Risk Averse Shortest Path Interdiction

Yongjia Song* Siqian Shen†

Abstract

We consider a Stackelberg game in a network, where a leader minimizes the cost of interdicting arcs and a follower seeks the shortest distance between given origin and destination nodes under uncertain arc traveling cost. In particular, we consider a risk-averse leader, who aims to keep high probability that the follower's traveling distance is longer than a given threshold, interpreted by a chance constraint. For the case with a wait-and-see follower, i.e., the follower selects a shortest path after seeing realizations of the random arc cost, we propose a branch-and-cut algorithm and apply lifting techniques to exploit the combinatorial structure of the risk-averse leader's interdiction problem. For the case with a here-and-now follower, we formulate a monolithic mixed-integer linear programming formulation for benchmark. We demonstrate the computational efficacy of our approaches, risk-averse interdiction solution patterns, and result sensitivity, via testing instances of randomly generated grid networks and real-world transportation networks.

Key words: Stochastic network interdiction; chance constraint; mixed-integer programming; branch-and-cut; lifting

1 Introduction

The concept of *network* is in wide use for describing and optimizing real-world problems that arise in application areas including telecommunication, transportation, and logistics. In the related network flow models, arcs are associated with traveling time on road segments or operational cost of performing tasks, which can be random due to the uncertainty of traffic conditions or task durations, respectively. The shortest path problem is one of the most fundamental and well-studied network flow problems [cf. 2], where the objective is to seek a path that has the shortest traveling distance between an origin and a destination. In particular, we consider stochastic shortest path problems where random arc costs follow some underlying, known distribution. The problems can be classified into two main cases [41, 7]: The first case follows a static setting and identifies a path with the minimum expected length or with the highest probability of being sufficiently short [25, 11]; the second case considers dynamic traveling decisions to be made at various time stages, corresponding to network information that is dynamically acquired during the travel [34, 27].

[*] Department of Statistical Sciences and Operations Research, Virginia Commonwealth University, Richmond, VA, USA; email: ysong3@vcu.edu

[†] Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA 48109; email: siqian@umich.edu

In this paper, we focus on static shortest path problems in a two-person network interdiction game under the uncertainty of the traveling cost as well as the uncertainty of the interdiction effect. In general, network interdiction models characterize *Stackelberg games* [42] in networks, where one player (a *leader* or an *interdictor*) acts first, followed by the other player's (a *follower*) reaction. The leader seeks a set of arcs to be interdicted to minimize the maximum gain, or to maximize the minimum loss of the follower, e.g., to reduce the maximum flow [43] or to lengthen the shortest path [12, 18]. Recent studies generalize this setting and investigate interdiction problems that are relevant to adversarial processes in various applications including critical infrastructure analysis [30], nuclear smuggling [33, 29], and product introduction [37]. We refer the readers to Wood [44], Morton [28], Dimitrov and Morton [10], Alderson et al. [3] for recent surveys of general network interdiction problems and to Lim and Smith [22], Shen [36] for surveys of existing literature on two-stage network interdiction models and the related, mainly decomposition-based algorithms [16]. In the shortest path interdiction problems, the leader interdicts a set of network entities (e.g., adding traveling cost to arcs) to prolong the follower's shortest traveling distance between given origin and destination nodes, before knowing the realizations of traveling distances. Existing studies commonly consider that the leader's objective is to maximize the expected traveling cost of the follower subject to an interdiction budget [e.g., 17, 45, 19, 9], i.e., the leader is risk neutral.

In this paper, we investigate a class of stochastic shortest path interdiction problems, called the risk-averse shortest path interdiction problem (RASPI), where the leader is risk averse, and targets on maintaining a high chance that the follower has an undesirably long distance to travel. This problem setting employs the *Value-at-Risk* (VaR) measure, characterized by a chance constraint. The research incorporates a wide class of applications in homeland security, military logistics, and disaster relief, to which the probability that the follower's traveling distance exceeds a certain threshold is the key, e.g., given a strict time limit for successful nuclear smuggling, transporting military goods, or delivering perishable products by the follower from origin to destination, the leader aims to maximize the chance that the follower fails the task by prolonging the traveling distance on a subset of links. Besides the chance-constrained models, other risk-averse models have also been considered in the context of network interdiction problems, e.g., [8] proposes to use the theory of coherent risk measures to develop a risk-averse variant of the network interdiction problem.

Throughout the paper, we assume that the uncertain traveling cost vector and interdiction effect vector follow a joint probability distribution with finite support, and a set of scenarios are used to characterize the random outcomes. In this paper, we mainly focus on a RASPI model variant with a *wait-and-see* follower, such that the follower selects a traveling path after knowing realizations of the uncertainty, and thus travels on the shortest path after the leader's interdiction action in each scenario. As a benchmark, in Section 4 we discuss the case with a *here-and-now* follower, who decides a path solution before seeing realizations of the uncertain parameters. In particular, we consider a risk-neutral follower who aims to minimize the expected traveling distance and briefly discuss how to generalize the model when other risk behaviors of the follower are concerned. In both variants, the leader's risk-averse objective is characterized by a chance constraint in the RASPI.

Relevant to the solution methodologies we use for solving RASPI, we review a variety of integer-programming-based approaches in the existing literature for optimizing generic chance-constrained stochastic programming (CCSP) models, which are computationally intractable in general, including facet-defining inequalities and/or cutting-plane algorithms [see, e.g., 1, 24, 20, 23]. Song and Luedtke [38] study the problem of minimizing the cost of building a subset of arcs to keep sufficiently high probability that an origin node and a destination node are connected under random arc failures. Song and Zhang [40] extend the results for designing networks to maintain high probability of connecting multiple terminals represented by a Steiner tree. However, to the best of our knowledge, no studies have considered a CCSP model in a stochastic network interdiction setting. Most existing work considers the leader being risk neutral and optimizing the expected interdiction outcome subject to a fixed budget, while our studies focus on minimizing the interdiction cost to ensure a sufficiently low risk of not achieving a certain threshold value of the random outcome for the leader's interdiction action. We develop a branch-and-cut algorithm with strengthened inequalities through lifting to solve the RASPI with a wait-and-see follower.

We organize the remainder of the paper as follows. Section 2 sets up the model for the RASPI with a wait-and-see follower. Section 3 discusses a branch-and-cut algorithm that incorporates scenario-based pack inequalities and probabilistic pack inequalities. Section 4 discusses the RASPI with a here-and-now follower, for which we provide a mixed-integer programming formulation. In Section 5, via testing instances of grid networks and two real-world networks, we demonstrate the computational efficacy of our approaches, result sensitivity for key parameter changes, and arc-interdiction solution patterns. We conclude the paper and discuss future research in Section 6.

2 Models of Risk-Averse Shortest Path Interdiction

We consider a shortest path interdiction problem, in which a leader interdicts a subset of arcs to maximize the length of the shortest path chosen by a follower in a "probabilistic" sense. The key assumption here is that the interdictor is the leader of the game, who knows exactly what the follower tries to optimize, and her interdiction actions can be fully observed by the follower. Given a directed graph $G = (V, A)$, let $r \in \mathbb{R}_+^{|A|}$ be the interdiction cost vector, and let $X := \{x \in \{0,1\}^{|A|} \mid r^\top x \le r_0\}$ define feasible interdiction actions, where $r_0 \in \mathbb{R}_+$ denotes the budget. Nodes s and t are the origin and destination of the follower's path, respectively. For each node $i \in V$, $\delta^+(i)$ and $\delta^-(i)$ are the sets of outgoing and incoming arcs, respectively. Moreover, $c_a \in \mathbb{R}_+$ is the cost of traveling on arc a, and d_a is the amount of traveling-cost increase after the leader interdicts arc a. The deterministic shortest path interdiction problem is formulated as [see 18]:

$$\max_{x \in X} \ \min_{y} \sum_{a \in A} (c_a + x_a d_a) y_a \tag{1a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} y_a - \sum_{a \in \delta^-(i)} y_a = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{o.w.} \end{cases}, \ \forall i \in V, \tag{1b}$$

$$y_a \geq 0,\ \forall a \in A. \tag{1c}$$

We next consider the stochastic variant of the shortest path interdiction problem, where the cost vector $\tilde{c} \in \mathbb{R}_+^{|A|}$ and the interdiction vector $\tilde{d} \in \mathbb{R}_+^{|A|}$ are random vectors following given known distributions. Moreover, $(\tilde{c}, \tilde{d})$ follows a discrete probability distribution with a finite support, which can be characterized by a finite set of scenarios $\{c^k, d^k\}_{k \in N}$, where $c^k = (c_a^k, a \in A)^\mathsf{T} \in \mathbb{R}_+^{|A|}$ and $d^k = (d_a^k, a \in A)^\mathsf{T} \in \mathbb{R}_+^{|A|}$. Each scenario k is realized with probability $p_k,\ \forall k \in N$. (If $(\tilde{c}, \tilde{d})$ follows a continuous distribution, with the sample average approximation (SAA) method, we can generate N scenarios $\{c^k, d^k\}_{k \in N}$ using Monte-Carlo sampling and solve the SAA model.)

2.1 Wait-and-see follower

If the follower's decision is made *after* observing $\{c^k, d^k\}_{k \in N}$, then she simply solves a deterministic shortest path problem in each scenario k with each arc a's traveling cost being c_a^k if it is not interdicted, and being $(c_a^k + d_a^k)$ otherwise. For each scenario $k \in N$ and arc $a \in A$, we use variables $y_a^k \in \{0, 1\},\ a \in A$ to represent the follower's traveling decisions, such that $y_a^k = 1$ if arc a belongs to the follower's shortest path, and $y_a^k = 0$ otherwise. Let $y^k = [y_a^k,\ a \in A]^\mathsf{T}$ be the wait-and-see follower's decision vector in each scenario $k \in N$.

For a risk-neutral leader, who aims at maximizing the expected follower's traveling distance from s to t, the problem is well studied in the literature, and can be formulated as [see, e.g., 45]

$$\max_{x \in X} \sum_{k \in N} p_k \min_{y^k} \sum_{a \in A} (c_a^k + x_a d_a^k) y_a^k \tag{2a}$$

$$\text{s.t. } \sum_{a \in \delta^+(i)} y_a^k - \sum_{a \in \delta^-(i)} y_a^k = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{o.w.} \end{cases},\quad \forall i \in V,\ \forall k \in N, \tag{2b}$$

$$y_a^k \geq 0,\ \forall a \in A,\ \forall k \in N. \tag{2c}$$

Model (2) contains bilinear terms $(c_a^k + x_a d_a^k) y_a^k$. Denote ω_i^k as the dual variables associated with the flow balance constraints (2b) for each node $i \in V$ and scenario $k \in K$. Taking the dual of the inner minimization, model (2) is equivalent to

$$\max_{x, \omega} \sum_{k \in N} p_k \theta_k \tag{3a}$$

$$\text{s.t. } \theta_k \leq \omega_t^k,\ \forall k \in N, \tag{3b}$$

$$-\omega_i^k + \omega_j^k \leq c_a^k + d_a^k x_a, \quad \forall a = (i, j) \in A,\ \forall k \in N, \tag{3c}$$

$$\omega_s^k = 0,\ \forall k \in N, \tag{3d}$$

$$x \in X, \tag{3e}$$

where θ_k provides the shortest traveling distance of the follower in each scenario $k \in N$. A variety of algorithms have applied decomposition and efficient algorithms for solving (3) as a mixed integer linear programming (MILP) model with a large number of scenarios [e.g., 9, 17, 19, 28].

Our focus is to study a risk-averse leader, who aims at ensuring that the follower's shortest traveling distance from s to t exceeds a given length ϕ with high probability. We introduce binary variables z_k, $\forall k \in N$, such that $z_k = 1$ if the follower's shortest s-t path length in scenario k exceeds ϕ, and $z_k = 0$ otherwise. Let $Z := \{z \in \{0,1\}^{|N|} \mid \sum_{k \in N} p_k z_k \geq 1 - \epsilon\}$. Denote $y^k(x) = [y^k_a(x),\ a \in A]^\mathsf{T}$ as the follower's decision vector dependent on the leader's decision x. We present an MILP formulation of the RASPI with a wait-and-see follower as follows:

$$\min_{x,z}\ r^\top x \tag{4a}$$

$$\text{s.t.}\ \sum_{a \in A}(c^k_a + d^k_a x_a) y^k_a(x) \geq \phi z_k,\ \forall k \in N, \tag{4b}$$

$$z \in Z,\ x_a \in \{0,1\},\ \forall a \in A, \tag{4c}$$

$$\text{where}\quad y^k(x) \in \arg\min_y \sum_{a \in A}(c^k_a + d^k_a x_a) y_a \tag{4d}$$

$$\text{s.t.}\ \sum_{a \in \delta^+(i)} y_a - \sum_{a \in \delta^-(i)} y_a = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t\ , \\ 0 & \text{o.w.} \end{cases} \quad \forall i \in V, \tag{4e}$$

$$y_a \geq 0,\ \forall a \in A. \tag{4f}$$

The objective (4a) minimizes the leader's interdiction cost. Constraints (4b) are nonlinear, since the coefficient vector $[y^k_a(x), a \in A, k \in N]^\mathsf{T}$ is a function of x. They ensure that if $z_k = 1$, then the traveling distances of *all* s-t paths under the realization (c^k, d^k) need to be at least ϕ , because the length of the shortest s-t path, represented by solution $y^k(x) = [y^k_a(x),\ a \in A]^\mathsf{T}$, is at least ϕ.

2.2 Illustration of the RASPI with wait-and-see follower

We demonstrate the above RASPI setting using a small example, which focuses on the sensitivity of the leader's optimal interdiction solutions to values of the path-length threshold ϕ and the risk parameter ϵ. In Figure 2, the traveling path options from $s = 1$ to $t = 4$ are

- $\mathcal{P}_1$: $= \{(1,2),(2,4)\}$; $\mathcal{P}_2$: $= \{(1,2),(2,3),(3,4)\}$; $\mathcal{P}_3$: $= \{(1,3),(3,4)\}$.

Table 1: Costs $c^k_a,\ a \in A$ and the before-interdiction lengths $l^k_1,\ l^k_2,\ l^k_3$ of paths $\mathcal{P}_1,\ \mathcal{P}_2,\ \mathcal{P}_3$ in each scenario k

k	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	l^k_1	l^k_2	l^k_3
1	1	3	1	4	1	5	3	4
2	2	1	4	1	3	3	9	4
3	3	4	3	1	4	4	10	8
4	4	3	2	3	1	7	8	4

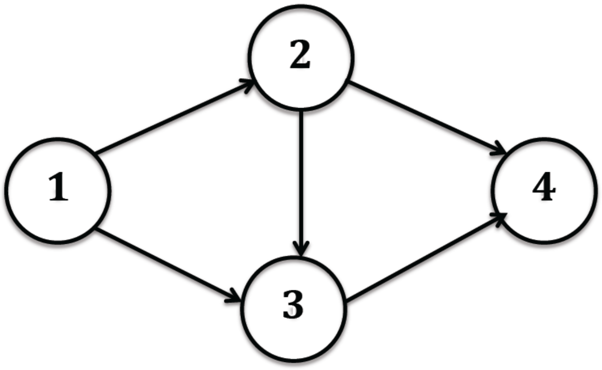

Figure 1: An Example Graph $G(V, A)$

We consider $|N| = 4$ scenarios and provide values of c_a^k for each arc $a \in A$, along with the respective lengths denoted as l_1^k, l_2^k, and l_3^k for each scenario k in Table 2. We assume unit cost of interdicting each arc (i.e., $r_a = 1, \ \forall a \in A$), and also unit increased distance for each arc in every scenario if the arc is interdicted (i.e., $d_a^k = 1, \ \forall a \in A, \ k = 1, \ldots, |N|$). The probability of each scenario k is $p_k = 0.25, \ \forall k = 1, \ldots, 4$. The leader aims to make the follower's traveling distance no less than $\phi = 4.75$ with probability at least $1 - \epsilon$. We vary $1 - \epsilon$ and demonstrate optimal interdiction solutions to model (4) as follows.

When no arcs are interdicted, a wait-and-see follower will travel on paths $\mathcal{P}_2$, $\mathcal{P}_1$, $\mathcal{P}_1$, and $\mathcal{P}_3$ in scenarios $k = 1, 2, 3, 4$, respectively. If $1 - \epsilon > 0.75$, we require the follower's shortest distances in all the scenarios to be no less than the threshold 4.75 (or 5 equivalently, given all integer distances in this example). Therefore, the leader needs to interdict (i) at least two arcs in path $\mathcal{P}_2$ and at least one arc in path $\mathcal{P}_3$ required by scenario $k = 1$; (ii) at least two arcs in path $\mathcal{P}_1$ and at least one arc in path $\mathcal{P}_3$ required by scenario $k = 2$; (iii) at least one arc in path $\mathcal{P}_1$ required by scenario $k = 3$; (iv) at least one arc in path $\mathcal{P}_3$ required by scenario $k = 4$. We take the union of all requirements from each scenario, and the leader needs to interdict at least two arcs in path $\mathcal{P}_1$, two arcs in path $\mathcal{P}_2$ and one arc in path $\mathcal{P}_3$. This can be achieved by interdicting arcs $(1, 2)$, $(3, 4)$, and $(2, 4)$ with the minimum total interdiction cost $= 3$. If $0.75 \geq 1 - \epsilon > 0.50$, following the same analysis above, the leader needs to interdict one arc in path $\mathcal{P}_1$, two arcs in path $\mathcal{P}_2$, and one arc in path $\mathcal{P}_3$. Thus, an optimal solution is to interdict the arcs $(1, 2), (3, 4)$ with the minimum interdiction cost $= 2$. If we continue decreasing $1 - \epsilon$ and let $0.50 \geq 1 - \epsilon > 0.25$, the follower's shortest traveling distances need to be no shorter than 5 in ≥ 2 scenarios. Thus, we interdict arcs $(1, 2)$ and $(3, 4)$ with the minimum interdiction cost $= 2$.

3 Solution Approaches

Note that in the above example, by increasing the interdiction budget, e.g., the total number of arcs that can be interdicted, we will surely increase the probability of the follower traveling on a sufficiently long path. Therefore, the corresponding RASPI model (4) involves a 0-1 covering structure with binary interdiction decisions $x_a,\ a \in A$. In this section, we use this observation and derive valid inequalities from the covering structure [cf. 31] that can be implemented in a branch-and-cut computational framework for solving model (4).

Denote $\mathcal{P}$ as the set of all s-t paths, whose size is exponential for a general graph G. Without loss of generality, we assume that for each scenario $k \in N$, there exists a path $P \in \mathcal{P}$ with $\sum_{a\in P}(c_a^k + d_a^k) \geq \phi$, and there exists a path $P \in \mathcal{P}$ with $\sum_{a\in P} c_a^k < \phi$. Otherwise, for the former, given any interdiction solution x, the shortest traveling distance between s and t is less than ϕ, and thus we have to enforce $z_k = 0$ for the corresponding scenario k; for the latter, the traveling distances of all s-t paths are at least ϕ regardless of the interdiction decision x, and therefore we can simply set $z_k = 1$ for the corresponding scenario k. Denote l_P^k as the length of path P in scenario k, for all $P \in \mathcal{P}$ and $k \in N$.

As mentioned in Section 2.1, the scenario-based constraint (4b) means that if $z_k = 1$, then the traveling distances of *all* s-t paths given the realization (c^k, d^k) need to be at least ϕ, and therefore, we reformulate model (4) as a path-based MILP model:

$$
\begin{aligned}
&\min_{x,z}\ r^\top x && \text{(5a)}\\
&\text{s.t.}\ \sum_{a\in P} d_a^k x_a \geq (\phi - l_P^k) z_k,\ \forall P \in \mathcal{P},\ \forall k \in N, && \text{(5b)}\\
&\qquad z \in Z,\ x_a \in \{0,1\},\ \forall a \in A, && \text{(5c)}
\end{aligned}
$$

which has exponential number of constraints (5b). Taking the dual of the shortest path problem involved in (5b) leads to a compact extended formulation of (5):

$$
\begin{aligned}
&\min_{x,z}\ r^\top x && \text{(6a)}\\
&\text{s.t.}\ \omega_t^k - \omega_s^k \geq \phi z_k,\ \forall k \in N, && \text{(6b)}\\
&\qquad \omega_j^k - \omega_i^k \leq c_a^k + d_a^k x_a,\ \forall (i,j) = a \in A,\ \forall k \in N, && \text{(6c)}\\
&\qquad \omega_s^k = 0,\ \forall k \in N, && \text{(6d)}\\
&\qquad z \in Z,\ x_a \in \{0,1\},\ \forall a \in A. && \text{(6e)}
\end{aligned}
$$

Although the above extended formulation is compact, it could be computationally intractable especially when the number $|N|$ of scenarios is large. On the other hand, note that constraints (5b) can be decomposed by scenario, and thus (5) can be solved in a Benders decomposition framework, using a branch-and-cut algorithm. Next, we solve a set of scenario-based subproblems to separate valid inequalities (5b), which can be cast as standard Benders cuts. Moreover, we exploit the covering structure of (5) by developing strong lifted pack inequalities. We present the details of the decomposition algorithm for solving (5) in the following sections.

3.1 Benders decomposition

We first present the solution approach for model (5) using the standard Benders decomposition approach implemented in a branch-and-cut algorithm. Denote $(\hat{x}, \hat{z})$ as the solution to a linear programming (LP) relaxation of model (5) at the current node in the branch-and-bound tree. We find a shortest path $P_k \in \mathcal{P}$ in each scenario k by using $(c_a^k \hat{z}_k + d_a^k \hat{x}_a)$ as the traveling cost on arc a, for all $a \in A$. If the length of this path P_k, $\sum_{a \in P_k}(c_a^k \hat{z}_k + d_a^k \hat{x}_a)$, is less than $\phi \hat{z}_k$, we can add the corresponding inequality (5b) into the relaxed master problem as a cut and continue the branch-and-cut procedure.

3.2 Scenario-based pack inequalities

We propose an additional set of valid inequalities to be generated iteratively in the branch-and-cut framework. These valid inequalities are motivated by the covering structure of model (5). The combinatorial structure within a chance-constrained setting has been analyzed in the context of a chance-constrained bin packing problem by Song et al. [39], who applied lifting procedures to strengthen the probabilistic cover inequalities in the space of the original x variables. However, different from a compact problem formulation (multi-dimensional knapsack problems) in their paper, we have *exponentially* many covering constraints (5b) for each scenario $k \in N$.

For each scenario $k \in N$, we define a subset of arcs $C_k \subseteq A$ as a *scenario-based pack* if there exists a path $P_k \in \mathcal{P}$ whose traveling cost is less than ϕ, even if we interdict all arcs in $A \setminus C_k$, i.e., $x_a = 1, \ \forall a \in A \setminus C_k$. This induces the following inequality, which we call *the scenario-based pack inequality*:

$$\sum_{a \in C_k} x_a \geq \psi(C_k) z_k,$$

where $\psi(C_k) \geq 1$ is the minimum number of arcs in C_k that need to be interdicted on path P_k so that the traveling distance of P_k is no less than ϕ. The value of $\psi(C_k)$ can be calculated by sorting the interdiction effects of arcs in $C_k \cap P_k$ in decreasing order, and then iteratively picking arcs to interdict until the threshold ϕ is achieved. The above inequality uses the combinatorial feature of the chance-constrained interdiction problem, and yields a pack-based MILP formulation:

$$\min_{x,z} \ r^\top x \tag{7a}$$

$$\text{s.t.} \ \sum_{a \in C_k} x_a \geq \psi(C_k) z_k, \ \forall C_k \in \mathcal{C}_k, \ \forall k \in N, \tag{7b}$$

$$\sum_{k \in N} p_k z_k \geq 1 - \epsilon, \tag{7c}$$

$$z_k \in [0,1], \ \forall k \in N, \ x_a \in \{0,1\}, \ \forall a \in A, \tag{7d}$$

where $\mathcal{C}_k$ is the set of all scenario-based packs for each $k \in N$. Note that the integrality requirement on variables z is relaxed in the reformulation (7) because: (i) formulation (7) remains an exact formulation when only the sets $C_k \in \mathcal{C}_k$ that correspond to $\psi(C_k) = 1$ are included; (ii) $\sum_{a \in C_k} x_a$

is integer since x's are binary variables; (iii) increasing a fractional $z_k \in (0, 1]$ to 1 does not violate (7b) or (7c) and does not increase the objective value of (7).

Similar to the Benders formulation (5), given a node relaxation solution from the current node in the branch-and-bound tree, we identify violated scenario-based pack inequalities (7b) and add them as cutting planes. The basic scenario-based pack inequalities (7b) may yield a weak relaxation bound. We next strengthen these inequalities through lifting.

3.3 Lifted scenario-based pack inequalities

For scenario k, a scenario-based pack C_k is *minimal* if $C_k \setminus \{a\}$ is no longer a pack for every $a \in C_k$. Constraints (7b) that correspond to minimal packs alone are sufficient to define the feasible region of model (7) in addition to constraints (7c) and (7d), in which $\psi(C_k) = 1$ for any minimal pack C_k based on the definition. The sufficiency comes from the fact that all integer infeasible solutions will be excluded by constraints (7b) with minimal scenario-based packs. Therefore, constraints (7b) can be replaced by the following minimal scenario-based pack inequalities:

$$\sum_{a \in C} x_a \geq z_k, \ \forall C \in \bar{\mathcal{C}}_k, \tag{8}$$

where $\bar{\mathcal{C}}_k$ is the set of all minimal packs in scenario $k \in N$. Next, we discuss how lifting can be applied in our setting to strengthen the minimal scenario-based pack inequalities (8).

3.3.1 Minimal scenario-based pack initialization

Recall that $(\hat{x}, \hat{z})$ is the optimal solution to the LP relaxation at the current branch-and-bound tree node. To construct an initial minimal scenario-based pack for any scenario $k \in N$, we start with the arc set $C := \{a \in A \mid \hat{x}_a = 0\}$, and sequentially add arcs to this set according to an increasing order of $\hat{x}_a$ values, until C becomes a scenario-based pack for scenario k. We choose the sequence in this heuristic way since we would like the resulting inequality (8) to be mostly violated by $(\hat{x}, \hat{z})$, as suggested by [14]. At each step, we check if the length of the shortest path when all arcs in C are interdicted is less than the target ϕ, to determine if C is a scenario-based pack. If yes, we make it minimal by sequentially checking if arcs from C could be removed. Specifically, all arcs in C that are not included in the shortest path P that we found in the previous step can be removed. Then, we sequentially remove arcs $a \in C$ to see if $C \setminus \{a\}$ remains a scenario-based pack. We note that the above procedure does not lead to exact separation of minimal scenario-based pack inequality, which is known to be NP-hard even for the deterministic 0-1 knapsack problems [13].

3.3.2 Scenario-based lifting

Lifting is one of the most effective procedures for deriving strong valid inequalities for 0-1 programming problems from inequalities that are valid for its lower dimensional restrictions [32, 4]. Lifting is usually applied sequentially: variables are lifted iteratively by solving a separate optimization problem for each variable to determine their lifting coefficients (see, e.g., [13, 14]). Sequence independent lifting has also been studied by, e.g., [15, 4], which exploit the superadditive structure of the lifting function. In this paper, we apply the idea of sequential lifting to strengthen the basic minimal scenario-based pack inequalities (8). Following the notation in [14], given a scenario-based pack C for scenario $k \in N$, we let $C_1 = \{a \in C \mid \hat{x}_a > 0\}$, $C_2 = \{a \in C \mid \hat{x}_a = 0\}$, $F = \{a \in A \setminus C \mid \hat{x}_a \in (0, 1)\}$, and $R = \{a \in A \setminus C \mid \hat{x}_a = 1\}$. By definition, $z_k = 1$ implies that:

$$\sum_{a \in C_1} x_a \geq 1, \tag{9}$$

when all $x_a, a \in C_2$ are fixed to 0, and $x_a, a \in F \cup R$ are fixed to 1. In the literature, lifting a variable that is fixed to 0 (variables in C_2) is called *uplifting*, and lifting a variable that is fixed to 1 (variables in $F \cup R$) is called *downlifting*. Note that we have to uplift variables $x_a, a \in C_2$ to ensure the validity of the inequality. On the other hand, downlifting variables that are fixed to 1 ($x_a, a \in F \cup R$) is only optional, since the inequality without downlifting is already valid. A stronger inequality may be obtained by the extra computational effort on performing downlifting. We investigate the effectiveness of lifting in our computational experiments.

Different valid inequalities may be obtained using different lifting sequences, which may lead to different computational performances. For example, instead of starting with (9) with $x_a, a \in C_2$ fixed to 0 and $x_a, a \in F \cup R$ fixed to 1, one could start with the minimal scenario-based pack inequality (8) with C, and just apply downlifting on $x_a, a \in F \cup R$ to strengthen this inequality. However, motivated by Gu et al. [14], we apply the following "default" lifting sequence: first downlift variables $x_a, a \in F$, then uplift variables $x_a, a \in C_2$, and at last, optionally downlift variables $x_a, a \in R$. The lifting sequence $\{\pi_1, \pi_2, \ldots, \pi_{|F|}\}$ within set F is obtained by sorting the relaxation solution values $\{\hat{x}_a\}_{a \in F}$ in an increasing order, and the lifting sequence within sets C_2 and R could just be the lexicographic order. This sequence has been shown to yield the best computational performance according to extensive computational studies performed in [14] on 0-1 knapsack instances.

To illustrate the idea of lifting, assume that we perform sequential downlifting on variables $x_a, a \in F$ following a lifting sequence $\{\pi_1, \pi_2, \ldots, \pi_{|F|}\}$, while variables $x_a, a \in C_2$ are fixed to 0, and $x_a, a \in R$ are fixed to 1. Suppose that we have completed downlifting for variables $x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_{t-1}}$, and the corresponding lifting coefficients $\beta_{\pi_1}, \beta_{\pi_2}, \ldots, \beta_{\pi_{t-1}}$ have been obtained. Next, we perform downlifting for variable x_{π_t}, for which the subproblem for calculating the lifting coefficient is given by:

$$\eta_{\pi_t} = \min \sum_{a \in C_1} x_a + \sum_{j=1}^{t-1} \beta_{\pi_j} x_{\pi_j} \tag{10a}$$

$$\text{s.t.} \sum_{a \in P} d_a^k x_a \geq \phi - l_P^k, \ \forall P \in \mathcal{P}, \tag{10b}$$

$$x_a = 0, \ \forall a \in C_2, x_{\pi_t} = 0, \tag{10c}$$

$$x_a = 1, \ \forall a \in F \setminus \{\pi_1, \pi_2, \ldots, \pi_t\}, \tag{10d}$$

$$x_a = 1,\ \forall a \in R, \tag{10e}$$

$$x_a \in \{0,1\},\ \forall a \in C_1 \cup \{\pi_1, \pi_2, \ldots, \pi_{t-1}\}. \tag{10f}$$

The (exact) lifting coefficient is $\beta_{\pi_t} := \max\{0, \eta_{\pi_t} - 1 - \sum_{j=1}^{t-1} \beta_{\pi_j}\}$. The exact lifting problem (10) is hard, however, a lower bound of the optimal objective of lifting problem (10) is sufficient to provide a valid lifting coefficient, which can be obtained, e.g., by solving the LP relaxation of (10). An alternative lower bound of (10) can be obtained by considering a single path P in (10b) rather than the entire set $\mathcal{P}$ of paths. Lifting based on a single path P is straightforward, since we can see it as the reverse of a single-row 0-1 knapsack problem, where we can apply the standard lifting technique for knapsack covering inequalities [14]. Clearly, there exists a computational tradeoff between the effort and effectiveness of lifting, which will be investigated in our computational study. The lifting problems for variables $x_a, a \in C_2$ and variables $x_a, a \in R$ can be formulated in a similar way as (10).

After lifting for the whole sequence, we obtain the following lifted scenario-based pack inequality:

$$\sum_{a \in C_1} x_a + \sum_{a \in F} \beta_a x_a + \sum_{a \in C_2} \gamma_a x_a \geq \left(1 + \sum_{a \in F} \beta_a\right) z_k, \tag{11}$$

which can then be added as a cut if violated by a relaxation solution $(\hat{x}, \hat{z})$.

3.4 Lifted probabilistic pack inequalities

Motivated by the compact formulation without scenario-based variables z, first proposed by [39] for chance-constrained binary packing problems, we define a subset of arcs $C \subseteq A$ as a *probabilistic pack*, if $P(F(C)) := \sum_{k \in F(C)} p_k > \epsilon$, where $F(C)$ is the set of scenarios where C is a scenario-based pack. By definition, model (5) can be reformulated by including all probabilistic pack inequalities:

$$\min \sum_{a \in A} c_a x_a \tag{12a}$$

$$\text{s.t.} \sum_{a \in C} x_a \geq 1,\ \forall C \in \mathcal{C}, \tag{12b}$$

$$x \in \{0,1\}^{|A|}, \tag{12c}$$

where $\mathcal{C}$ is the set of all the probabilistic packs. Based on the definition, a probabilistic pack C can be initiated by taking the union of sufficiently many scenario-based packs. Model (12) is a weak relaxation, but it can be strengthened by performing lifting for the probabilistic pack inequalities (12b), which can be applied in a similar way as the lifted scenario-based pack inequalities: suppose that we perform a sequential downlifting on variables $x_a,\ a \in F$ following a lifting sequence $\{\pi_1, \pi_2, \ldots, \pi_{|F|}\}$, while variables $x_a,\ a \in C_2$ are fixed to 0, and $x_a,\ a \in R$ are fixed to 1. (Recall that $F = \{a \in A \setminus C \mid \hat{x}_a \in (0,1)\}$, $C_2 = \{a \in C \mid \hat{x}_a = 0\}$, and $R = \{a \in A \setminus C \mid \hat{x}_a = 1\}$.) Given the lifting coefficients $\beta_{\pi_1}, \beta_{\pi_2}, \ldots, \beta_{\pi_{t-1}}$ for variables $x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_{t-1}}$, we formulate the problem for computing the coefficient of variable x_{π_t} after lifting as:

$$\begin{aligned}
\eta_{\pi_t} = \min \ & \sum_{a \in C_1} x_a + \sum_{j=1}^{t-1} \beta_{\pi_j} x_{\pi_j} && \text{(13a)}\\
\text{s.t.} \ & \sum_{a \in P} d_a^k x_a \geq (\phi - l_P^k) z_k, \ \forall P \in \mathcal{P}, \ \forall k \in N, && \text{(13b)}\\
& \sum_{k \in N} p_k z_k \geq 1 - \epsilon, && \text{(13c)}\\
& x_a = 0, \ \forall a \in C_2, \ x_{\pi_t} = 0, && \text{(13d)}\\
& x_a = 1, \ \forall a \in R \cup F \setminus \{\pi_1, \pi_2, \ldots, \pi_t\}, && \text{(13e)}\\
& x_a \in \{0,1\}, \ \forall a \in C_1 \cup \{\pi_1, \pi_2, \ldots, \pi_{t-1}\}, \ z_k \in \{0,1\}, \ \forall k \in N. && \text{(13f)}
\end{aligned}$$

The above lifting problem (13) can be seen as the original chance-constrained problem (5), with some variables being fixed according to constraints (13d) and (13e). Similar to model (5), exactly solving (13) for computing the lifting coefficients is difficult. Instead, we look for a lower bound for the optimal objective value of (13) that can be obtained more efficiently. Specifically, we apply the quantile-based bound [see, e.g., 39] to calculate the approximate lifting coefficients. First, we compute for each scenario

$$\eta(k) := \min \{ \sum_{a \in C_1} x_a + \sum_{j=1}^{t-1} \beta_{\pi_j} x_{\pi_j} \mid \sum_{a \in P_k} d_a^k x_a \geq (\phi - l_{P_k}^k), (13d) - (13f) \},$$

using a single path P_k. This path P_k in each scenario k is chosen as the shortest path identified in scenario k when initializing the probabilistic pack C. Then we sort $\{\eta(k)\}_{k \in N}$ in a nonincreasing order $\eta_{\sigma_1} \geq \eta_{\sigma_2} \geq \cdots \eta_{\sigma_{|N|}}$, and η_{σ_q} gives a valid lower bound for the optimal objective value of (13), where $q := \min\{v \mid \sum_{k=1}^{v} p_{\sigma_k} \geq 1 - \epsilon\}$. This lifting procedure leverages the efficient sequential lifting for a (single-row) 0-1 knapsack problem separately for each scenario. After lifting for the whole sequence, we obtain the following lifted probabilistic pack inequality:

$$\sum_{a \in C_1} x_a + \sum_{a \in F} \beta_a x_a + \sum_{a \in C_2} \gamma_a x_a \geq 1 + \sum_{a \in F} \beta_a. \tag{14}$$

3.5 Embedding the valid inequalities in a branch-and-cut framework

We summarize the proposed valid inequalities and show how they are embedded in a branch-and-cut framework in Algorithm 1. In our computational experiments, we investigate various combinations between different families of valid inequalities, and empirically study the effectiveness of each class of valid inequalities on various RASPI instances with wait-and-see followers.

Algorithm 1 A cut-generation subroutine for solving model (4).

```
Input: a relaxation solution $(\hat{x}, \hat{z})$, a threshold $\delta \in (0,1)$ to choose scenarios for cut generation.
for every $k \in N: \ \hat{z}_k > \delta$ do
    Separate inequality (5b) by solving a shortest path problem using $(c_a^k \hat{z}_k + d_a^k \hat{x}_a)$ as the cost for each arc $a \in A$.
    Initialize a scenario-based pack $C_k(\hat{x}) := \{a \in A \mid \hat{x}_a = 0\}$.
    if $C_k(\hat{x})$ is a scenario-based pack then
        for $L = 1$ to $\psi(C_k(\hat{x}))$ do
            Construct a minimal pack for each level $L$ by sequentially removing arcs, and add the corresponding
            unlifted scenario-based pack inequalities (7b) if it is violated by $(\hat{x}, \hat{z})$.
        end for
    else
        Construct a scenario-based pack $\bar{C}_k(\hat{x})$ and an associated shortest path $P_k$ by adding arcs following an
        increasing order of $\hat{x}_a$. Then make it minimal by sequentially removing arcs from $\bar{C}_k(\hat{x})$.
        Optionally, perform lifting based on this path $P_k$ or by solving the LP relaxation of the lifting problem (10),
        and add the lifted scenario-based pack inequality (11) if it is violated by $(\hat{x}, \hat{z})$.
    end if
end for
Initiate $C(\hat{x}) := \{a \in A \mid \hat{x}_a = 0\}$.
if $C(\hat{x})$ is a probabilistic pack then
    Make $C(\hat{x})$ minimal by sequentially removing arcs, and add the probabilistic pack inequality (14) if it is
    violated by $(\hat{x}, \hat{z})$.
else
    Construct a minimal probabilistic pack $C(\hat{x})$ by first adding arcs following an increasing order of $\hat{x}_a$, and then
    performing sequential reduction. Obtain an associated set of paths $\{P_k\}_{k \in N}$.
    Optionally, perform lifting based on the set of paths $\{P_k\}_{k \in N}$, and add the probabilistic lifted pack inequality
    (12b) if it is violated by $(\hat{x}, \hat{z})$.
end if
```

4 RASPI with Here-and-Now Follower

We consider the risk-averse shortest path interdiction (RASPI) problem for the case when the follower's decision has to be made *before* the observation of $\{c^k, d^k\}_{k \in N}$, i.e., she picks an s-t path without knowing the realizations of the traveling cost and random interdiction effect. We assume that the follower is risk-neutral, who minimizes the expected traveling distance between s and t. Equivalently, the follower optimizes a deterministic shortest path problem using the expected arc costs $\mathbb{E}[\tilde{c}]$ and the expected interdiction effects $\mathbb{E}[\tilde{d}]$ given any interdiction solution x. Meanwhile, the leader is risk-averse, who wants the probability that the path chosen by the follower exceeds a given threshold ϕ to be sufficiently high. Let $\bar{c} = [\bar{c}_a, \ a \in A]^\mathsf{T} = \mathbb{E}[\tilde{c}]$ and $\bar{d} = [\bar{d}_a, \ a \in A]^\mathsf{T} = \mathbb{E}[\tilde{d}]$ be the respective expected values of $\tilde{c}$ and $\tilde{d}$. The problem is formulated as a bilevel mixed-integer nonlinear program:

$$\begin{aligned}
\min_{x,z} \ & r^\top x \\
\text{s.t.} \ & \sum_{a \in A} (c_a^k + d_a^k x_a) y_a(x) \geq \phi z_k, \forall k \in N, && \text{(15a)} \\
& \sum_{k \in N} p_k z_k \geq 1 - \epsilon, && \text{(15b)} \\
& z_k \in \{0,1\}, \ \forall k \in N, \ x_a \in \{0,1\}, \ \forall a \in A, && \text{(15c)}
\end{aligned}$$

where

$$y(x) \in \arg\min_{y} \sum_{a\in A} (\bar{c}_a + \bar{d}_a x_a) y_a \tag{15d}$$

$$\text{s.t.} \sum_{a\in\delta^+(i)} y_a - \sum_{a\in\delta^-(i)} y_a = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{o.w.} \end{cases}, \ \forall i \in V, \tag{15e}$$

$$y_a \geq 0, \ \forall a \in A, \tag{15f}$$

where the follower's path-selection decision $y(x) = [y_a(x), \ a \in A]^\mathsf{T}$ is a function of the leader's interdiction decision x, which corresponds to the shortest path given traveling cost vector $[\bar{c}_a + \bar{d}_a x_a, \ a \in A]^\mathsf{T}$.

4.1 Illustrating the RASPI with here-and-now follower

We revisit Figure 2, of which recall that the traveling options from $s = 1$ to $t = 4$ are

- $\mathcal{P}_1 := \{(1,2),(2,4)\}$; $\mathcal{P}_2 := \{(1,2),(2,3),(3,4)\}$; $\mathcal{P}_3 := \{(1,3),(3,4)\}$.

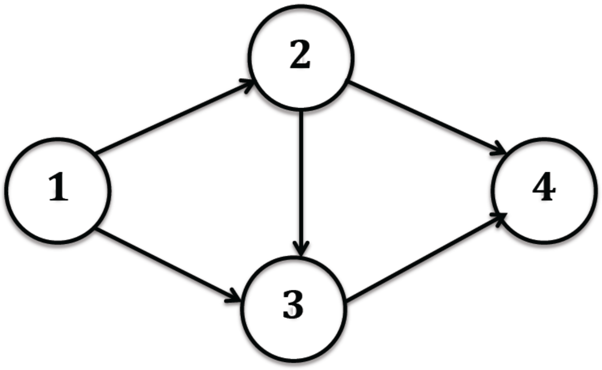

Figure 2: An example graph $G(V, A)$ with four nodes and five arcs.

Table 2 provides values of c_a^k for each arc $a \in A$, along with the respective lengths denoted as l_1^k, l_2^k, and l_3^k for each scenario $k = 1, \ldots, 4$. We assume unit cost of interdicting each arc (i.e., $r_a = 1, \ \forall a \in A$), and also unit increased distance for each arc in every scenario if the arc is interdicted (i.e., $d_a^k = 1, \ \forall a \in A, \ k = 1, \ldots, |N|$). The probability of each scenario k is $p_k = 0.25, \ \forall k = 1, \ldots, 4$. The leader aims to make the follower's traveling distance being no less

Table 2: Costs c_a^k, $a \in A$ and the before-interdiction lengths l_1^k, l_2^k, l_3^k of paths $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ in each scenario k

k	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	l_1^k	l_2^k	l_3^k
1	1	3	1	4	1	5	3	4
2	2	1	4	1	3	3	9	4
3	3	4	3	1	4	4	10	8
4	4	3	2	3	1	7	8	4

than $\phi = 4.75$ with probability at least $1-\epsilon$. Here, the follower is here-and-now and risk neutral in model (15), who picks the path that minimizes the expected traveling distance before the random arc costs and interdiction effects are realized. When no arcs are interdicted, the follower picks path $\mathcal{P}_1$ that has the shortest expected traveling distance, 4.75. For a risk-averse leader, without interdiction, the probability that the length of path $\mathcal{P}_1$ is ≥ 4.75 is 0.50. Therefore, the optimal solution will interdict no arcs if $1-\epsilon \leq 0.50$.

When $0.50 < 1-\epsilon \leq 0.75$, if the leader only interdicts arc $(1,2)$, the average lengths of path $\mathcal{P}_1$ and path $\mathcal{P}_2$ are both increased by 1, and become 5.75 and 8.50, respectively. Therefore, the follower will instead travel on path $\mathcal{P}_3$ which currently has an average traveling distance 5. However, the chance of the follower's traveling distance being ≥ 4.75, when the follower travels on $\mathcal{P}_3$, is only 0.25. This is a paradox that the leader's interdiction may lead to a lower probability that a here-and-now follower's traveling distance is sufficiently long. If the leader interdicts both arcs $(1,2)$ and $(3,4)$, the interdiction plan re-enforces the follower to take path $\mathcal{P}_1$, and with arc $(1,2)$ being interdicted, the follower's traveling distances are $6, 4, 5, 8$ in the four scenarios, resulting in the chance of a follower's distance≥ 4.75 being 0.75. If $1-\epsilon > 0.75$, we also need to interdict at least three arcs, e.g., arcs $(1,2)$, $(2,4)$, and $(3,4)$. Then the average lengths of paths $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$ are increased by 2, 2, and 1, becoming 6.75, 9.5, and 6, respectively. Thus, the follower picks path $\mathcal{P}_3$ and has travel distances 5, 5, 9, and 5 in the four scenarios. The chance of a follower's distance≥ 4.75 is 1.

4.2 Solution approaches

Shown by the above example, via interdicting more arcs, the leader does not necessarily achieve better interdiction result in terms of probabilistically prolonging the follower's traveling distance. The covering structure of the wait-and-see follower case does not apply here. We employ bilevel programming and linearization techniques, which is more computationally challenging.

The bilevel program (15) can be viewed as an extension of the shortest path interdiction problem with asymmetric information [see 6]. Given a relaxation solution $\hat{x}$, the follower's problem is a linear program with deterministic parameters. Let u_i, $\forall i \in V$ be the dual variables associated with the flow balance constraints (15e). We propose to reformulate (15) as a mixed-integer nonlinear programming model based on strong duality:

$$\min_{x,z,y,u} \ r^\top x$$

$$\text{s.t.} \ \sum_{a\in A}(c_a^k + x_a d_a^k) y_a \geq \phi z_k, \ \forall k \in N \tag{16a}$$

$$\sum_{a\in\delta^+(i)} y_a - \sum_{a\in\delta^-(i)} y_a = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{o.w.} \end{cases}, \ \forall i \in V \tag{16b}$$

$$\sum_{a\in A}(\bar{c}_a + x_a \bar{d}_a) y_a = u_s - u_t \tag{16c}$$

$$u_i - u_j \leq \bar{c}_a + x_a\bar{d}_a, \ \forall a = (i,j) \in A \tag{16d}$$

$$z \in Z, \ x_a \in \{0,1\}, \ y_a \geq 0, \ \forall a \in A. \tag{16e}$$

The bilinear terms $w_a \equiv x_a y_a$ in (16a) and (16c) can be linearized through the standard McCormick reformulation [26], because variables $x_a, \ a \in A$ are binary variables. This leads to the following mixed-integer linear programming (MILP) reformulation of the bilevel program (15):

$$\min_{x,z,y,u,w} \ r^\top x$$

$$\text{s.t.} \ \sum_{a\in A}(c_a^k y_a + d_a^k w_a) \geq \phi z_k, \ \forall k \in N \tag{17a}$$

$$\sum_{a\in A}(\bar{c}_a y_a + \bar{d}_a w_a) = u_s - u_t \tag{17b}$$

$$\text{(16b), (16d), (16e)}$$

$$w_a \leq y_a, \ \forall a \in A \tag{17c}$$

$$w_a \leq x_a, \ \forall a \in A \tag{17d}$$

$$w_a \geq x_a + y_a - 1, \ \forall a \in A \tag{17e}$$

$$w_a, y_a \geq 0, \ \forall a \in A. \tag{17f}$$

The McCormick inequities (17c)–(17f) ensure $w_q = 1$ if both $x_a = y_a = 1$, and $w_a = 0$ if $x_a = 0$ or $y_a = 0$. However, model (17) cannot be decomposed by scenario, and we solve it directly by using off-the-shelf MILP solvers.

5 Computational Results

In this section, we conduct computational experiments for the RASPI with wait-and-see followers. Without further specification, the risk aversion of the leader is modeled by a chance constraint with risk tolerance parameter $\epsilon = 0.1$ and 0.2. We also assume that all scenarios have the same probability, i.e., $p_k = 1/|N|, \ \forall k \in N$.

5.1 Experimental setup

We implement the proposed MILP models and branch-and-cut algorithms using commercial software CPLEX 12.5 in C++. We set the number of threads to be one. For the branch-and-cut algorithm with delayed constraint generation, we set the CPLEX `MIPEmphasis` parameter to "optimality" since CPLEX starts with an incomplete feasible set. We add the violated valid inequalities to CPLEX solver using `cutcallback` routines: for integral relaxation solutions we use `LazyConstraintCallback`, and for fractional relaxation solutions, we use `UserCutCallback`. CPLEX pre-solve is turned off when these routines are applied. We iteratively generate cuts at the root node as long as there exists any, and for other nodes in the branch-and-bound tree, we limit the cut generation to be at most one round. We use a cut violation threshold 10^{-3} for separating scenario-based cuts, and we use 5×10^{-4} for separating probabilistic pack inequalities. We look for violated scenario-based pack inequalities for scenarios with $\hat{z}_k > \delta = 10^{-3}$. CPLEX default settings are used for all the computational parameters unless specified later. We use the graph library LEMON [see 21] for solving shortest path subproblems. All tests are performed in a Linux workstation with four 2.50 GHz processors (although only one thread is used) and 3.6 GB memory.

First, we test randomly generated instances of $n \times n$ grid networks, for $n = 5, 6, 7, 8$, which are commonly used in the previously reviewed network interdiction literature. We assume that the network is directed, and for each edge $[i, j]$ we have arcs in both directions (i, j) and (j, i). We generate the costs of arcs randomly according to a uniform distribution between 1 and 10. We generate the interdiction effects in two stages. At the first stage, we take a random sample uniformly distributed between 0 and 1, and the interdiction is a "failure", i.e., the interdiction effect is 0, if the realization of the sample is less than 0.25. For a "successful" interdiction, we sample its effect randomly according to a uniform distribution between 1 and 5. The interdiction cost r_a for each arc a is generated by sampling from a uniform distribution between 1 and 5, which is then rounded to the nearest integer value. We set the threshold distance $\phi = 0.8 \times L$ for all the grid network instances unless specified otherwise, where L is the length of the shortest path using the expected arc cost $\mathbb{E}[\tilde{c}_a]$ for each arc $a \in A$.

Besides, we generate test instances based on two real-world network structures. Specifically, we use the Sioux-Falls road network [5] and the Russian railway network from Schrijver [35], containing 24 nodes, 76 links and 44 nodes, 200 links, respectively. Note that the two real transportation networks are about the same sizes of 5×5 and 7×7 grid networks, respectively. The threshold distance ϕ is chosen as $1.0 \times L$ for the Sioux-Falls instances, and $0.8 \times L$ for the Russian railway instances. For both networks, the random interdiction costs are generated in the same way as the grid network instances described above. The arc length c_a^k for each arc a in each scenario k is sampled independently from a normal distribution with mean value being the nominal arc length $\bar{c}_a$, and standard deviation being $\frac{1}{3}\bar{c}_a$. The interdiction effects are generated in the same two-stage approach described above for the grid network: (i) in the first stage, we use the same way to determine if an interdiction is "successful" or not; (ii) in the second stage, the interdiction effect $d_a^k = \frac{1}{3}\tilde{d}_a^k$, where $\tilde{d}_a^k$ is sampled in the same way that the arc lengths c_a^k are sampled. Note that the Russian railway network originally used for testing maximum-flow problem instances, where the arc distances are not given in Schrijver [35]. We manually measure the nominal distance values according to the map provided and use the estimated results.

For each instance, we test five replications and report the average results. The following abbreviations are used throughout this section:

- AvT: Average computational time (in seconds).
- AvN: Average number of branch-and-bound nodes processed.
- AvR: Average root optimality gap, where the root optimality gap for each instance is calculated as $(S^* - S^R)/S^*$, where S^* is the optimal objective value, and S^R is the lower bound provided at the root node after it is processed.

We set a time limit of 3600 seconds for all our experiments. If not all the five replications are solved to optimality within the time limit, under AvT result columns, we show in parentheses the number of instances solved within the time limit out of the five replications, and the average optimality gap among the remaining instances. Also, under AvN result columns, we show the average number of nodes processed up to the time limit, with the "$\geq$" indicating that the calculated average is a lower bound in this case.

5.2 A benchmark heuristic approach for model (4)

Given an integer interdiction solution $\bar{x} \in \{0,1\}^{|A|}$, we apply a heuristic based on $\bar{x}$ to find a feasible interdiction solution to model (4) as follows. First, we modify the arc cost as $c_a^k + d_a^k \bar{x}_a$ for all arcs $a \in A$ in each scenario $k = 1, \ldots, |N|$. We solve a shortest path problem in each scenario k and obtain the corresponding shortest path lengths $l_1^*, \ldots, l_{|N|}^*$. We sort the shortest path lengths in each scenario as $l_{\sigma_1}^* \leq \cdots \leq l_{\sigma_{|N|}}^*$, and identify a threshold scenario σ_{k^*} satisfying $\sum_{k=k^*}^{|N|} p_{\sigma_k} \geq 1-\epsilon$ and $\sum_{k=k^*+1}^{|N|} p_{\sigma_k} < 1-\epsilon$. (When all $p_k = 1/|N|$, the threshold $\sigma_{k^*} = \sigma_{\lceil |N|(1-\epsilon) \rceil}$.) If $\phi \leq l_{\sigma_{k^*}}^*$, the given solution $\bar{x}$ is feasible and we exit the algorithm. Otherwise, we denote $P_{\sigma_{\bar{k}}}$ as the shortest path in scenario $\sigma_{\bar{k}}$, where $\sigma_{\bar{k}}$ corresponds to the largest $l_{\sigma_k}^* < \phi$. We update solution $\bar{x}$ by sequentially setting arcs $a \in \{a \in P_{\sigma_{\bar{k}}} \mid \bar{x}_a = 0\}$ to 1 according to an increasing order of the interdiction cost r_a until the length of the path is at least ϕ. The algorithm will converge to a feasible solution because at least one extra scenario is forced to be feasible at every iteration. This heuristic is used to provide a starting feasible solution to CPLEX (`MIPstart`) by setting $\bar{x}_a = 0$, and used as a routine to generate heuristic solutions at each node by letting $\bar{x}_a = 1$ if $\hat{x}_a = 1$, and $\bar{x}_a = 0$ otherwise, where $\hat{x}_a$ is the current node relaxation solution.

5.3 Results of the RASPI with a wait-and-see follower

We first present the benefit of combinatorial information that we exploit by formulation (7) ("Pack-based"), compared to the "standard" Benders formulation (5) ("Benders"), the extended formulation (6) ("Extended"), as well as an exhaustive enumeration approach ("Enumeration"). In option "Pack-based", we only generate scenario-based pack inequalities (7b).

In Table 3, we can see that the exhaustive enumeration approach spends a long time searching for the optimal solution, even on these relatively small instances. The extended formulation (6) and Benders decomposition (5) give similar computational results, and neither of them are competitive compared to the pack-based reformulation (7). In particular, due to the large number of the Benders cuts generated at each iteration, Benders decomposition consumes a large amount of memory, especially for instances with a large number of scenarios. Therefore, we will focus next on how the pack-based reformulation can be solved more efficiently using the lifting technique.

Next, we compare the computational performances for different variants of the branch-and-cut algorithm for solving model (7). In option "No lifting", we only focus on the scenario-based minimal pack inequalities, and do not perform lifting. In option "Single path-based lifting" and option "LP-based lifting", we apply the "default" sequence of lifting using a single path identified

Table 3: Average computational time for the simple enumeration method, average computational time and number of nodes explored in the branch-and-bound procedure for solving the extended formulation (6), Benders formulation (5), and pack-based reformulation (7) using the best computational setting.

Instances			Enumeration	Extended		Benders		Pack-based	
Instance	ϵ	N	AvT	AvT	AvN	AvT	AvN	AvT	AvN
5×5	0.1	100	-†	15.9	1738	15.6	45K	0.2	56
$\lvert V\rvert = 25$, $\lvert A\rvert = 80$		1000	-	22%(0)	>18K	30.5%(0)	>607K	5.8	189
	0.2	100	-	25.4	2181	72.6	178K	0.3	88
		1000	-	34%(0)	>16K	M‡	M	20.9	554
SiouxFalls	0.1	1000	1508.7	27.1	539	1.81	411	0.2	0
$\lvert V\rvert = 24$, $\lvert A\rvert = 76$		3000	>2694.6(2)♯	299.1	1586	12.2	834	1.2	1
	0.2	1000	40.7	46.1	1840	5.9	1435	0.6	11
		3000	151.1	932.1	6697	53.0	4388	3.1	11

†: "-" indicates that the method hits the time limit in all instances

‡: "M" indicates that the method hits the memory limit in all instances

♯: only two instances are solved within 3600s, and 3600s is used in the calculation of AvT for others

by the minimal scenario-based pack, and using the LP relaxation of the lifting problem (10), respectively. During our computational study, we observed that including the Benders cuts (5b) made the computational performance worse. Therefore, we exclude the Benders cut generation from Algorithm 1.

Table 4: Average computational time (optimality gap) and average number of nodes explored in the branch-and-bound procedure for solving the pack-based reformulation (7) without lifting, using the single path-based lifting, and using LP-based lifting.

Instances			No lifting			Single path-based lifting			LP-based lifting		
Ins	ϵ	N	AvT	AvN	AvR	AvT	AvN	AvR	AvT	AvN	AvR
6×6	0.1	100	1.3	254	20.1%	1.5	172	17.4%	31.0	187	16.3%
$\lvert V\rvert = 36$		500	26.0	1388	23.0%	26.7	1074	20.9%	254.5	1102	19.6%
$\lvert A\rvert = 120$		1000	88.6	2442	25.2%	79.4	2027	22.9%	727.9	2096	22.0%
	0.2	100	1.8	376	25.5%	2.6	278	23.3%	29.9	269	22.7%
		500	117.6	6577	32.8%	88.1	4471	30.5%	369.5	3995	30.1%
		1000	946.1	18K	34.2%	915.4	12K	32.4%	0.9%(4)	>10K	31.7%
7×7	0.1	100	77.8	9099	24.1%	39.2	4729	19.3%	131.1	4138	18.9%
$\lvert V\rvert = 49$		500	4.3%(4)	>49K	30.0%	1.4%(4)	>26K	26.7%	1.8%(4)	>27K	25.2%
$\lvert A\rvert = 168$		1000	4.9%(2)	>35K	30.7%	2.9%(3)	>24K	27.4%	5.8%(2)	>16K	26.5%
	0.2	100	102.5	9561	32.1%	118.0	8750	28.4%	182.3	6664	26.6%
		500	3.6%(0)	>41K	37.0%	2.5%(2)	>36K	34.5%	2.5%(1)	>32K	33.4%
		1000	11.7%(0)	>27K	39.9%	8.2%(0)	>18K	36.8%	9.9%(0)	>13K	35.5%
8×8	0.1	100	1222.6	61K	30.0%	448.9	20K	25.4%	547.6	16K	22.4%
$\lvert V\rvert = 64$		500	12.4%(0)	>53K	34.0%	8.6%(0)	>38K	30.1%	9.4%(0)	>25K	27.8%
$\lvert A\rvert = 224$		1000	18.5%(0)	>28K	36.3%	15.0%(0)	>22K	31.8%	18.4%(0)	>6184	30.9%
	0.2	100	5.0%(4)	>60K	35.3%	784.1	25K	31.5%	1133.3	28K	30.6%
		500	20.4%(0)	>33K	42.7%	16.0%(0)	>25K	39.1%	16.6%(0)	>18K	37.9%
		1000	28.4%(0)	>18K	46.4%	23.5%(0)	>13K	42.3%	27.0%(0)	>5722	41.7%
Russian	0.1	100	0.7	51	18.5%	0.8	48	16.7%	34.6	39	16.3%
$\lvert V\rvert = 44$		1000	80.6	741	32.3%	51.2	508	30.6%	1326.5	563	29.5%
$\lvert A\rvert = 200$		3000	459.5	1341	31.9%	340.5	1006	30.1%	26.1%(1)	>297	35.9%
	0.2	100	1.0	74	25.4%	0.8	63	23.4%	28.0	67	23.3%
		1000	26.2	234	28.5%	19.8	165	28.1%	424.9	153	27.9%
		3000	209.0	487	33.7%	185.3	400	32.7%	7.0%(4)	>312	32.3%

From Table 4 we see that lifting helps to improve the relaxation bounds significantly, and in particular for larger instances. Comparing the two different lifting strategies, we see that the LP-based lifting gives better root relaxation bounds, but lifting based on a single s-t path gives much less overall computational time than LP-based lifting. This is because lifting procedure of LP-based lifting takes a significant amount of time from iteratively solving the LPs, whereas lifting based on a single s-t path can be done as efficient as lifting for a single-row 0-1 knapsack problem. We also observe that even if LP-based lifting is applied, the root relaxation gap is still very large, which leads to a large number of nodes to explore in the branch-and-bound tree. This motivates further investigation on other valid inequalities that exploit the problem structure.

We next show the effect of using probabilistic pack inequalities in addition to the scenario-based lifted pack inequalities. We use the best option "Single path-based lifting" in Table 4 as a benchmark for comparison. Given the observation that the LP-based lifting is time consuming, we apply the single path-based lifting for generating lifted probabilistic pack inequalities.

Table 5: Average computational time (optimality gap) and average number of nodes explored in the branch-and-bound procedure for solving the pack-based reformulation using the single path-based lifting, with and without probabilistic pack inequalities (12b).

Instances			No-prob			With-prob		
Ins	ϵ	N	AvT	AvN	AvR	AvT	AvN	AvR
6×6	0.1	100	1.5	172	17.4%	1.4	197	17.0%
$\|V\| = 36$		500	26.7	1074	20.9%	47.5	1102	20.9%
$\|A\| = 120$		1000	79.4	2027	22.9%	184.4	1745	22.8%
	0.2	100	2.6	278	23.3%	1.9	278	23.3%
		500	88.1	4471	30.5%	147.8	4514	30.6%
		1000	915.4	12K	32.4%	1.8%(4)	>8728	32.3%
7×7	0.1	100	39.2	4729	19.3%	41.1	3879	19.7%
$\|V\| = 49$		500	1.4%(4)	>26K	26.7%	2.7%(3)	>22K	26.8%
$\|A\| = 168$		1000	2.9%(3)	>24K	27.4%	6.7%(1)	>9075	27.8%
	0.2	100	118.0	8750	28.4%	125.0	8657	28.5%
		500	2.5%(2)	>36K	34.5%	4.3%(0)	>23K	34.6%
		1000	8.2%(0)	>18K	36.8%	12.6%(0)	>7871	36.6%
8×8	0.1	100	448.9	20K	25.4%	377.8	16K	25.2%
$\|V\| = 64$		500	8.6%(0)	>38K	30.1%	10.1%(0)	>20K	30.1%
$\|A\| = 224$		1000	15.0%(0)	>22K	31.8%	19.3%(0)	>5308	32.4%
	0.2	100	784.1	25K	31.5%	875.1	26K	31.5%
		500	16.0%(0)	>25K	39.1%	17.5%(0)	>15K	39.6%
		1000	23.5%(0)	>13K	42.3%	29.1%(0)	>4650	44.5%
Russian	0.1	100	0.8	48	16.7%	0.9	48	16.7%
$\|V\| = 44$		1000	51.2	508	30.6%	163.0	578	30.4%
$\|A\| = 200$		3000	340.5	1006	30.1%	1869.6	925	30.3%
	0.2	100	0.8	63	23.4%	0.9	63.4	23.4%
		1000	19.8	165	28.1%	64.3	212	28.1%
		3000	185.3	400	32.7%	728.0	397	32.9%

We see from Table 5 that the additional probabilistic pack inequalities (12b) do not help improve the root relaxation bound for most of our test instances. The extra time spent on generating these inequalities does not pay off. We conclude that these inequalities are not as effective as what has been discovered in [39]. The reason is that, the approximate lifting gives very weak lifting coefficients because only one path (constraint) is used for the entire lifting, while there are exponentially many paths (constraints). On the other hand, it is too time-consuming to consider all possible paths, e.g., in an LP-based lifting. This motivates further study on how lifting should be handled for lifted probabilistic pack inequalities that balances its effectiveness and effort.

In all, based on the test results, the scenario-based pack inequality formulation (7) with the additional lifted scenario-based pack inequalities (11) using the approximate lifting based on a single path yields the best computational performance overall. We can also see in Tables 3, 4 and 5 that the proposed approaches yield better computational performance on the two real-world instances – Sioux-Falls and Russian railway network instances than the randomly generated grid network instances. The behaviors of different computational options are consistent over both types of instances.

5.4 Sensitivity analysis

We analyze the solution sensitivity of the RASPI in terms of the optimal interdiction cost and risk tolerance parameter ϵ, using three different levels of distance threshold ϕ: $0.8L$, $0.85L$, and $0.9L$. In Figure 3 and Figure 4 we show the scatter plot of the RASPI solutions with various parameter settings, using one instance from the Russian railway network, and one randomly generated 6×6 grid network instance, respectively. Both instances have 1000 scenarios.

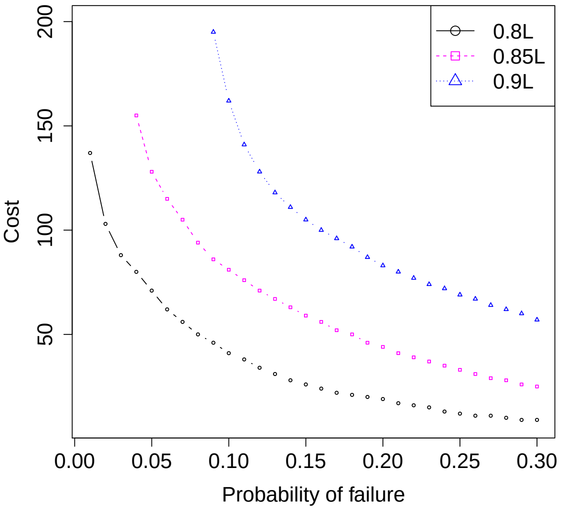

Figure 3: Scatter plot of the RASPI solutions using varying ϵ values and three levels of distance threshold for a Russian railway network instance.

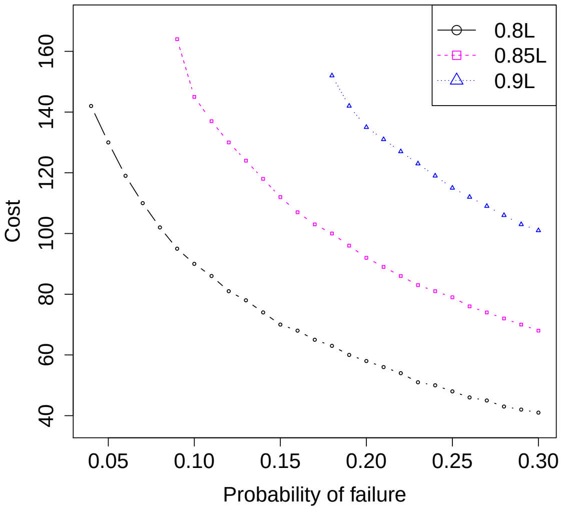

Figure 4: Scatter plot of the RASPI solutions using varying ϵ values and three levels of distance threshold for a randomly generated 6×6 grid network instance.

Figure 3 and Figure 4 demonstrate that the proposed RASPI model has the potential to yield a wide variety of solutions on the efficient frontier of interdiction cost and risk by varying the risk tolerance parameter ϵ. Decision maker can also pick the appropriate distance threshold ϕ given the interdiction budget and her risk tolerance according to these solutions.

We illustrate the solution patterns with different values of the risk tolerance ϵ and the distance threshold ϕ for the 6×6 grid network and the Russian railway network in Appendix A. As we increase the threshold ϕ, the interdicted arcs increase in an almost nested pattern for the grid network, but not necessarily show a nested pattern for the real-world transportation network. In fact, solutions with $\phi = 0.8L, 0.85L, 0.9L$ interdict very different sets of arcs at optimum in the Russian railway network.

5.5 Results of RASPI with a here-and-now follower

We set up grid-network instances and their parameters following the same procedures as before, to test the MILP formulation (17) for the RASPI with here-and-now risk-neutral follower. We show the preliminary results in Table 6.

Table 6: Average computational time (optimality gap) and average number of nodes explored in the branch-and-bound procedure for solving the MILP reformulation (17)

Instances		$\epsilon = 0.05$		$\epsilon = 0.1$	
Ins	N	AvT	AvN	AvT	AvN
4×4	100	91.5	135K	105.8	168K
	500	69.8%(0)	>561K	72.4%(0)	>602K
5×5	100	45.8%(1)	>3008K	26.0%(4)	>1124K
	500	89.1%(0)	>644K	79.9%(0)	>765K
6×6	100	40.4%(0)	>3113K	25.3%(4)	>1369K
	500	91.3%(0)	>525K	87.7%(0)	>611K

From Table 6, we can see that the MILP formulation (17) is very challenging to solve on our test instances. Most of the instances cannot be solved within the time limit. Even for the smallest test instances, not all of them can be solved to optimality. A large optimality gap remains after one-hour of computational time, and a huge number of branch-and-bound nodes are explored because of the weak relaxation bound and the large number of binary variables as well as extra constraints used to model the bilinear term $x_a y_a$ in (16). Moreover, there is no obvious decomposition structure in formulation (16), as the variables $\{y_a\}_{a \in A}$ are linked together in all the scenarios. The preliminary results above motivate future investigation on algorithm development for this class of problems, for which the details are presented in the main body of the paper.

6 Conclusions

In this paper, we have studied the stochastic shortest path interdiction problem with a risk averse interdictor. The risk aversion of the interdictor is characterized by a chance constraint. There are two different cases of the problem, depending on whether the follower's decision is made after (wait-and-see) or before (here-and-now) the realization of uncertainty. We focused on the wait-and-see case, where we have proposed a branch-and-cut algorithm according to a path-based reformulation of the model. We applied lifting technique to exploit the combinatorial structure of the problem. Moreover, we described a benchmark MILP formulation for the here-and-now case. We demonstrated the efficacy of our approaches, solution patterns, and result sensitivity by testing various network instances.

For future research, we will continue tackling the RASPI with a here-and-now follower with diverse risk behavior. In particular, we are interested in both risk-neutral and risk-averse behavior of the follower arising in different application contexts. The former type of follower aims at minimizing the expected travel distance as studied in Section 4 and the latter will only make an attempt if she has sufficient confidence of traveling on a sufficiently short path. The resulting formulations are bilevel mixed-integer nonlinear programs, for which we will investigate reformulation techniques and efficient decomposition algorithms. We will also investigate more valid inequalities to better exploit the RASPI structure in this paper, including strategies that balance between the effect and efficiency of the lifting procedure.

Acknowledgement. Dr. Shen acknowledges partial support by the National Science Foundation under grant grant CMMI-1433066. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Ahmed, S. and Shapiro, A. (2008). Solving chance-constrained stochastic programs via sampling and integer programming. In Chen, L. and Raghavan, S., editors, *Tutorials in Operations Research: State-of-the-Art Decision-Making Tools in the Information-Intensive Age*, pages 261–268. INFORMS.

[2] Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall.

[3] Alderson, D., Brown, G., and Carlyle, W. (2014). Assessing and improving operational resilience of critical infrastructures and other systems. In *Tutorials in Operations Research: Bridging Data and Decisions*, pages 180–215. INFORMS.

[4] Atamtürk, A. (2004). Sequence independent lifting for mixed-integer programming. *Operations Research*, 52(3):487–490.

[5] Bar-Gera, H. (2009). Transportation test problems. `http://www.bgu.ac.il/~bargera/tntp/`.

[6] Bayrak, H. and Bailey, M. D. (2008). Shortest path network interdiction with asymmetric information. *Networks*, 52(3):133–140.

[7] Beigy, H. and Meybodi, M. (2006). Utilizing distributed learning automata to solve stochastic shortest path problems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14(05):591–615.

[8] Collado, R. (2012). Stochastic network interdiction. Technical report, RUTCOR Technical Report RRR 5-2012, Rutgers Center for Operations Research, Rutgers University, 640 Bartholomew Road Piscataway, New Jersey.

[9] Cormican, K., Morton, D., and Wood, R. (1998). Stochastic network interdiction. *Operations Research*, 46(2):184–197.

[10] Dimitrov, N. and Morton, D. (2013). Interdiction models and applications. In *Handbook of Operations Research for Homeland Security*, pages 73–103. Springer.

[11] Frank, H. (1969). Shortest paths in probabilistic graphs. *Operations Research*, 17(4):583–599.

[12] Golden, B. (1978). A problem in network interdiction. *Naval Research Logistics Quarterly*, 25(4):711–713.

[13] Gu, Z., Nemhauser, G., and Savelsbergh, M. (1998a). Lifted cover inequalities for 0-1 integer programs: Complexity. *INFORMS Journal on Computing*, 11:117–123.

[14] Gu, Z., Nemhauser, G., and Savelsbergh, M. (1998b). Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10(4):427–437.

[15] Gu, Z., Nemhauser, G., and Savelsbergh, M. (2000). Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization*, 4(1):109–129.

[16] Held, H., Hemmecke, R., and Woodruff, D. (2005). A decomposition algorithm applied to planning the interdiction of stochastic networks. *Naval Research Logistics*, 52(4):321–328.

[17] Hemmecke, R., Schultz, R., and Woodruff, D. L. (2002). Interdicting stochastic networks. In Woodruff, D. L., editor, *Network Interdiction and Stochastic Integer Programming*, pages 71–80. Kluwer Academic Publishers, Boston, MA.

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

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

[20] Küçükyavuz, S. (2012). On mixing sets arising in chance-constrained programming. *Mathematical Programming*, 132(1–2):31–56.

[21] LEMON (2009). LEMON – Library for Efficient Modeling and Optimization in Networks. `http://lemon.cs.elte.hu/`.

[22] Lim, C. and Smith, J. (2007). Algorithms for discrete and continuous multicommodity flow network interdiction problems. *IIE Transactions*, 39(1):15–26.

[23] Luedtke, J. (2014). A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, 146:219–244.

[24] Luedtke, J., Ahmed, S., and Nemhauser, G. (2010). An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, 122(2):247–272.

[25] Martin, J. (1965). Distribution of the time through a directed, acyclic network. *Operations Research*, 13(1):46–66.

[26] McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Mathematical Programming*, 10(1):147–175.

[27] Miller-Hooks, E. (2001). Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1):35–52.

[28] Morton, D. (2010). Stochastic network interdiction. In Cochran, J. J., editor, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc.

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

[30] Murray, A. T., Matisziw, T. C., and Grubesic, T. H. (2007). Critical network infrastructure analysis: Interdiction and system flow. *Journal of Geographical Systems*, 9(2):103–117.

[31] Nemhauser, G. and Wolsey, L. (1999). *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY.

[32] Padberg, M. (1975). A note on zero-one programming. *Operations Research*, 23:833–837.

[33] Pan, F., Charlton, D., Morton, D., and Smith, J. (2007). Interdicting nuclear smuggling on a bipartite network. Technical report, Department of Mechanical Engineering, The University of Texas, Austin, TX.

[34] Polychronopoulos, G. and Tsitsiklis, J. (1996). Stochastic shortest path problems with recourse. *Networks*, 27(2):133–143.

[35] Schrijver, A. (2002). On the history of the transportation and maximum flow problems. *Mathematical Programming*, 91(3):437–445.

[36] Shen, S. (2011). *Reformulation and Cutting-Plane Approaches for Solving Two-Stage Optimization and Network Interdiction Problems.* PhD thesis, University of Florida, Gainesville, FL.

[37] Smith, J., Lim, C., and Alptekinoglu, A. (2009). Optimal mixed-integer programming and heuristic methods for a bilevel Stackelberg product introduction game. *Naval Research Logistics*, 56(8):714–729.

[38] Song, Y. and Luedtke, J. (2013). Branch-and-cut approaches for chance-constrained formulations of reliable network design problems. *Mathematical Programming Computation*, 5(4):397–432.

[39] Song, Y., Luedtke, J., and Küçükyavuz, S. (2014). Chance-constrained binary packing problems. *INFORMS Journal on Computing*, 26(4):735–747.

[40] Song, Y. and Zhang, M. (2015). Chance-constrained multi-terminal network design problems. *Naval Research Logistics*, 62:321–334.

[41] Waller, T. and Ziliaskopoulos, A. (2002). On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227.

[42] Washburn, A. and Wood, R. (1995). Two-person zero-sum games for network interdiction. *Operations Research*, 43(2):243–251.

[43] Wollmer, R. (1964). Removing arcs from a network. *Operations Research*, 12(6):934–940.

[44] Wood, R. (2010). Bilevel network interdiction models: Formulations and solutions. *Networks*, 174:175.

[45] Woodruff, D. L. (2003). *Network Interdiction and Stochastic Integer Programming.* Springer.

Appendix

A Example Solutions of the RASPI with a Wait-and-See Follower

In this section, we present two example solutions for solving the RASPI with a wait-and-see follower, on a randomly generated grid network instance and a real-world network instance. First, we show the optimal interdiction solutions for using threshold values $\phi = 0.8L, 0.85L, 0.9L$. Recall that L is the length of the shortest path using the expected arc cost $\bar{c}_a$ for each arc $a \in A$. Figures 5, 6, 7 respectively illustrate example solutions of $\phi = 0.8L, 0.85L, 0.9L$ in a 6×6 grid network with nodes numbered from 0 to 35. The follower seeks a shortest path from Node 0 to Node 35, while the leader's goal is to guarantee that the expected path length is no less than ϕ with at least 80% probability (i.e., $\epsilon = 0.2$). The dashed lines in the figures represent the remaining arcs after interdiction, which for $\phi = 0.9L$ are mostly in the two corners of the grid that do not involve the source and destination nodes. We specify the interdicted arcs in each set of solutions under different ϕ-values as follows.

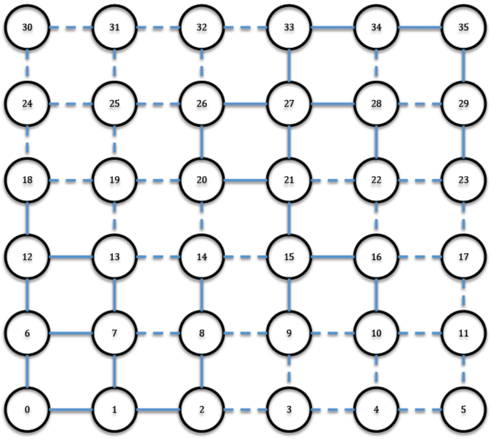

Figure 5: Example optimal arc-interdiction solution for $\phi = 0.8L$ in a 6×6 grid network with dashed lines representing the remaining arcs after interdiction.

Interdicted arcs for $\phi = 0.8L$: [0,1] [0,6] [1,2] [1,7] [2,8] [6,7] [6,12] [7,13] [8,14] [9,15] [10,16] **[12,13]** [12,18] [15,16] [15,21] [20,21] [20,26] [21,27] [22,28] [23,29] [26,27] [27,28] [27,33] [29,35] [33,34] [34,35]

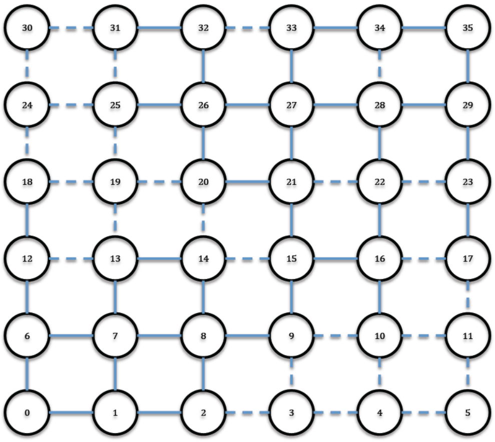

Figure 6: Example optimal arc-interdiction solution for $\phi = 0.85L$ in a 6×6 grid network with dashed lines representing the remaining arcs after interdiction.

Interdicted arcs for $\phi = 0.85L$: [0,1] [0,6] [1,2] [1,7] **[2,3]** [2,8] [6,7] [6,12] **[7,8]** [7,13] **[8,9]** [8,14] [9,15] [10,16] [12,18] **[13,14]** [15,16] [15,21] **[16,22]** **[17,23]** [20,21] [20,26] [21,22] [21,27] [22,28] [23,29] **[25,26]** [26,27] **[26,32]** [27,28] [27,33] **[28,29]** **[28,34]** [29,35] **[31,32]** [33,34] [34,35]

Interdicted arcs for $\phi = 0.9L$: [0,1] [0,6] [1,2] [1,7] [2,3] [2,8] **[3,9]** **[5,11]** [6,7] [6,12] [7,8] [7,13] [8,9] [8,14] [9,15] [10,16] **[11,17]** **[12,13]** [12,18] [13,14] **[13,19]** **[14,15]** **[14,20]** [15,16] [15,21] [16,22] [17,23] **[18,19]** **[18,24]** **[19,20]** **[19,25]** [20,21] [20,26] [21,22] [21,27] **[22,23]** [22,28] [23,29] **[24,25]** [25,26] [26,27] [26,32] [27,28] [27,33] [28,29] [28,34] [29,35] [31,32] [33,34] [34,35]

In each solution set above, we bold the arcs that are not all interdicted in the other two sets of solutions. For example, when $\phi = 0.8L$, we interdict the least number of arcs, among which

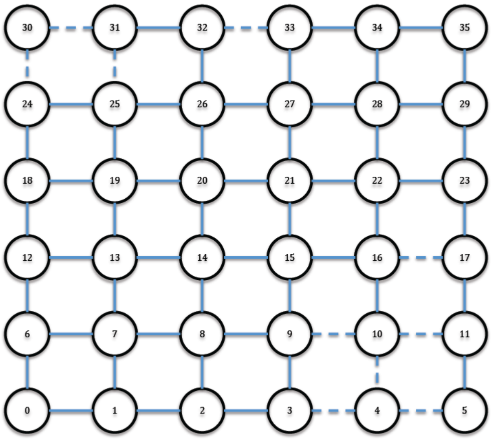

Figure 7: Example optimal arc-interdiction solution for $\phi = 0.9L$ in a 6×6 grid network with dashed lines representing the remaining arcs after interdiction.

arc $[12, 13]$ is not interdicted in $\phi = 0.85L$. For the other two cases, all the arcs interdicted under $\phi = 0.85L$ are all interdicted by $\phi = 0.9L$, and the latter interdicts additional arcs in bold. Due the special grid topology and non-uniform arc distances we use, the interdiction solutions are mostly nested as the distance threshold ϕ increases, which is very different from the solution pattern that we observe for the real transportation network instance given below.

We present in Figure 8 the optimal solutions for $\phi = 0.8L, 0.85L, 0.9L$ and risk parameter $\epsilon = 0.1$ based on the Russian railway network used in Schrijver [35].

The arcs marked by the dashed lines with smaller intervals (e.g., arcs [origin, 5] and [origin, 1]) are interdicted in all the three solutions by setting $\phi = 0.8L, 0.85L, 0.9L$. The arcs marked by the dashed lines with longer intervals (e.g., arcs [origin, 0] and [origin, 11]) are interdicted in solutions of $\phi = 0.85L$ and $0.9L$. We mark arcs only interdicted in the solution of $\phi = 0.8L$ by short thick lines (e.g., arc [18, 25]); arcs only interdicted in the solution of $\phi = 0.85L$ by rectangles (e.g., [38, 41]); and arcs only interdicted in the solution of $\phi = 0.9L$ by circles (e.g., [7, 14]). Although many interdicted arcs are in common, the optimal solutions with increasing distance threshold ϕ are not necessarily nested.

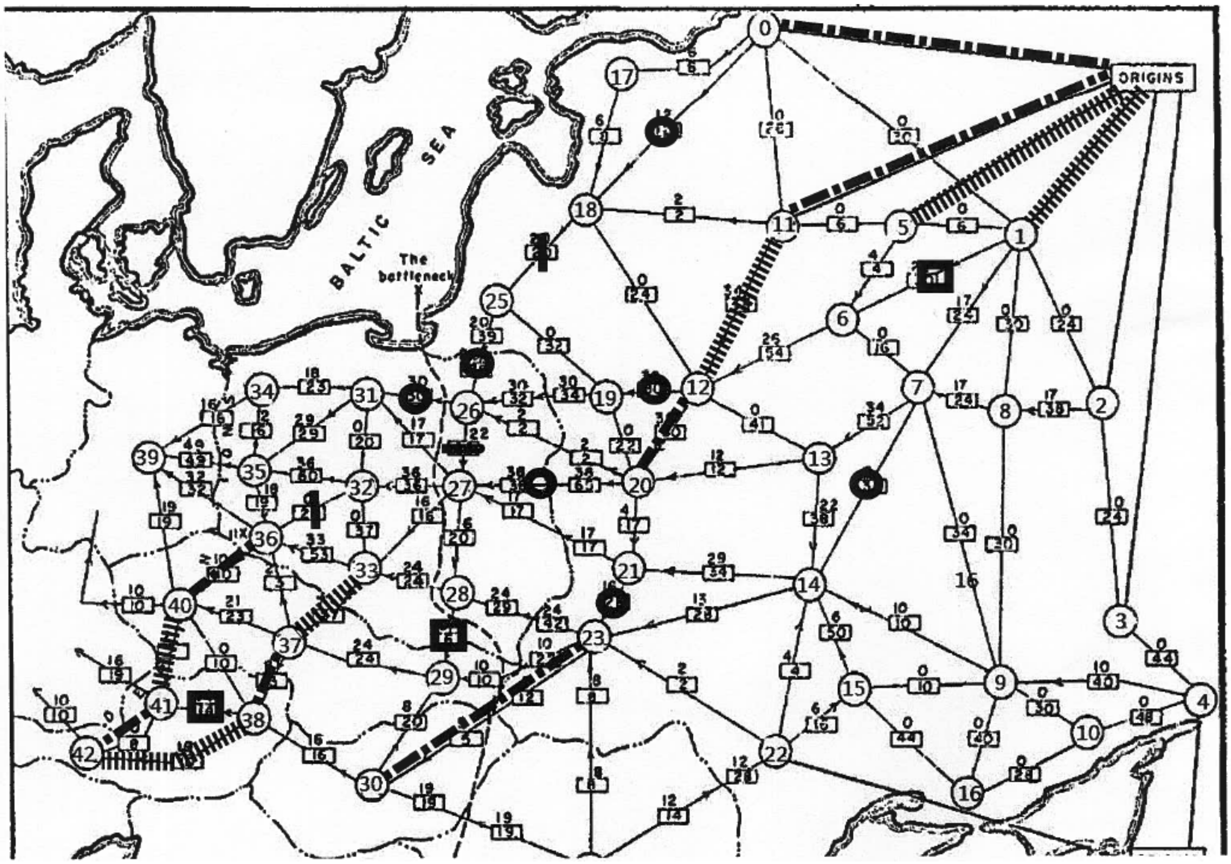

Figure 8: The optimal set of interdicted Arcs in the Russian Railway Network instance [35].