions, conditions regarding the shape of these functions, or restrictions on the topology of the network itself. Additionally, future research questions may consider variants of the SPIP-RS itself. For instance, one may examine the case in which the follower uses a robust approach to selecting a preferred path, possibly using value-at-risk computations, instead of the shortest expected cost. Alternatively, one might consider a simultaneous variant, where the follower cannot observe the interdiction strategy prior to selecting a path. Such a model may also provide better approximation bounds for the Stackelberg game presented in this paper. Yet another variation of this problem might extend our two-stage game into a three-stage game, where the agent selecting a path can fortify arcs before the interdicting agent begins the two-stage SPIP-RS problem studied here.

Acknowledgments

The authors are grateful to Artelys for providing a high-performance cluster license of their software for this study. We also thank two anonymous referees for their helpful critiques. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Navy, the United States Air Force, the Department of Defense, or the United States Government.

Appendix A. Spatial Branch and Bound

Algorithm A.1 is a spatial branch-and-bound algorithm for solving the convex SPIP-RS within an absolute optimality tolerance gap of some given parameter $\epsilon > 0$. In each iteration, the algorithm solves the LP relaxation over the convex hull of (9)–(13). It then iteratively branches on t_a variables until a solution is available that is provably within ϵ of the true optimal objective.

Lines 1 and 2 initialize a node list for the branch-and-bound tree with a single root node. Each node, v, in the tree is associated with upper and lower bounds on the decision variables $\mathbf{t}$, denoted $\overline{\mathbf{t}^v}$ and $\underline{\mathbf{t}^v}$, respectively. For the root node, r, line 2 sets $\overline{\mathbf{t}^r} = \{b\}^m$ and $\underline{\mathbf{t}^r} = \{0\}^m$. For each node, immediately after setting $\overline{\mathbf{t}^v}$ and $\underline{\mathbf{t}^v}$, we solve the LP relaxation of (9)–(13) over the region bounded by $\overline{\mathbf{t}^v}$ and $\underline{\mathbf{t}^v}$ by interpolating the expected cost on the RHS of (37):

$$P\left(\overline{\mathbf{t}^v}, \underline{\mathbf{t}^v}\right):$$

$$\max \quad d_s^v \tag{36}$$

$$\text{s.t.} \quad d_i^v - d_j^v \le \mathrm{E}\left(c_{ij}(\tau)|\underline{t_{ij}^v}\right) + \left(\frac{t_{ij}^v - \underline{t_{ij}^v}}{\overline{t_{ij}^v} - \underline{t_{ij}^v}}\right)\left(\mathrm{E}\left(c_{ij}(\tau)|\overline{t_{ij}^v}\right) - \mathrm{E}\left(c_{ij}(\tau)|\underline{t_{ij}^v}\right)\right) \quad \forall (i,j) \in \mathcal{A}, \tag{37}$$

$$d_t^v = 0, \tag{38}$$

$$\underline{t_a^v} \le t_a^v \le \overline{t_a^v} \qquad \forall a \in \mathcal{A}, \tag{39}$$

$$\sum_{a \in \mathcal{A}} t_a = b, \tag{40}$$

where $\tau = \sum_{h \in \mathcal{H}} \chi_{ha}$ for each $a \in \mathcal{A}$ and $\chi_{ha} \sim \mathrm{Binom}(t_a/b, b)$ (based on Lemma 3). Since (36)–(40) is a relaxation of the SPIP-RS (from the interdictor's perspective), the optimal objective, d_s^v, is an upper bound on the optimal objective for (7) ≡ (1) within the same bounds. The initialization portion concludes in line 3 by defining a global lower bound value, LB, and setting it to $-\infty$.

After initialization is completed, the algorithm begins the branching and pruning process. Each iteration begins in lines 5–6 by choosing a node having the highest upper bound on the objective, d_s^v, and removing that node from the list of nodes to search. Line 7 checks whether the incumbent solution is within ϵ of d_s^v; if so, then all remaining nodes are fathomed and the algorithm returns the incumbent solution. Otherwise, line 8 checks if this node achieves an objective better than the current best lower bound, using $X^*(\mathbf{t}^v)$ defined as in (8) and $z(X^*(\mathbf{t}^v))$ computed using the method of Holzmann and Smith (2019) described in Section 1.3.

Operations Research, 2021, vol. 69, no. 1, pp. 82–99, © 2020 INFORMS

If $z(X^*(\mathbf{t}^v)) > \texttt{LB}$, then this node's objective is better than the incumbent, and lines 9–10 establish this node as the new incumbent. Next, line 11 effectively fathoms this node if $d_s^v \leq \texttt{LB} + \epsilon$. Otherwise, line 12 adds two child nodes. Branching occurs on an arc having the greatest gap between the interpolated cost and true expected cost, chosen in line 13. Lines 14–15 establish two children: one child is restricted by $t_a \leq t_a^*$, and the other is restricted by $t_a \geq t_a^*$. Finally, problem (36)–(40) is solved for each child, and the algorithm reiterates. When no unpruned nodes remain in the tree, the algorithm terminates in line 16, returning the incumbent solution.

Algorithm A.1 (Spatial Branch and Bound Algorithm)

Input: $G = (\mathcal{N}, \mathcal{A})$: A directed graph
$s, t \in \mathcal{N}$: Source and terminal nodes
$b \in \mathbb{Z}_+$: Interdiction budget
$\{c_a : \{0, \ldots, b\} \to \mathbb{R}_+\}_{a \in \mathcal{A}}$: Convex cost functions for each $a \in \mathcal{A}$
$\epsilon > 0$: Optimality tolerance parameter.
Output: $X \in [0,1]^{b \times m}$: An optimal interdiction strategy

1 $\texttt{NodeList} \leftarrow \{\texttt{r}\}$
2 $\underline{\mathbf{t}}^r \leftarrow \{0\}^m$; $\overline{\mathbf{t}}^r \leftarrow \{b\}^m$; $(\mathbf{t}^r, \mathbf{d}^r) \leftarrow \texttt{Solve}(\texttt{P}(\overline{\mathbf{t}}^r, \underline{\mathbf{t}}^r))$
3 $\texttt{LB} \leftarrow -\infty$
4 **while** $\texttt{NodeList} \neq \emptyset$ **do**
5 $\quad n \leftarrow \arg\max_{k \in \texttt{NodeList}} \{d_s^k\}$
6 $\quad \texttt{NodeList} \leftarrow \texttt{NodeList} \setminus \{\texttt{n}\}$
7 $\quad$ **if** $d_s^v > LB + \epsilon$ **then return** $X^*(\mathbf{t}^{\text{Incumbent}})$
8 $\quad$ **if** $z(X^*(\mathbf{t}^v)) > \texttt{LB}$ **then**
9 $\qquad \texttt{Incumbent} \leftarrow \texttt{n}$
10 $\qquad \texttt{LB} \leftarrow z(X^*(\mathbf{t}^v))$
11 $\quad$ **if** $d_s^v > LB + \epsilon$ **then**
12 $\qquad \texttt{NodeList} \leftarrow \texttt{NodeList} \cup \{c_1, c_2\}$
13 $\qquad a \leftarrow \arg\max_{\alpha \in \mathcal{A}} \{\mathrm{E}(c_\alpha(\tau) \,|\, \mathbf{x}^*(\underline{t_\alpha^v})) + (\frac{t_\alpha^v - \underline{t_\alpha^v}}{\overline{t_\alpha^v} - \underline{t_\alpha^v}})(\mathrm{E}(c_\alpha(\tau) \,|\, \mathbf{x}^*(\overline{t_\alpha^v})) - \mathrm{E}(c_\alpha(\tau) \,|\, \mathbf{x}^*(\underline{t_\alpha^v}))) - \mathrm{E}(c_\alpha(\tau) \,|\, \mathbf{x}^*(t_\alpha^v))\}$
14 $\qquad \underline{\mathbf{t}}^{c_1} \leftarrow \underline{\mathbf{t}}^v$; $\overline{\mathbf{t}}^{c_1} \leftarrow [\overline{t_1^v}, \ldots, \overline{t_{a-1}^v}, t_a^*, \overline{t_{a+1}^v}, \ldots, \overline{t_m^v}]$; $(\mathbf{t}^{c_1}, \mathbf{d}^{c_1}) \leftarrow \texttt{Solve}(P(\overline{\mathbf{t}}^{c_1}, \underline{\mathbf{t}}^{c_1}))$
15 $\qquad \underline{\mathbf{t}}^{c_2} \leftarrow [\underline{t_1^v}, \ldots, \underline{t_{a-1}^v}, t_a^*, \underline{t_{a+1}^v}, \ldots, \underline{t_m^v}]$; $\overline{\mathbf{t}}^{c_2} \leftarrow \overline{\mathbf{t}}^v$; $(\mathbf{t}^{c_2}, \mathbf{d}^{c_2}) \leftarrow \texttt{Solve}(P(\overline{\mathbf{t}}^{c_2}, \underline{\mathbf{t}}^{c_2}))$
16 **return** $X^*(\mathbf{t}^{\text{Incumbent}})$

Appendix B. Lazy Construction

Algorithm B.1 describes a "lazy" approach to implementing SPIP-RS algorithms in this paper. In each iteration, the algorithm directs the solver to examine a subgraph of the input graph. By using the returned interdiction strategy, the algorithm then identifies a shortest path on the original graph. If the shortest path is contained in the subgraph examined in the current iteration, then the algorithm terminates. Otherwise, the algorithm adds the arcs from the shortest path to generate a new subgraph, and the algorithm reiterates.

Line 1 initializes an empty subgraph arc set, $\mathcal{A}^0$, and an initial solution, $X^0 = [0]$. Lines 2–8 give the main loop. Each iteration, $k \in \{1, 2, \ldots\}$ begins in lines 3 and 4 by setting a vector, $\mathbf{w}$, of expected arc costs given by the previous iteration's strategy, X^{k-1}. Line 5 finds the follower's shortest expected-cost path on the full graph, G. If the shortest path uses only arcs contained in the previous iteration's subgraph, $\mathcal{A}^{k-1}$, then line 6 terminates the algorithm. Otherwise, line 7 generates a new arc set, $\mathcal{A}^k$, and line 8 solves the SPIP-RS problem on the subgraph induced by $\mathcal{A}^k$ to give an updated solution, X. In our implementation, we allowed the solvers to "warm start" in each iteration with the previous solution, X^{k-1}. Finally, lines 9–10 convert $X \in [0,1]^{b \times |\mathcal{A}^k|}$ to $X^k \in [0,1]^{b \times m}$ by adding zero-columns for arcs not in $\mathcal{A}^k$.

Algorithm B.1 (Lazy Algorithm)

Input: $G = (\mathcal{N}, \mathcal{A})$: A directed graph
$b \in \mathbb{Z}_+$: Interdiction budget
$\{c_a : \{0, \ldots, b\} \to \mathbb{R}_+\}_{a \in \mathcal{A}}$: Nondecreasing cost functions for each $a \in \mathcal{A}$
Dependencies: Dijkstra($G, u, v, \mathbf{w}$): Returns set of arcs representing a shortest $u \to v$ path in G, given arc costs $\mathbf{w} \in \mathbb{R}_+^m$
Solve(G, b): Returns an optimal (or near-optimal) solution to the SPIP-RS represented by G and budget b
Output: $X \in [0,1]^{b \times m}$: A (near-)optimal interdiction strategy

1 $\mathcal{A}^0 \leftarrow \{\}$; $X^0 \leftarrow [0]^{b \times m}$
2 **for** $k \in \{1, 2, \ldots\}$ **do**
3 $\quad$ **for** $a \in \mathcal{A}$ **do**
4 $\qquad w_a \leftarrow \sum_{t=0}^{b} c_a(t) \Pi^b(\mathbf{x}_a^{k-1}, t)$
5 $\quad P^k \leftarrow \texttt{Dijkstra}(G, s, t, \mathbf{w})$
6 $\quad$ **if** $P^k \subseteq \mathcal{A}^{k-1}$ **then return** X^{k-1}
7 $\quad \mathcal{A}^k \leftarrow \mathcal{A}^{k-1} \cup P^k$
8 $\quad X \leftarrow \texttt{Solve}(G(\mathcal{N}, \mathcal{A}^k), b)$
9 $\quad$ **for** $a \in \mathcal{A}$ **do**
10 $\qquad$ **if** $a \in \mathcal{A}^k$ **then** $\mathbf{x}_a^k \leftarrow \mathbf{x}_a$ **else** $\mathbf{x}_a^k \leftarrow [0]^b$

References

Bahamondes B, Correa J, Matuschke J, Oriolo G (2017) Adaptivity in network interdiction. Rass S, An B, Kiekintveld C, Fang F, Schauer S, eds., *Decision and Game Theory for Security* (Springer, Berlin), 40–52.

Ball MO, Golden BL, Vohra RV (1989) Finding the most vital arcs in a network. *Oper. Res. Lett.* 8(2):73–76.

Bar-Noy A, Khuller S, Schieber B (1995) The complexity of finding most vital arcs and nodes. Technical report UMIACS-TR-95-96, University of Maryland Institute for Advanced Computer Studies, College Park, MD.

Bayrak H, Bailey MD (2008) Shortest path network interdiction with asymmetric information. *Networks* 52(3):133–140.

Borndörfer R, Buwaya J, Sagnol G, Swarat E (2013) Optimizing toll enforcement in transportation networks: a game-theoretic approach. *Electron. Notes Discrete Math.* 41:253–260.

Borrero JS, Prokopyev OA, Sauré D (2015) Sequential shortest path interdiction with incomplete information. *Decision. Anal.* 13(1): 68–98.

Byrd RH, Nocedal J, Waltz RA (2006) KNITRO: An integrated package for nonlinear optimization. di Pillo G, Roma M, eds., *Large-Scale Nonlinear Optimization* (Springer, Boston), 35–59.

Cormican KJ, Morton DP, Wood RK (1998) Stochastic network interdiction. *Oper. Res.* 46(2):184–197.

Correa J, Harks T, Kreuzen VJC, Matuschke J (2017) Fare evasion in transit networks. *Oper. Res.* 65(1):165–183.

Csardi G, Nepusz T (2006) The igraph software package for complex network research. *InterJournal Complex Systems* 1695.

Erdős P, Rényi A (1959) On random graphs. *Publ. Math. Debrecen.* 6:290–297.

Falk JE, Soland RM (1969) Algorithm for separable nonconvex programming problems. *Management Sci.* 15(9):550–569.

Fulkerson DR, Harding GC (1977) Maximizing minimum source-sink path subject to a budget constraint. *Math. Programming* 13(1):116–118.

Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., Princeton, NJ).

Goel V, Grossmann IE (2006) A class of stochastic programs with decision dependent uncertainty. *Math. Programming* 106(2):355–394.

Held H, Woodruff DL (2005) Heuristics for multi-stage interdiction of stochastic networks. *J. Heuristics* 11(6):483–500.

Held H, Hemmecke R, Woodruff DL (2005) A decomposition algorithm applied to planning the interdiction of stochastic networks. *Naval Res. Logist.* 52(4):321–328.

Hemmecke R, Schultz R, Woodruff DL (2003) Interdicting stochastic networks. Woodruff DL, ed., *Network Interdiction and Stochastic Integer Programming* (Kluwer, Boston), 71–80.

Holzmann T (2019) Network interdiction models with uncertainties. Unpublished doctoral dissertation, Clemson University.

Holzmann T, Smith JC (2019) Optimizing randomized interdiction strategies in shortest path interdiction problems. Romeijn H, Schaefer A, Thomas R, eds., *Proc. 2019 IISE Annual Conf.* (Institute of Industrial and Systems Engineers, Norcross, GA), 1–6.

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

Janjarassuk U, Linderoth J (2008) Reformulation and sampling to solve a stochastic network interdiction problem. *Networks* 52(3):120–132.

Jonsbraten TW, Wets RJB, Woodruff DL (1998) A class of stochastic programs with decision dependent random elements. *Ann. Oper. Res.* 82:83–106.

Kutz GD, Cooney JW (2007) Border security: Security vulnerabilities at unmanned and unmonitored U.S. border locations. Technical report GAO-07-884T, General Accounting Office, Washington, DC.

Lozano L, Smith JC (2017) A backward sampling framework for interdiction problems with fortification. *INFORMS J. Comput.* 29(1):123–139.

Lunday BJ, Sherali HD (2012) Network interdiction to minimize the maximum probability of evasion with synergy between applied resources. *Ann. Oper. Res.* 196(1):411–442.

Marshall A, Olkin I (1979) *Inequalities: Theory of Majorization and Its Application* (Academies Press, New York).

Ostrowski A (1952) Sur quelques applications des fonctions convexes et concaves au sens de I. Schur. *J. Math. Pures Appl. (9)* 31: 253–292.

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

Salmerón J (2012) Deception tactics for network interdiction: A multiobjective approach. *Networks* 60(1):45–58.

Shapiro A, Dentcheva D, Ruszczynski A (2009) *Lectures in Stochastic Programming: Modeling and Theory* (SIAM, Philadelphia).

Smith JC, Prince M, Geunes J (2013) Modern network interdiction problems and algorithms. Pardalos PM, Du DZ, Graham R, eds., *Handbook of Combinatorial Optimization* (Springer, New York), 1949–1987.

Sullivan KM, Morton DP, Pan F, Smith JC (2014) Securing a border under asymmetric information. *Naval Res. Logist.* 61(2):91–100.

Washburn A, Wood RK (1995) Two-person zero-sum games for network interdiction. *Oper. Res.* 43(2):243–251.

Wood RK (1993) Deterministic network interdiction. *Math. Comput. Model.* 17(2):1–18.

Tim Holzmann is an assistant professor for operations research at the Air Force Institute of Technology. He received his PhD in industrial engineering from Clemson University, South Carolina. His research interests are network optimization and decision making under uncertainty.

J. Cole Smith is Dean of the College of Engineering and Computer Science at Syracuse University and prior to that served as Department Chair of Industrial Engineering at Clemson University. Smith currently serves on the board of directors for INFORMS and the board of trustees for IISE. His research focuses on integer programming, network optimization, and interdiction problems. He is a Fellow of the Institute of Industrial and Systems Engineers.