Copyright
by
Feng Pan
2005

The Dissertation Committee for Feng Pan
certifies that this is the approved version of the following dissertation:

Stochastic Network Interdiction: Models and Methods

Committee:

David P. Morton, Supervisor

William S. Charlton, Co-Supervisor

Jonathan F. Bard

Paul A. Jensen

Elmira Popova

Stochastic Network Interdiction: Models and Methods

by

Feng Pan, B.S., M.A.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2005

To my grandparents.

Acknowledgments

First and foremost, I would like to thank my advisor, Professor David Morton for his guidance and support. He introduced stochastic programming to me and provided me a positive and active research environment. I will miss the "Friday Morning" seminars. His knowledge, enthusiasm and insight have been truly invaluable. More personally, his encouragement and advice are vital in completing my dissertation.

I would like to thank my co-advisor, Professor Bill Charlton. He introduced me to the field of nonproliferation and to many exciting works in Los Alamos National Laboratory.

I thank my dissertation committee members, Prof. Jonathan Bard, Prof. Paul Jensen, and Prof. Elmira Popova for evaluating my dissertation and providing suggestions for its improvement. Especially, Prof. Popova advised me and supported me financially at the beginning of my program. I thank Prof. Crawford for encouraging me to pursuit a Ph.D. degree in operations research.

I would like to thank D-4 group and Second Line of Defense program office in Los Alamos National Laboratory for supporting my research. Especially, D-4 group management provided me a positive working environment where I was able to conduct my research. I want to thank Kevin Saeger for his support and discussion on many different and interesting applications in operations research.

I would like thank my friends, Guzin for interesting chatting of many different topics, Yong for answering my endless questions about CPLEX and Linux, and Anukal for discussion on stochastic programming and life in general.

I want to thank my parents for their patient and support. Finally, this work would not have been possible without the constant love and support from my wife, Mika.

Stochastic Network Interdiction: Models and Methods

Publication No. __________________

Feng Pan, Ph.D.
The University of Texas at Austin, 2005

Supervisors: David P. Morton
William S. Charlton

We develop stochastic network interdiction models and associated solution methods. In its simplest form, our model consists of a smuggler who wishes to traverse a network from an origin to a destination without being detected. Probabilities associated with the indigenous transportation network specify likelihoods that a smuggler can traverse each arc in the network undetected. By installing a detector on an arc we can decrease that probability. The decision-making problem is to select arcs to receive detectors subject to budget and policy constraints. The goal is to minimize the probability that a smuggler evades detection when the smuggler's origin-destination pair is known only through a probability distribution. The model has two stages: first we install detectors then the random origin-destination pair of the smuggler is revealed and the smuggler selects a maximum-reliability path knowing detector locations and detection probabilities. When we consider that detectors can only be installed on the "boundary" of the network, we show that the model can be reduced

to an interdiction problem on a simpler bipartite network. In other variants of the model, the smuggler has partial information on detector locations and may have a different perception (than the interdictor) of the detection probabilities. These models are cast as stochastic mixed-integer programs, and the complexity of the models is investigated. Our solution procedure includes scenario reduction, other preprocessing techniques and decomposition methods, all exploiting special structures in our stochastic network interdiction problems. We further enhance our solution procedures by developing a class of valid inequalities to tighten the integer-programming formulation.

This work is motivated by the Second Line of Defense (SLD) program, a cooperative effort between the US Department of Energy and the Russian Federation State Customs Committee. SLD's primary goal is to minimize the risk of illicit trafficking of nuclear materials and technologies through detection and deterrence by enhancing border detection capabilities.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

"Russian Foreign Minister Yevgeniy Primakov routed an FSK (Federalnaya Sluzhba Konterrazvedki) report ... the plutonium smuggled into Munich in August 1994 was stolen and then sold by four Russian citizens in Obninsk ..."

– OMRI Daily Digest entry, 13 February 1996

"On 16 March 1997, 10 'containers of radioactive material' were stolen from an arms factory located in Fier, Albania, ..."

– The Times, 21 March 1997, by Richard Owen

"The Ukrainian Main Directorate for Fighting Organized Crime detained a suspect attempting to transport a container holding 400g of radioactive material across the Ukrainian-Hungarian border on 24 February 2004, ..."

– Interfax-Ukraine, 26 February 2004

Illicit trafficking of nuclear material, equipment and technology poses an imminent threat to global security and world peace. The US government and much of the world community continually attempt to prevent the proliferation of weapons of mass destruction. In this dissertation we develop and solve mathematical programming models known as a stochastic network interdiction models, that are designed to decrease the likelihood that nuclear material can be smuggled across international borders.

Interdiction is the act of confronting and halting an activity. Interdiction models can provide tactics and strategies to thwart the activities of an adversary operating in some system. Network interdiction models are appropriate when the system in question can be modeled by a network. Network interdiction models typically involve the allocation of scarce resources to degrade the performance of an adversary whose behavior is modeled via a network optimization problem. Such models have application in the military, national security, law enforcement, computer security, business planning, and disease control.

The study of network interdiction models in operations research began in the 1970s. During the Vietnam War, McMasters and Mustin [40] and Ghare et al [27] developed deterministic mathematical programs to disrupt flow of enemy troops and materiel. The problem of maximizing an adversary's shortest path is considered in [26] and [28]. A closely related problem concerns maximizing the longest path in an adversary's PERT network [51]. The above models are solved as linear programs, and the interdictor can continuously increase the length of an arc, subject to a budget constraint. A discrete version of maximizing the shortest path removes an interdicted arc from the network, and when the budget constraint is simply a cardinality constraint, this is known as the k-most-vital arcs problem [20, 38]. The related decision problem is strongly NP-Complete [8, 13]. Generalizations of the k-most-vital arcs problem, and associated solution procedures, are considered in [33]. Wollmer [61] developed an algorithm for minimizing the maximum flow by removing arcs from a network. Wood [64] used integer programming to formulate the interdiction problem of removing arcs to minimize flow in an adversary's maximum-flow network and showed the (decision) problem to be strongly NP-Complete. See [59] for game-theoretic approaches to related network interdiction problems, [19] for an interdiction model on a minimum-cost-flow network, and [2] for an interdiction model for cost effective control of nosocomial infections in hospitals. Often, uncertainty is associated with the underlying network or the effectiveness of interdiction attempts. For instance, arc capacities, arc length, and even the topology of a network may be known only through a probability distribution. Also, the success of an interdiction attempt, e.g., increasing an arc's length, or decreasing an arc's capacity, can be modeled in a probabilistic manner. In [21], the work of [64] on interdicting a maximum-flow network is generalized to allow for random arc capacities and random interdiction successes. In these models, the interdictor does not know whether an interdiction attempt will successfully remove an arc and the interdictor does not know the true capacity of some arcs. However, the adversary knows the realizations of these random variables and maximizes flow in the residual network. The interdictor's goal is to minimize the expected value of this maximum flow. This problem is modeled as a two-stage stochastic program. Hemmecke et al. [31] developed models to interdict the minimum path while the underlying network topology is uncertain. For more on network interdiction problems, see the collection of papers in the recent book [63].

In Sections 1.2 and 1.3, we briefly review mixed-integer programming and stochastic programming. The emphasis is on topics relevant to this dissertation.

1.2 Mixed-integer Programming

Among early applications of integer programming, Dantzig, Fulkerson and Johnson [23] solved a 49-city traveling salesman problem by using a combination of the simplex method and cutting planes. In 1960, Land and Doig [37] presented a branch-and-bound algorithm for integer programming. Branch-and-bound algorithms rely on being able to solve a large number of linear programming (LP) problems within a reasonable amount of time. Branch-and-cut methods [22, 32, 45, 46] improve the effectiveness of branch-and-bound methods by adding cutting planes in the course of the branch-and-bound method. Since the 1950s, along with advances in column-generation methods, branch-and-price algorithms have become an effective method and remain an active research area [11, 12, 25, 34, 55, 57]. In a branch-and-price method, we can tighten an integer program's LP relaxation by a reformulation that involves a huge number of variables.

In the last paragraph we mentioned three major LP-based solution methods for integer programming. All these methods repeatedly solve LP relaxations, and the quality of the LP relaxation directly effects the methods. With respect to quality, it is ideal to have the feasible region of the LP relaxation coincide with the convex hull of the integer program's feasible region, at least in the neighborhood of the optimal solution. In branch-and-cut algorithms, cutting planes are added to restrict the feasible region of the LP relaxation. In 1958 Gomory [29, 30] proposed fractional cuts and proved that an associated cutting-plane algorithm converges finitely. Although these structure-independent cutting planes are valid for general integer programs, the algorithm, by itself, is not competitive with branch-and-bound schemes. In the 1990s, lift-and-project techniques [4, 5] were used to improve the performance of structure-independent cutting planes in the context of branch-and-bound methods. Structure-dependent cutting planes have been used in a variety of specially-structured problems, including knapsack problems [6, 22, 48, 60], vertex packing [3, 42, 43, 47], and the traveling salesman problem [7, 14, 46]. These structure-dependent cutting planes often prove to be enormously helpful, from a computational perspective.

The solution methods we discussed above can be applied to mixed-integer programming (MIP). In this dissertation, cutting-plane methods are a key solution method, and we introduce some relevant definitions in the rest of this section (see [62] for details). Cutting planes are essential in tightening LP relaxations. We also want to make sure that by adding inequalities, we don't eliminate any integer feasible solutions. Let $X \subseteq R^n$ be a polyhedron.

Definition 1.2.1. *An inequality $\pi x \leq \pi_0$ is a valid inequality for the polyhedron X if $\pi x \leq \pi_0$ for all $x \in X$*

In the content of MIP, X can be the convex hull of the feasible region of a mixed-integer linear program. Valid inequalities are added to eliminate some feasible fractional solutions of the LP relaxation while keeping all integral feasible solutions of the mixed-integer linear program intact.

There can be infinitely many valid inequalities for a given polyhedron, so we seek a theoretical measure of the "quality" of a valid inequality.

Definition 1.2.2. *If $\pi x \leq \pi_0$ and $\mu x \leq \mu_0$ are both valid inequalities for the polyhedron $X \subseteq R^n_+$, $\pi x \leq \pi_0$* dominates *$\mu x \leq \mu_0$* if there exists a scalar $u > 0$ such that $\pi \geq u\mu$ and $\pi_0 \leq u\mu_0$, and $(\pi, \pi_0) \neq (u\mu, u\mu_0)$.

A valid inequality becomes redundant to the description of the convex hull of a mixed-integer linear program's feasible region if it is dominated by other valid inequalities.

Definition 1.2.3. *For a valid inequality $\pi x \leq \pi_0$ of the polyhedron X, F is a* face *of X if $F = \{x \in X : \pi x = \pi_0\} \neq \emptyset$; a face becomes a* facet *if* $\dim(F) = \dim(X) - 1$.

Facets are strong valid inequalities for a polyhedron X, and they are necessary for the description of X, at least when representing it as an intersection of half spaces. Thus, when a facet is removed, the resulting polyhedron differs from the original one.

For an NP-Complete MIP problem, it is impossible to obtain the explicit description of the convex hull of the mixed-integer linear program's feasible region in polynomial time unless $NP = P$. However, by adding valid inequalities, the convex hulls of a mixed-integer linear program and its LP relaxation in the neighborhood of optimal solution can become closer, and the gap between the objective values of the mixed-integer linear program and its LP relaxation can decrease. However, adding a large number of valid inequalities typically increases the computational effort required to solve the LP relaxation. Thus, identifying facets becomes important. One approach to show that a valid inequality defines a facet of the polyhedron X is to find $\dim(X)$ affinely independent points of X at which the valid inequality holds with equality.

In this dissertation, we mainly focus on the branch-and-bound and cutting-plane methods to solve our MIP problems. For other solution techniques, see [39, 44, 62].

1.3 Stochastic Programming

Stochastic programming (SP) deals with decision making under certainty. In contrast to deterministic optimization, model parameters are not known with certainty but rather via probability distributions. Stochastic programming can be viewed as generalizing deterministic mathematical programming. There are several different classes of stochastic programming models [18, 35, 50], and in this dissertation, we concentrate on integer programming variants of the following two-stage stochastic linear program with recourse

$$
\begin{aligned}
\min_{x} \quad & cx + E[h(x,\xi)] \\
\text{s.t.} \quad & Ax = b \\
& x \geq 0,
\end{aligned}
\tag{1.1}
$$

where

$$
\begin{aligned}
h(x,\xi^\omega) = \min_{y} \quad & f^\omega y \\
\text{s.t.} \quad & D^\omega y = d^\omega - B^\omega x \\
& y \geq 0.
\end{aligned}
\tag{1.2}
$$

We call (1.2) the second-stage problem and $E[h(x,\xi)]$ the recourse function. Here, ξ is a random vector consisting of the random elements of D, B, f and d, and ξ^ω, $\omega \in \Omega$, denotes a realization of ξ. Timing is a key issue in SP: a here-and-now decision x is made first, then a scenario ξ^ω, $\omega \in \Omega$, is realized, and a recourse action y is taken. In contrast to other areas of stochastic optimization, such as statistical decision theory, Markov decision processes, stochastic dynamic programming, and stochastic control, stochastic programming is a branch of mathematical programming, and so it inherits a rich class of models and solution techniques. See the textbooks [18, 35, 50] for detailed introductions to stochastic programming. The two-stage stochastic linear program was introduced by Dantzig [24] and Beale [15]. Broadly speaking, there are three categories of solution methods for such problems. A stochastic program with a finite number of scenarios can be transformed into an equivalent large-scale, but deterministic, optimization problem, and solution techniques from large-scale optimization can be applied, including decomposition-based methods, such as the L-shaped method [58], its variants [17, 54], and Lagrangian methods [52, 53]. Deterministic approximation methods are used to deal with the difficulty of high-dimensional integration, typically by reducing the number of scenarios and iteratively improving the upper and lower bounds on the optimal value of the original problem. For stochastic programs with a large number of scenarios and involving many dimensions, Monte Carlo simulation is widely regarded as the method of choice.

In this dissertation, we use the L-shaped method, which is closely related to Benders decomposition from mixed-integer linear programming and Kelley's cutting-plane method from nonlinear programming, to solve our stochastic programs. The second-stage problem, (1.2), is a parametric optimization problem in x and ξ^ω. When D is deterministic, we say (1.2) has fixed recourse. For any feasible first stage decision variable x, if the second-stage problem (1.2) is feasible for all $\omega \in \Omega$ then we say the problem has relatively complete recourse.

Theorem 1.3.1. *Assume that the second-stage problem (1.2) has fixed and relatively complete recourse and* $X = \{x : Ax = b, x \geq 0\}$ *then*

1. *if* $\xi = (d, B)$ *then* $h(x, \cdot)$ *is piecewise linear and convex on the convex hull of* ξ*'s support provided* $x \in X$*.*

2. *if* $\xi = (f)$ *then* $h(x, \cdot)$ *is piecewise linear and concave on the convex hull of* ξ*'s support provided* $x \in X$*;*

3. $h(\cdot, \xi^\omega)$ *is piecewise linear and convex on* X*.*

Under a discrete probability distribution in which p^ω denotes the probability of realization ξ^ω, the two-stage stochastic program may be written

$$\begin{aligned} \min_{x\geq 0,\theta} \quad & cx + \sum_{\omega\in\Omega} p^\omega \theta^\omega \\ \text{s.t.} \quad & Ax = b \\ & \theta^\omega \geq h(x,\xi^\omega) \qquad \forall \omega \in \Omega. \end{aligned} \tag{1.3}$$

The dual of the second-stage problem (1.2) is

$$\begin{aligned} h(x,\xi^\omega) = \max_{\pi} \quad & \pi(d^\omega - B^\omega x) \\ \text{s.t.} \quad & \pi D \leq f^\omega, \end{aligned} \tag{1.4}$$

and we call (1.4) the subproblem. In this dissertation, we assume that the optimal value of (1.2) is finite for all x satisfying the constraints of (1.1) and all scenario ω. We can express (1.4) as follows

$$h(x,\xi^\omega) = \max_{i\in\{1,2,\cdots,L^\omega\}} \pi^{\omega,i}(d^\omega - B^\omega x),$$

where $\pi^{\omega,1},\cdots,\pi^{\omega,L^\omega}$ are the extreme points of the second-stage dual feasible region $\{\pi : \pi D \leq f^\omega\}$. Let $g^{\omega,i} = \pi^{\omega,i} d^\omega$ and $G^{\omega,i} = \pi^{\omega,i} B^\omega$. We call

$$\theta^\omega \geq g^{\omega,i} - G^{\omega,i} x \tag{1.5}$$

an optimality cut. Explicitly expressing $h(x,\xi^\omega)$ in term of optimality cuts, we can rewrite (1.3) as

$$\begin{aligned} \min_{x\geq 0,\theta} \quad & cx + \sum_{\omega\in\Omega} p^\omega \theta^\omega \\ \text{s.t.} \quad & Ax = b \\ & \theta^\omega \geq g^{\omega,i} - G^{\omega,i} x, \quad i = 1,\cdots,L^\omega, \ \omega \in \Omega, \end{aligned} \tag{1.6}$$

which is the full master program because it contains all optimality cuts for each scenario. The number of scenarios and the number of extreme points of $\{\pi : \pi D \leq f^\omega\}$ can make the full master program very large. So, we start with a relaxed master program,

$$\begin{aligned}
\min_{x\geq 0,\theta} \quad & cx + \sum_{\omega\in\Omega} p^\omega \theta^\omega \\
\text{s.t.} \quad & Ax = b \\
& \theta^\omega \geq g^{\omega,i} - G^{\omega,i}x, \quad i = 1,\cdots,l^\omega,\ \omega\in\Omega,
\end{aligned} \tag{1.7}$$

where only subsets of optimality cuts are included for each $\omega \in \Omega$ when $l^\omega < L^\omega$. Since we may not need all the optimality cuts to find an optimal solution, we may obtain an optimal solution from a relaxed master program which is much smaller than the full master program A multi-cut version of the L-shaped algorithm is described as following.

Multi-cut L-shaped Algorithm

Step 0 Define the tolerance $\epsilon \geq 0$. Set $\bar{z} = \infty$ and $\underline{z} = -M$. Initialize the set of cuts with $\theta^\omega \geq -M$, $\omega \in \Omega$.

Step 1 Solve the relaxed master problem (1.7). Obtain the optimal solution $(\hat{x}, \hat{\theta})$, and let $\underline{z} = c\hat{x} + \sum_{\omega\in\Omega} p^\omega \hat{\theta}^\omega$.

Step 2 For $\omega \in \Omega$, solve subproblem (1.4) with $x = \hat{x}$ and obtain the primal and dual optimal solution $(\hat{y}^\omega, \hat{\pi}^\omega)$. Let $\hat{z} = c\hat{x} + \sum_{\omega\in\Omega} p^\omega f^\omega \hat{y}^\omega$. If $\hat{z} < \bar{z}$, then put $\bar{z} = \hat{z}$ and $x^* = \hat{x}$.

Step 3 If $\bar{z} - \underline{z} \leq \epsilon \min(|\bar{z}|, |\underline{z}|)$, then stop: x^* is a solution with an objective function value within $(100\epsilon)\%$ of optimal.

Step 4 For $\omega \in \Omega$, if $\hat{\theta}^\omega < \hat{\pi}^\omega(d^\omega - B^\omega \hat{x})$, let $l^\omega = l^\omega + 1$ and add

$$\theta^\omega \geq \hat{\pi}^\omega d^\omega - \hat{\pi}^\omega B^\omega x$$

to the relaxed master problem (1.7). Go to **Step 1**.

We call this procedure the multi-cut L-shaped algorithm because during each iteration, it possibly generates an optimality cut for each scenario. The multi-cut L-shaped algorithm provides better resolution of the approximation of the recourse function, and it is expected to require fewer iterations of solving the master problem. If the master problem is a mixed-integer linear program, reducing the number of times we must solve it can reduce the overall computational effort. For more detailed discussions on solutions methods for SP, see [18, 35, 50].

1.4 Overview of the Contents

In this dissertation, we introduce new stochastic network interdiction models and develop associated solution methods. Chapter 2 develops our basic stochastic network interdiction model in which the evader's maximum-reliability path and the associated evasion probability are calculated under the assumption that the evader knows the detector locations and the detection probabilities. Variants of this basic interdiction model are developed under different assumptions that the smuggler has partial information on detector locations or different perceptions (than the interdictor) of detection probabilities. We also consider a special case in which all interdictions can be only made on the border of a country, and in this case the basic interdiction model reduces to a stochastic bipartite network interdiction problem. Complexity of these interdiction models is also discussed.

Chapter 3 concentrates on solving the stochastic bipartite network interdiction problem. We consider the model in its deterministic equivalent form, which is a mixed-integer linear program. Coefficient reduction and other preprocessing techniques are used to reduce the model size and valid inequalities are introduced to strengthen the model's LP relaxation. We provide structural properties for the bipartite model, we introduce a class of valid inequalities, and we give computational results for our solution techniques.

Chapter 4 discusses solution methods for the basic stochastic network interdiction model on a general network. Preprocessing is used to improve the original formulation. We develop a solution method combining the multi-cut L-shaped algorithm and an extension of the valid inequality from Chapter 3 to solve this stochastic network interdiction problem. Local search and associated optimality cuts are used to improve the performance of the multi-cut L-shaped algorithm. Computational results are discussed.

Chapter 5 gives conclusions on our stochastic network interdiction models and solution techniques. We also suggest future research areas in network interdiction.

Chapter 2

Stochastic Network Interdiction Models

2.1 Introduction

In Chapter 1, we reviewed network interdiction models and their applications. A network interdiction problem can be viewed as a bi-level optimization problem (e.g., [10, 56]) with the higher level being a resource allocation problem and the lower level being a network flow problem. In contrast to the interdiction models described in Chapter 1, the objective of our model is to minimize the probability of a successful smuggling attempt through a transportation network, where the smuggler is assumed to choose a maximum-reliability path at the lower level. This model may be appropriate when the consequence of a successful smuggling attempt could be catastrophic (e.g., smuggling of a weapon of mass destruction). In these cases, it is important to reduce the probability of a successful smuggling rather than minimizing the volume of smuggled material, or maximizing the time and cost of a smuggling attempt.

Our motivation comes from the Second Line of Defense (SLD) program, which is a cooperative effort between the US Department of Energy and the Russian Federation State Customs Committee. The objective of SLD is to reduce the risk of illicit trafficking of nuclear material and technology through detection and deterrence by enhancing detection capabilities at border crossings. Consider a smuggler who obtains weapons-grade nuclear material from a facility in Russia and then tries to reach a final destination outside of Russia. Without nuclear material detectors, the smuggler may be caught by the police or other law enforcement through random or routine checks at road blocks or customs sites. By installing detectors at customs sites, the chance of detecting the smuggler is improved. Due to resource limitations, the question then becomes at which customs sites equipment should be deployed to maximize the detection probability.

We model two adversaries, an interdictor and an evader (also called a smuggler), and an underlying network $G(N, A)$ on which the evader travels. In the *deterministic* version of the model, the evader starts at a specified source node $s \in N$ and wishes to reach a specified terminal node $t \in N$. The model is deterministic in that this origin-destination pair is known. If the interdictor has not installed a *sensor* on arc $(i, j) \in A$, then the probability that the evader can traverse (i, j) undetected is p_{ij}, and this probability is $q_{ij} < p_{ij}$ if the interdictor has installed a detector on (i, j). (We use the terms sensor and detector interchangeably.) The events of the evader being detected on distinct arcs are assumed to be mutually independent. The evader chooses a path from s to t so as to maximize the probability of traversing the network without being detected. With limited resources, the interdictor must select arcs on which to install detectors in order to minimize the probability the evader travels from s to t undetected. The evader's optimization problem is known as the maximum-reliability path problem and, even though there are probability values on the arcs (p_{ij}'s and q_{ij}'s), this problem can be solved as a deterministic shortest-path problem (e.g., exercise 4.39 of [1]).

Our *stochastic* network interdiction model differs from the above description only in that the (s,t) pair for the evader is unknown when the interdictor must select sites for installing sensors. However, the origin-destination pair (s,t) is assumed to be governed by a known probability mass function, $p^\omega = P\{(s,t) = (s^\omega, t^\omega)\}$, $\omega \in \Omega$. The interdictor's goal is to minimize the probability that the evader traverses the network undetected, i.e., the objective function is a sum of (conditional) evasion probabilities, each weighted by p^ω, over all possible origin-destination pairs. The timing of decisions and realizations is key in this interdiction problem. First, the interdictor installs sensors. Then, a random origin-destination pair for the evader is revealed and the evader selects an s^ω–t^ω path to maximize the probability of avoiding detection. The evader selects this path with knowledge of the locations of the detectors and the evasion probabilities p_{ij} and q_{ij}, respectively, for all $(i,j) \in A$. An evader can be caught by indigenous law enforcement without detection equipment and so $p_{ij} < 1$. To date, nuclear smuggling attempts that have been stopped have been by this means.

The values of p_{ij} and q_{ij} are important parameters in our model. How do we estimate these values? Also, does an evader always travel on a maximum-reliability path? More generally, how does an evader behave? These questions are beyond the scope of this dissertation.

In Section 2.2, we introduce further notation and make our modeling assumptions precise. In Section 2.3 we describe our basic stochastic network interdiction problem (SNIP). In SNIP, we assume that an informed evader knows the location of detectors and has the same perception of p_{ij} and q_{ij} as the interdictor does. Our model is a two-stage program, using a "min-max" structure because the interdictor is minimizing the evader's maximum evasion probability. We then develop an equivalent two-stage stochastic Linear mixed-integer program. The need for the MIP formulation is justified by the fact that SNIP or, more precisely, the related decision problem, SNIP-DECISION, is strongly NP-Complete (Section 2.8). In Section 2.4, the U-SNIP model for an uninformed evader is derived, and Section 2.5 gives a hybrid model with a mixed population of informed and uninformed evaders. In Section 2.7, we consider an important special case of SNIP that arose in our SLD work in which sensors can only be installed at border crossings of a single country. In this special case, the underlying network and the associated mixed-integer linear program can be simplified to a bipartite stochastic network interdiction problem (BiSNIP), which is also shown to be NP-Complete in Section 2.8.

2.2 Notation and Assumptions

Our stochastic network interdiction models require the following parameters: the probability a smuggler can traverse a physical transportation arc undetected, the probability that sensitive material will be detected by an installed sensor, and the probability a smuggler steals material from a particular origin and wants to travel to a specific destination. We define the basic notation for our stochastic network interdiction models as follows.

Network and Sets:

$G(N, A)$	directed network with node set N and arc set A
$FS(i)$	set of arcs leaving node i
$RS(i)$	set of arcs entering node i
$AD \subset A$	set of arcs on which a detector can be installed

Data:

b	total budget for installing detectors
a_{ij}	cost of installing a detector on arc $(i, j) \in AD$
p_{ij}	probability evader can traverse arc (i, j) undetected when no detector is installed
q_{ij}	probability evader can traverse arc (i, j) undetected when a detector is installed

Random Elements:

(s^ω, t^ω)	realization of random origin-destination pair
$\omega \in \Omega$	sample point and sample space
p^ω	probability mass function

Interdictor's Decision Variables:

x_{ij}	1 if a detector is installed on arc $(i, j) \in AD$ and 0 otherwise

Evader's Decision Variables:

y_{ij}	arc flow which takes positive value only if evader traverses arc (i, j) and no detector is installed on that arc
z_{ij}	arc flow which takes positive value only if evader traverses arc (i, j) and a detector is installed on that arc

Boundary Conditions:

$x_{ij} \equiv 0 \quad (i, j) \notin AD$

$z_{ij} \equiv 0 \quad (i, j) \notin AD$

Sample space Ω is the collection of origin-destination pairs, and Ω

doesn't include more detail on the "type" of evader unless we specify otherwise. The following assumptions are assumed to hold for all the variants of our stochastic network interdiction models.

General Assumptions:

1. The events of detecting an evader on distinct arcs are mutually independent.
2. The network topology $G(N, A)$ is known to both the interdictor and the evader, and there is an s^ω–t^ω path, $\forall\ \omega \in \Omega$.
3. The interdictor knows the method by which the evader selects a path and the probability distribution governing the random (s, t) pair.

2.3 SNIP: Stochastic Network Interdiction Model for the Informed Evader

In our base model, SNIP, we deal with an informed evader. In addition to General Assumptions 1–3, the following are assumed to hold.

SNIP Assumptions:

1. An informed evader knows the evasion probability, p_{ij}, on each indigenous arc $(i, j) \in A$.
2. The informed evader knows the location of all the detectors, x_{ij}, $(i, j) \in AD$, and the evasion probability, $q_{ij} < p_{ij}$, on each detector arc $(i, j) \in AD$.

SNIP is formulated as follows

$$\min_{x \in X} \sum_{\omega \in \Omega} p^\omega h(x, (s^\omega, t^\omega)), \tag{2.1}$$

where

$$X = \left\{ x : \sum_{(i,j) \in AD} a_{ij} x_{ij} \leq b,\ x_{ij} \in \{0, 1\},\ (i, j) \in AD \right\},$$

and where

$$h(x, (s^\omega, t^\omega)) =$$

$$
\begin{aligned}
\max_{y \geq 0, z \geq 0} \quad & y_{t^\omega} && \text{(2.2a)} \\
\text{s.t.} \quad & \sum_{(s^\omega, j) \in FS(s^\omega)} (y_{s^\omega j} + z_{s^\omega j}) = 1 && \text{(2.2b)} \\
& \sum_{(i,j) \in FS(i)} (y_{ij} + z_{ij}) - \\
& \sum_{(j,i) \in RS(i)} (p_{ji} y_{ji} + q_{ji} z_{ji}) = 0, \quad i \in N \setminus \{s^\omega, t^\omega\} && \text{(2.2c)} \\
& y_{t^\omega} - \sum_{(j,t^\omega) \in RS(t^\omega)} (p_{jt^\omega} y_{jt^\omega} + q_{jt^\omega} z_{jt^\omega}) = 0 && \text{(2.2d)} \\
& y_{ij} \leq 1 - x_{ij}, \qquad (i,j) \in AD && \text{(2.2e)} \\
& z_{ij} \leq x_{ij}, \qquad (i,j) \in AD. && \text{(2.2f)}
\end{aligned}
$$

The conditional probability a smuggler avoids detection, given (s^ω, t^ω), is $h(x, (s^\omega, t^\omega))$ as defined in (2.2). The objective function in (2.1) is the unconditional evasion probability, formed by the weighted sum over all possible origin-destination pairs. The set of feasible detector installation locations defined through X is governed by a budget constraint and binary restrictions on x.

The network on which the evader travels contains interdictable and non-interdictable arcs (see Figure 2.1). Each link in the network on which a detector can be placed may be viewed as two arcs in parallel. If a detector is installed, i.e., $x_{ij} = 1$, then flow may only occur on the "detector" arc with flow variable z_{ij}, and there is no flow on the indigenous arc, i.e., $y_{ij} = 0$.

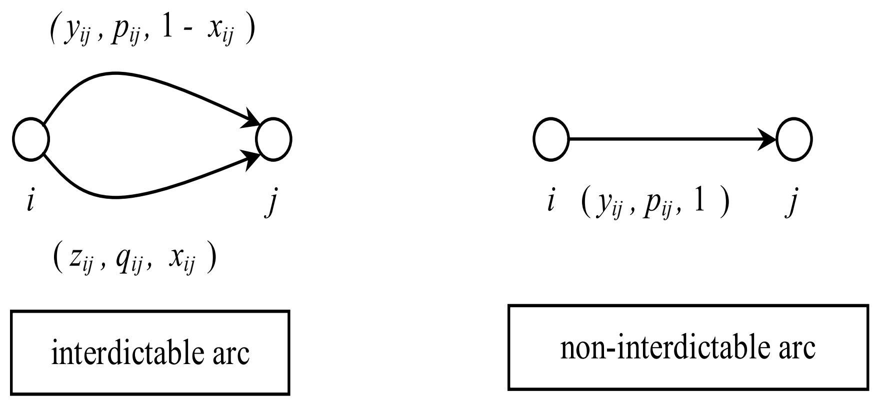

Figure 2.1: The graph shows two arc types: an interdictable arc and a non-interdictable arc. The triple associated with each arc denotes (flow decision variable, gain, capacity). Arc capacities on interdictable arcs force flow onto the "detector" arc if a detector is installed.

Conversely, if no detector is installed, i.e., $x_{ij} = 0$, flow can only occur on the indigenous arc. A unit of flow on arc (i, j) is multiplied by that arc's gain (either p_{ij} or q_{ij}). So, if P_{s^ω,t^ω} is a path from s^ω to t^ω then

$$y_{t^\omega} = \prod_{(i,j)\in P_{s^\omega,t^\omega}} [p_{ij}(1 - x_{ij}) + q_{ij}x_{ij}] \tag{2.3}$$

is the probability that an evader can travel from s^ω to t^ω on P_{s^ω,t^ω} without being detected. The evader's goal is to select a path P_{s^ω,t^ω} that maximizes y_{t^ω}. The evader's subproblem (2.2) accomplishes this by forcing one unit of flow out of s^ω in (2.2b), enforcing flow conservation at all intermediate nodes in (2.2c), defining the flow that reaches t^ω as y_{t^ω} in (2.2d) and maximizing that value in (2.2a). Flow is forced on the appropriate arc, and incurs the associated gain (actually, loss), by the interdictor's decision variable x_{ij} in constraints (2.2e) and (2.2f).

Important elements of SNIP include the timing of the interdictor's and evader's decisions related to the realization of the smuggler's origin-destination pair, and the information available to the interdictor and evader. The timing of decisions and the realization of the uncertainty in (2.1) is as follows: First, the interdictor selects sites for installing sensors, subject to the budget constraint. Next, an (s^{ω}, t^{ω}) realization is revealed, and the evader selects an s^{ω}–t^{ω} path that maximizes the probability of not being detected. Both the General Assumptions and the SNIP Assumptions are in place for SNIP. Assuming the smuggler solves an optimization model to select an s^{ω}–t^{ω} path is a behavioral assumption. In Sections 2.4–2.6, we develop model variants with different behavioral assumptions. We note that even in cases when this assumption may not be valid, the optimal value of (2.1) still provides a potentially useful *pessimistic* prediction of the evasion probability.

The SNIP model (2.1) is a bi-level stochastic mixed-integer linear program. In bi-level programs (e.g., [10, 9, 16, 56]) both players have an objective function, and these can differ because the players' motives differ. In our case, the objective function is the same for both players, but the interdictor seeks to minimize that function while the evader wishes to maximize it. SNIP (2.1) is formulated with a nested "min-max" structure, and so it is not possible to solve in this form as a single large-scale mathematical program. One natural way to attempt to circumvent this difficulty (see e.g., [26, 64]) is to take the dual of the linear programming subproblem (2.2) so that the problem is expressed in a nested "min-min" form. We could then construct a single optimization model in which we simultaneously minimize over the interdictor's decision x and the (dual) decision variables of the evader under each scenario, $\omega \in \Omega$. The difficulty with this approach is that there are nonlinear terms involving x and the dual variables associated with constraints (2.2e) and (2.2f) and the prospects for solving realistically-sized instances of the resulting stochastic nonlinear nonconvex mixed-integer program are not good. Instead we employ the exact-penalty result of Lemma 2.3.1, which is adapted from [41, Lemma 2], in order to reformulate (2.2).

Lemma 2.3.1. *Consider the following linear program*

$$\begin{aligned} z_1^* = \quad & \max_{x \geq 0} \quad cx \\ & \text{s.t.} \quad Ax = b \quad : \pi \\ & \qquad\quad x \leq u \quad\; : \gamma, \end{aligned} \tag{2.4}$$

where $A \in \Re^{m \times n}$, the remaining vectors are dimensioned to conform, and π and γ are dual row vectors. Assume (2.4) has a finite optimal solution, (π^*, γ^*) is an arbitrary optimal dual vector, and consider

$$\begin{aligned} z_2^* = \quad & \max_{x \geq 0} \quad cx - \gamma'(x-u)^+ \\ & \text{s.t.} \quad Ax = b, \end{aligned} \tag{2.5}$$

where $(x-u)^+ = \max(x-u, 0)$ and $\gamma' \in \Re^n$. If $\gamma' \geq \gamma^*$ then $z_1^* = z_2^*$.

Proof. The dual of (2.4) is

$$\begin{aligned} z_1^D = \quad & \min_{\pi, \gamma} \quad \pi b + \gamma u \\ & \text{s.t.} \quad \pi A + \gamma \geq c \\ & \qquad\quad \gamma \geq 0, \end{aligned} \tag{2.6}$$

and by linear programming strong duality, $z_1^D = z_1^*$. Adding an upper bound constraint on γ, we have the following linear program

$$
\begin{array}{rlll}
z_2^D = & \displaystyle\min_{\pi,\gamma} & \pi b + \gamma u & \\
& \text{s.t.} & \pi A + \gamma \geq c & : x \\
& & \gamma \leq \gamma' & : -y \\
& & \gamma \geq 0. &
\end{array}
\tag{2.7}
$$

By hypothesis, $\gamma' \geq \gamma^*$, and thus $z_2^D = z_1^D$. The dual of (2.7) is

$$
\begin{array}{rll}
z_3^* = & \displaystyle\max_{x\geq 0, y\geq 0} & cx - \gamma' y \\
& \text{s.t.} & Ax = b \\
& & x - y \leq u.
\end{array}
\tag{2.8}
$$

By strong duality, $z_3^* = z_2^D$. Since $y \geq 0$ and $\gamma' \geq 0$, we can ensure there is an optimal solution with $y = (x-u)^+$. Therefore, $z_2^* = z_3^* = z_1^*$. $\square$

The following theorem uses Lemma 2.3.1 to establish an equivalent expression to (2.2) for $h(x, (s^\omega, t^\omega))$.

Theorem 2.3.2. *Assume that G has an s^ω–t^ω path $\forall\, \omega \in \Omega$, $0 \leq p_{ij} \leq 1$, $(i,j) \in A$, and $0 \leq q_{ij} \leq 1$, $(i,j) \in AD$. Then, for all $x \in X$ and $\omega \in \Omega$, $h(x, (s^\omega, t^\omega))$ is the optimal value of the following linear program*

$$
\begin{array}{llll}
\displaystyle\min_{\pi} & \pi_{s^\omega} & & \\
\text{s.t.} & \pi_i - p_{ij}\pi_j \geq 0, & (i,j) \in A \setminus AD & \text{(2.9a)} \\
& \pi_i - p_{ij}\pi_j \geq -x_{ij}, & (i,j) \in AD & \text{(2.9b)} \\
& \pi_i - q_{ij}\pi_j \geq x_{ij} - 1, & (i,j) \in AD & \text{(2.9c)} \\
& \pi_{t^\omega} = 1. & &
\end{array}
$$

Proof. Let $\omega \in \Omega$ and $x \in X$. G has an s^ω-t^ω path and hence (2.2) is feasible and has a finite optimal solution. Let λ^*_{ij} and γ^*_{ij}, $(i,j) \in AD$, be optimal dual variables for constraints (2.2e) and (2.2f), respectively. These dual variables are bounded above by one because the network gains, p_{ij}, $(i,j) \in A$, and q_{ij}, $(i,j) \in AD$, are at most unity and hence an increase in the capacity of an arc by ϵ can increase the flow exiting that arc by no more than ϵ and therefore contribute at most ϵ to the flow reaching t^ω. So, employing Lemma 2.3.1 we can conclude that $h(x,(s^\omega,t^\omega))$ is the optimal value of

$$\begin{aligned}
\max_{y\geq 0, z\geq 0} \quad & y_{t^\omega} - \sum_{(i,j)\in AD} \left[(y_{ij} - (1-x_{ij}))^+ + (z_{ij} - x_{ij})^+\right] \\
\text{s.t.} \quad & \sum_{(s^\omega,j)\in FS(s^\omega)} (y_{s^\omega j} + z_{s^\omega j}) = 1 \\
& \sum_{(i,j)\in FS(i)} (y_{ij} + z_{ij}) - \\
& \sum_{(j,i)\in RS(i)} (p_{ji}y_{ji} + q_{ji}z_{ji}) = 0, \quad i \in N \setminus \{s^\omega, t^\omega\} \\
& y_{t^\omega} - \sum_{(j,t^\omega)\in RS(t^\omega)} (p_{jt^\omega}y_{jt^\omega} + q_{jt^\omega}z_{jt^\omega}) = 0.
\end{aligned} \tag{2.10}$$

Because of the binary nature of $x \in X$, we have $(y_{ij} - (1-x_{ij}))^+ = x_{ij}y_{ij}$ and $(z_{ij} - x_{ij})^+ = (1-x_{ij})z_{ij}$ for y_{ij} and z_{ij}, $(i,j) \in AD$, satisfying the constraints of (2.10). Making these substitutions in the objective function of (2.10) and taking the dual of the resulting linear program yields (2.9). □

The expression for $h(x,(s^\omega,t^\omega))$ in (2.9) has the following interpretation: The dual variable π_i is the conditional probability of the evader traveling from node i to destination t^ω undetected, given that the evader has reached node i undetected. Constraints (2.9a)-(2.9c), coupled with minimizing the objective function, ensure the correct computation of π_{s^ω}, i.e., the probability of traversing the network from s^ω to t^ω undetected. In (2.9), the evader selects a maximum-reliability path, and constraints (2.9a)-(2.9c) are tight for (i,j) on this selected path. When $x_{ij} = 1$, i.e., a detector is installed on (i,j), the indigenous-arc constraint (2.9b) is vacuous and when $x_{ij} = 0$ the detector-arc constraint (2.9c) is vacuous. The unit multiplicative coefficients on the right-hand sides of (2.9b) and (2.9c) can be decreased to tighten the MIP formulation of SNIP that we present below in (2.11). For example, (2.9b) can be rewritten $\pi_i - p_{ij}\pi_j \geq -\alpha_{ij}x_{ij}$, provided the coefficient α_{ij} satisfies $\alpha_{ij} \geq p_{ij}\pi_j$. We do not pursue this issue now but will return to it in our computational work in Chapter 4.

The value of Theorem 2.3.2 is that we can now express our original nested "min-max" formulation of the interdiction model for the informed evader, i.e., (2.1) with (2.2), as the following two-stage stochastic mixed-integer linear program, which we denote by SNIP

$$
\begin{aligned}
\min_{x,\pi} \quad & \sum_{\omega\in\Omega} p^\omega \pi^\omega_{s^\omega} \\
\text{s.t.} \quad & x \in X \\
& \pi^\omega_i - p_{ij}\pi^\omega_j \geq 0, && (i,j) \in A \setminus AD, \omega \in \Omega && \text{(2.11a)} \\
& \pi^\omega_i - p_{ij}\pi^\omega_j + x_{ij} \geq 0, && (i,j) \in AD, \omega \in \Omega && \text{(2.11b)} \\
& \pi^\omega_i - q_{ij}\pi^\omega_j + (1 - x_{ij}) \geq 0, && (i,j) \in AD, \omega \in \Omega && \text{(2.11c)} \\
& \pi_{t^\omega} = 1, && \omega \in \Omega. && \text{(2.11d)}
\end{aligned}
$$

2.4 U-SNIP: Stochastic Network Interdiction Model for the Uninformed Evader

For an uninformed evader, in addition to General Assumptions 1-3, the following are assumed to hold.

U-SNIP Assumptions:

1. The evader knows the evasion probability, p_{ij}, on each indigenous arc $(i, j) \in A$.
2. The evader has no information regarding sensor locations and the evasion probabilities, q_{ij}, on the detector arcs, $(i, j) \in AD$.

In U-SNIP, the evader behaves as if detectors do not exist and selects a maximum-reliability path based on the indigenous network. As indicated in the General Assumptions, the interdictor knows the means by which the uninformed evader's path is selected.

Assume, for the moment, that for each origin-destination pair, (s^ω, t^ω), $\omega \in \Omega$, that the evader's optimal path, $P^*(s^\omega, t^\omega)$, in the indigenous network is unique. Then the interdictor can restrict sensor installations to the interdictable arcs on these optimal paths. The U-SNIP model can then be stated

$$
\begin{aligned}
\min_{x,\pi} \quad & \sum_{\omega \in \Omega} p^{\omega} \pi^{\omega}_{s^{\omega}} \\
\text{s.t.} \quad & x \in X \\
& \pi^{\omega}_i - p_{ij}\pi^{\omega}_j \geq 0, && (i,j) \in P^*(s^{\omega}, t^{\omega}) \setminus AD, \omega \in \Omega && (2.12a) \\
& \pi^{\omega}_i - p_{ij}\pi^{\omega}_j + x_{ij} \geq 0, && (i,j) \in AD(s^{\omega}, t^{\omega}), \omega \in \Omega && (2.12b) \\
& \pi^{\omega}_i - q_{ij}\pi^{\omega}_j + (1 - x_{ij}) \geq 0, && (i,j) \in AD(s^{\omega}, t^{\omega}), \omega \in \Omega && (2.12c) \\
& \pi_{t^{\omega}} = 1, && \omega \in \Omega, && (2.12d)
\end{aligned}
$$

where $AD(s^{\omega}, t^{\omega}) = P^*(s^{\omega}, t^{\omega}) \cap AD$. In forming an instance of U-SNIP, we can compute $P^*(s^{\omega}, t^{\omega})$, $\omega \in \Omega$, in a pre-processing step because the evader's path does not depend on x. If the evader's optimal path in the indigenous network is not unique then further assumptions are required. One possibility is that the interdictor knows the means by which the smuggler breaks ties. Hence, $P^*(s^{\omega}, t^{\omega})$, $\omega \in \Omega$, is known. Alternatively, we could assume that the evader breaks ties arbitrarily, i.e., that every optimal path is equally likely to be selected, and simply expand the sample space and associated distribution accordingly. For example, if $\hat{\omega}$ has three optimal paths then we replace sample point $\hat{\omega}$ with $\hat{\omega}^1$, $\hat{\omega}^2$, and $\hat{\omega}^3$ with probabilities $\frac{1}{3}p^{\hat{\omega}^1}$, $\frac{1}{3}p^{\hat{\omega}^2}$, and $\frac{1}{3}p^{\hat{\omega}^3}$. Of course, other distributions governing tie-breaking can be captured similarly. So, when multiple maximum-reliability paths exist, we may still achieve a model in the form of U-SNIP.

2.5 UI-SNIP: Stochastic Network Interdiction Model for Mixed Population of Evaders

So far we have developed models in which the evaders are informed (SNIP) and in which the evaders are uninformed (U-SNIP). Next, we develop a hybrid model in which the population of possible evaders consists of both types. Let $\Omega = \Omega_U \cup \Omega_I$ partition the sample space into uninformed and informed evaders and assume p^ω, $\omega \in \Omega$, is known. Then we formulate our model UI-SNIP for the mixed population of evaders as

$$\begin{aligned}
\min_{x,\pi} \quad & \sum_{\omega \in \Omega} p^\omega \pi^\omega_{s^\omega} \\
\text{s.t.} \quad & x \in X \\
& \pi^\omega \text{ satisfies } (2.11a) - (2.11d), \quad \omega \in \Omega_I \\
& \pi^\omega \text{ satisfies } (2.12a) - (2.12d), \quad \omega \in \Omega_U.
\end{aligned}$$

In UI-SNIP, when an uninformed evader has multiple optimal paths, this can be handled as described above for U-SNIP. We note that UI-SNIP generalizes both SNIP and U-SNIP in this sense that both these models are special cases of UI-SNIP.

2.6 P-SNIP: Stochastic Network Interdiction Model for the Evader with Different Perceptions

In last three sections, we developed stochastic network interdiction models for populations that may contain some mixture of informed and uninformed evaders. Although these two types of evaders have access to different levels of information and hence react differently to interdiction decisions, they have the same perception as the interdictor of the network parameters (p_{ij} and q_{ij}) that they observe. In this section, we generalize this assumption in model P-SNIP by allowing the interdictor and evader to have different perceptions of these network parameters. In addition to the General Assumptions, we assume the following for P-SNIP.

P-SNIP Assumptions:

1. The evader and interdictor have possibly different perceptions of the evasion probabilities on the indigenous arcs and the evasion probabilities on the detector arcs.

We redefine some of the notation from Section 2.2 to accommodate two sets of perceptions.

Data:

p^1_{ij} interdictor's perception of the probability that the evader can traverse arc (i, j) undetected when no detector is installed

q^1_{ij} interdictor's perception of the probability that the evader can traverse arc (i, j) undetected when a detector is installed; $q^1_{ij} < p^1_{ij}$

p^2_{ij} evader's perception of the probability that the evader can traverse arc (i, j) undetected when no detector is installed

q^2_{ij} evader's perception of the probability that the evader can traverse arc (i, j) undetected when a detector is installed; $q^2_{ij} \leq p^2_{ij}$

Interdictor's Decision Variables:

x_{ij} 1 if a detector is installed on arc $(i,j) \in AD$ and 0 otherwise

Evader's Decision Variables:

y^1_{ij}, z^1_{ij} are analogous to y^2_{ij} and z^2_{ij} defined below except that they are used to compute the interdictor's perception of the probability the evader avoids detection

y^2_{ij} arc flow which takes positive value only if the evader perceives no detector installed on that arc (i,j) and traverses arc (i,j)

z^2_{ij} arc flow which takes positive value only if the evader perceives a detector installed on arc (i,j) and traverses arc (i,j)

Boundary Conditions:

$x_{ij} \equiv 0 \quad (i,j) \notin AD$

$z^1_{ij} \equiv 0 \quad (i,j) \notin AD$

$z^2_{ij} \equiv 0 \quad (i,j) \notin AD$

Here, General Assumption 3 means that the interdictor knows p^1_{ij}, q^1_{ij}, p^2_{ij}, and q^2_{ij} for all arcs. Hence, the interdictor knows all the information the evader will use to select what the evader perceives to be a maximum reliability s^ω–t^ω path, $\omega \in \Omega$.

P-SNIP can capture the possibility that the evader is only aware of a subset of detector locations. In particular, if $p^2_{ij} > q^2_{ij}$, the evader knows whether a sensor is installed on arc (i,j); if $p^2_{ij} = q^2_{ij}$, the evader doesn't become aware whether a sensor is installed on arc (i,j). In the latter case, q^2_{ij} and z^2_{ij} can be eliminated from the model, but for the notational simplicity we won't treat this separately.

Similar to the SNIP model, x is the first stage decision variable. After an origin-destination pair (s^ω, t^ω) is revealed, the evader chooses an s^ω–t^ω path to maximize the probability of avoiding detection, and the evader selects a path with the knowledge of sensor locations and the (p^2_{ij}/q^2_{ij}) perception on the network parameters. The interdictor reevaluates the evasion probability of the path the evader has chosen using the "true" network parameters, p^1_{ij}/q^1_{ij}. In the second stage, the interdictor's reevaluation is done with decision variables y^1 and z^1 and the evader's maximum-reliability path is selected via y^2 and z^2.

We formulate P-SNIP for the stochastic network interdiction model with different perceptions as

$$\min_{x \in X} \sum_{\omega \in \Omega} p^{\omega} h(x, (s^{\omega}, t^{\omega})), \tag{2.14}$$

where

$$X = \left\{ x \, : \, \sum_{(i,j) \in AD} a_{ij} x_{ij} \leq b, \; x_{ij} \in \{0,1\}, \, (i,j) \in AD \right\},$$

and where

$$h(x,(s^\omega,t^\omega)) =$$

$$\max_{y^1,z^1,y^2,z^2} \; y^1_{t^\omega} \tag{2.15a}$$

$$\text{s.t.} \sum_{(s^\omega,j)\in FS(s^\omega)} \left(y^1_{s^\omega j} + z^1_{s^\omega j}\right) = 1 \qquad : \pi^1_{s^\omega} \tag{2.15b}$$

$$\sum_{(i,j)\in FS(i)} (y^1_{ij} + z^1_{ij}) - \sum_{(j,i)\in RS(i)} (p^1_{ji} y^1_{ji} + q^1_{ji} z^1_{ji}) = 0, \quad i \in N \setminus \{s^\omega, t^\omega\} \quad : \pi^1_i \tag{2.15c}$$

$$y^1_{t^\omega} - \sum_{(j,t^\omega)\in RS(t^\omega)} \left(p^1_{jt^\omega} y^1_{jt^\omega} + q^1_{jt^\omega} z^1_{jt^\omega}\right) = 0 \qquad : \pi^1_{t^\omega} \tag{2.15d}$$

$$y^1_{ij} \le 1 - x_{ij}, \qquad (i,j) \in AD \qquad : \lambda^1_{ij} \tag{2.15e}$$

$$y^1_{ij} \le M y^2_{ij}, \qquad (i,j) \in A \qquad : \alpha_{ij} \tag{2.15f}$$

$$z^1_{ij} \le x_{ij}, \qquad (i,j) \in AD \qquad : \gamma^1_{ij} \tag{2.15g}$$

$$z^1_{ij} \le M y^2_{ij} + M z^2_{ij}, \qquad (i,j) \in AD \qquad : \beta_{ij} \tag{2.15h}$$

$$y^1_{ij} \ge 0, \qquad (i,j) \in A \tag{2.15i}$$

$$z^1_{ij} \ge 0, \qquad (i,j) \in AD \tag{2.15j}$$

$$(y^2, z^2) \in Y^2(x,\omega). \tag{2.15k}$$

Constraints (2.15e) and (2.15f) allow y^1_{ij} to take positive flow if there is no sensor on arc (i,j) ($x_{ij} = 0$), and arc (i,j) is on the evader's optimal path ($y^2_{ij} > 0$). In a similar way, constraints (2.15g) and (2.15h) allow z^1_{ij} to be positive if $x_{ij} = 1$ and either $y^2_{ij} > 0$ or $z^2_{ij} > 0$. We state the latter condition in this form because the evader may be unaware of a potential sensor ($p^2_{ij} = q^2_{ij}$) and could traverse (i,j) as if it has no sensor. For a given interdiction plan

x and scenario ω, $Y^2(x,\omega)$ is the argmax of the evader's maximum-reliability problem,

$$\max_{y^2,z^2} \quad y^2_{t^\omega} \tag{2.16a}$$

$$\text{s.t.} \quad \sum_{(s^\omega,j)\in FS(s^\omega)} \left(y^2_{s^\omega j} + z^2_{s^\omega j}\right) = 1 \tag{2.16b}$$

$$\sum_{(i,j)\in FS(i)} (y^2_{ij} + z^2_{ij}) - \sum_{(j,i)\in RS(i)} (p^2_{ji} y^2_{ji} + q^2_{ji} z^2_{ji}) = 0, \quad i \in N \setminus \{s^\omega, t^\omega\} \tag{2.16c}$$

$$y^2_{t^\omega} - \sum_{(j,t^\omega)\in RS(t^\omega)} \left(p^2_{jt^\omega} y^2_{jt^\omega} + q^2_{jt^\omega} z^2_{jt^\omega}\right) = 0 \tag{2.16d}$$

$$0 \le y^2_{ij} \le 1 - x_{ij}, \qquad (i,j) \in AD \tag{2.16e}$$

$$0 \le z^2_{ij} \le x_{ij}, \qquad (i,j) \in AD. \tag{2.16f}$$

In contrast to SNIP (2.1) with second-stage problem (2.2), in P-SNIP the second-stage problem is a bilevel program. The P-SNIP model generalizes the models of Sections 2.3–2.5. In (2.14), if $p^2 = p^1$ and $q^2 = q^1$, P-SNIP becomes SNIP for the informed evader; if $p^2 = p^1$ and $q^2 = p^1$, P-SNIP becomes U-SNIP for the uninformed evader. The mixed-population model can be captured by appending an ω index to $q^{2,\omega}$ and making the former or latter assignments for $\omega \in \Omega_I$ and $\omega \in \Omega_U$, respectively.

To solve (2.14), we seek to reformulate the model as a single non-nested mathematical program. Since the second-stage problem is a bilevel optimization problem, we first convert it to a single mathematical program. In our problem, the feasible region of the interdictor's second-stage problem (2.15) depends on the evader's solution (y^2, z^2). The evader's maximization problem is of the same form as (2.2) in Section 2.3, and the equivalent dual formulation is given as (2.9) in Theorem 2.3.2. We enforce $(y^2, z^2) \in Y^2(x, \omega)$ by replacing it with

$$\sum_{(s^\omega, j) \in FS(s^\omega)} \left(y^2_{s^\omega j} + z^2_{s^\omega j}\right) = 1 \qquad : \quad \pi^3_{s^\omega} \tag{2.17a}$$

$$\sum_{(i,j) \in FS(i)} (y^2_{ij} + z^2_{ij}) - \sum_{(j,i) \in RS(i)} (p^2_{ji} y^2_{ji} + q^2_{ji} z^2_{ji}) = 0, \qquad i \in N \setminus \{s^\omega, t^\omega\} \qquad : \quad \pi^3_i \tag{2.17b}$$

$$y^2_{t^\omega} - \sum_{(j,t^\omega) \in RS(t^\omega)} \left(p^2_{jt^\omega} y^2_{jt^\omega} + q^2_{jt^\omega} z^2_{jt^\omega}\right) = 0 \qquad : \quad \pi^3_{t^\omega} \tag{2.17c}$$

$$0 \le y^2_{ij} \le 1 - x_{ij}, \qquad (i,j) \in AD \qquad : \quad \lambda^3_{ij} \tag{2.17d}$$

$$0 \le z^2_{ij} \le x_{ij}, \qquad (i,j) \in AD \qquad : \quad \gamma^3_{ij} \tag{2.17e}$$

$$\pi^\omega_i - p^2_{ij} \pi^\omega_j \ge 0, \quad (i,j) \in A \setminus AD \qquad : -y^3_{ij} \tag{2.17f}$$

$$\pi^\omega_i - p^2_{ij} \pi^\omega_j + x_{ij} \ge 0, \qquad (i,j) \in AD \qquad : -y^3_{ij} \tag{2.17g}$$

$$\pi^\omega_i - q^2_{ij} \pi^\omega_j + (1 - x_{ij}) \ge 0, \qquad (i,j) \in AD \qquad : -z^3_{ij} \tag{2.17h}$$

$$\pi_{t^\omega} = 1 \qquad : -y^3_{t^\omega} \tag{2.17i}$$

$$\pi_{s^\omega} = y_{t^\omega} \qquad : \quad \theta \tag{2.17j}$$

Constraints (2.17a)–(2.17e) describe the primal feasible region, and constraints (2.17f)–(2.17i) provide the dual feasible region. Constraint (2.17j) is the condition for strong duality. Since (2.16) is a LP, for any element $(y^2, z^2) \in Y^2(x, \omega)$, there exists a dual optimal solution π such that (y^2, z^2, π) satisfies (2.17).

Therefore,

$$h(x, (s^\omega, t^\omega)) = \max_{y^1, z^1, y^2, z^2, \pi} \quad y^1_{t^\omega} \tag{2.18a}$$

$$\text{s.t.} \quad (y^1, z^1) \;\; \text{satisfies } (2.15b) - (2.15j) \tag{2.18b}$$

$$(y^2, z^2, \pi) \;\; \text{satisfies } (2.17). \tag{2.18c}$$

With (2.18) the second-stage problem is expressed as a linear program, and the resulting two-stage network interdiction model has a "min-max" structure. To transform the model to a single mathematical program, we will take the dual of the second-stage model, but first we replace (2.17g) with

$$\pi^\omega_i - p^2_{ij}\pi^\omega_j + \hat{\lambda}_{ij} \geq 0, \qquad (i,j) \in AD \tag{2.19a}$$

$$0 \leq \hat{\lambda}_{ij} \leq x_{ij}, \qquad (i,j) \in AD. \tag{2.19b}$$

Similarly, we replace (2.17h) with

$$\pi^\omega_i - q^2_{ij}\pi^\omega_j + \hat{\gamma}_{ij} \geq 0, \qquad (i,j) \in AD \tag{2.20a}$$

$$0 \leq \hat{\gamma}_{ij} \leq 1 - x_{ij}, \qquad (i,j) \in AD. \tag{2.20b}$$

In Section 2.3 we applied Lemma 2.3.1 to the second-stage linear program (2.2) in order to avoid nonlinearities which would otherwise arise when taking the dual of (2.2). With analogous motivation, we apply Lemma 2.3.1 to (2.18) and specifically to the simple bound constraints (2.15e), (2.15g), (2.17d), (2.17e), (2.19b) and (2.20b). As a result the objective function of (2.18) is replaced by

$$y^1_{t^\omega} - \sum_{(i,j)\in AD} (y^1_{ij} + y^2_{ij} + \hat{\gamma}_{ij})x_{ij} + (z^1_{ij} + z^2_{ij} + \hat{\lambda}_{ij})(1 - x_{ij}), \tag{2.21}$$

and the dual of the resulting linear program is

$$
\begin{aligned}
h(x,(s^\omega,t^\omega)) = \quad & \min_{\pi^1,y^3,z^3,\pi^3,\theta,\alpha,\beta} \pi^1_{s^\omega} + \pi^3_{s^\omega} - y^3_{t^\omega} \\
\text{s.t.} \quad & \pi^1_i - p^1_{ij}\pi^1_j + \alpha_{ij} \geq 0, \qquad (i,j) \in A \setminus AD && \text{(2.22a)} \\
& \pi^1_i - p^1_{ij}\pi^1_j + x_{ij} + \alpha_{ij} \geq 0, \qquad (i,j) \in AD && \text{(2.22b)} \\
& \pi^1_i - q^1_{ij}\pi^1_j + (1 - x_{ij}) + \beta_{ij} \geq 0, \qquad (i,j) \in AD && \text{(2.22c)} \\
& \pi^1_{t^\omega} \geq 1 && \text{(2.22d)} \\
& \pi^3_i - p^2_{ij}\pi^3_j - M\alpha_{ij} \geq 0, \quad (i,j) \in A \setminus AD && \text{(2.22e)} \\
& \pi^3_i - p^2_{ij}\pi^3_j + x_{ij} - M(\alpha_{ij} + \beta_{ij}) \geq 0, \; (i,j) \in AD && \text{(2.22f)} \\
& \pi^3_i - q^2_{ij}\pi^3_j + (1 - x_{ij}) - M\beta_{ij} \geq 0, \quad (i,j) \in AD && \text{(2.22g)} \\
& \pi^3_{t^\omega} - \theta \geq 0 && \text{(2.22h)} \\
& \sum_{(s^\omega,j)\in FS(s^\omega)} \left(y^3_{s^\omega j} + z^3_{s^\omega j}\right) = \theta && \text{(2.22i)} \\
& \sum_{(i,j)\in FS(i)} (y^3_{ij} + z^3_{ij}) - \sum_{(j,i)\in RS(i)} (p^3_{ji}y^3_{ji} + q^2_{ji}z^3_{ji}) = 0, \\
& i \in N \setminus \{s^\omega, t^\omega\} && \text{(2.22j)} \\
& y^3_{t^\omega} - \sum_{(j,t^\omega)\in RS(t^\omega)} \left(p^2_{jt^\omega}y^3_{jt^\omega} + q^2_{jt^\omega}z^3_{jt^\omega}\right) = 0 && \text{(2.22k)} \\
& 0 \leq y^3_{ij} \leq 1 - x_{ij}, \qquad (i,j) \in AD && \text{(2.22l)} \\
& 0 \leq z^3_{ij} \leq x_{ij}, \qquad (i,j) \in AD && \text{(2.22m)} \\
& \alpha_{ij} \geq 0, \beta_{ij} \geq 0, \qquad (i,j) \in AD. && \text{(2.22n)}
\end{aligned}
$$

Now, we have a "min-min" structure as we can combine (2.14) and (2.22) into a single mixed-integer linear program.

We have introduced several stochastic network interdiction models which involve different information levels and different perceptions of the network parameters. In the remainder of the dissertation, we will focus on the stochastic network interdiction model for an informed evader (SNIP) and we first discuss an important special case on a bipartite network.

2.7 Bipartite Stochastic Network Interdiction Model

In certain applications, an evader attempts to smuggle material out of a single country, and the interdictor is restricted to install detectors on the border of that country. In such cases, when the origin is inside the country and the destination outside the country, we assume the smuggler will encounter at most one sensor, namely when crossing the country's border. As discussed in the introduction, the SLD program motivates such an example for material stored within Russia and with detectors installed on Russia's border, including airports, seaports, and land crossings.

Our underlying transportation network model for a single country has four basic location entities: *sources* from which sensitive material could be stolen, geographic *provinces*, *destinations* outside the country where a evader may desire to go, and *border checkpoints* or *border crossing* where sensors can be installed. The nominal transportation network has a node representing each of these locations (some aggregation is possible as we describe below). These nodes are linked by arcs representing transport by surface roads, railroads, airline flights, ship transport, etc. A border checkpoint is modeled by two nodes with an associated arc which represents traveling through the check-

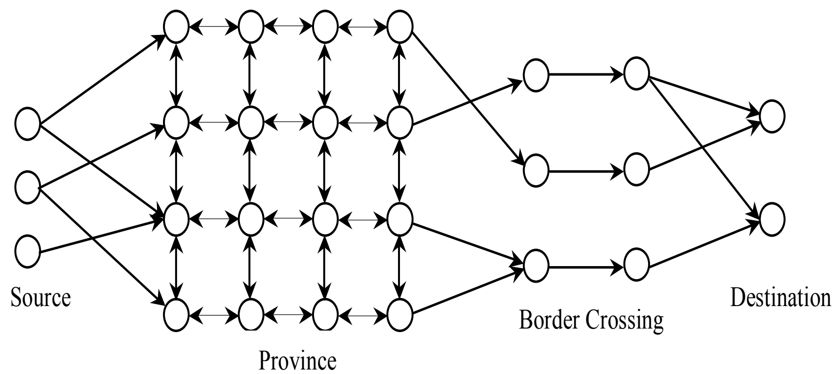

Figure 2.2: The graph shows the interdiction network for a single country.

point. Figure 2.2 illustrates four location entities and the connectivity among them.

The key to simplifying the formulation, when we consider the border checkpoints of a single country, is that on each potential smuggling route (i.e., each possible s^ω-t^ω path) there is exactly one arc on which the smuggler could encounter a sensor. We formalize this in the following manner: Let $\mathcal{P}^\omega$ be the set of all paths for origin-destination pair (s^ω, t^ω). (These paths need not be enumerated.) Then, in our BiSNIP model (bipartite SNIP, for reasons soon apparent) we assume that each path in $\mathcal{P}^\omega$ contains exactly one arc in AD, i.e., each path has exactly one arc that is a candidate to receive a sensor. Let $AD^\omega = \{(i,j) : (i,j) \in AD, (i,j) \in \mathcal{P}^\omega\}$ be all such checkpoint arcs for $\omega \in \Omega$. The evader, under scenario ω, must select an s^ω-t^ω path, but this now depends on the sensor locations in a much simpler way than in the SNIP model. For each ω, we perform a preprocessing step to compute the value of the maximum-reliability path from s^ω to the tail of each checkpoint arc and the value of the maximum-reliability path from the head of each checkpoint arc to t^ω. Call the product of these two probabilities γ_c^ω, $c = (i,j) \in AD^\omega$. Figure 2.3 shows the preprocessing steps to transform the network in Figure 2.2 to a bipartite network. Then, the value of the maximum-reliability path under scenario ω is

$$h(x, (s^\omega, t^\omega)) = \max_{c \in AD^\omega} \left\{ \gamma_c^\omega p_c (1 - x_c), \gamma_c^\omega q_c x_c \right\}. \tag{2.23}$$

By linearizing (2.23), we can express BiSNIP as the following mixed-integer linear program

$$\begin{aligned}
\min_{x,\theta} \quad & \sum_{\omega \in \Omega} p^\omega \theta^\omega \\
\text{s.t.} \quad & \sum_{c \in AD} a_c x_c \leq b && (2.24a) \\
& \theta^\omega \geq \gamma_c^\omega p_c (1 - x_c), \qquad c \in AD^\omega, \quad \omega \in \Omega && (2.24b) \\
& \theta^\omega \geq \gamma_c^\omega q_c x_c, \qquad c \in AD^\omega, \quad \omega \in \Omega && (2.24c) \\
& x_c \in \{0, 1\}, \qquad c \in AD. && (2.24d)
\end{aligned}$$

BiSNIP (2.24) may be visualized on an underlying bipartite network with node sets Ω and AD as shown in Figure 2.3(b). Arcs (ω, c) link each scenario $\omega \in \Omega$ (facility-destination pair) with its possible intermediate checkpoints, $c \in AD^\omega$. Excluding the possibility of being detected at the checkpoint, γ_c^ω is the evader's probability of traveling from ω's facility to ω's destination, via c, undetected. This reliability is multiplied by q_c or p_c depending on whether a detector is

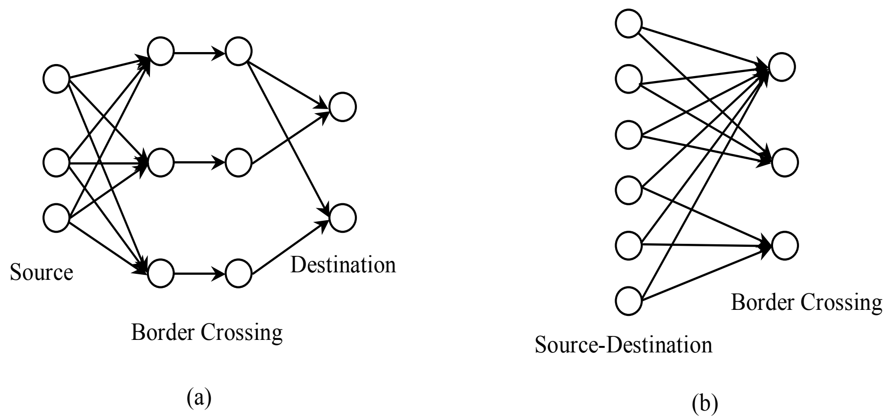

Figure 2.3: (a) The graph shows that preprocessing reduces the network to two pieces, from a source to the tail of a bordering-crossing arc and from the head of that bordering-crossing arc to a destination. (b) The graph shows the bipartite network structure and the connectivity between each scenario ω and each border crossing c.

installed at checkpoint c. In our computational work, we will assume $a_{ij} = 1$ for all interdictable arcs and in this case (2.24a) simply reduces to a cardinality constraint, where b is the number of sensors that can be installed.

2.8 Complexity

In this section, we establish the complexity of our stochastic network interdiction models. First, we prove that BiSNIP is NP-Complete by transforming the Clique problem to the corresponding decision problem BiSNIP-Decision.

The following defines the Clique decision problem:

CLIQUE:

INSTANCE: Graph $G(V, E)$, a positive integer $k \leq |V|$.

QUESTION: Does G have a clique of size k or more, that is, a subset $W \subseteq V$ such that $|W| \geq k$ and every two vertices in W are joined by an edge in E?

Here, we consider a special case of BiSNIP (2.24). We assume unit installation costs, $a_c = 1$, $c \in AD$, and "certain" interdiction success probabilities at each customs site, $q_c = 0$, $c \in AD$. We also assume a uniform underlying distribution, $p^\omega = \frac{1}{|\Omega|}$, $\omega \in \Omega$. Let $r_c^\omega = \gamma_c^\omega p_c$, $c \in AD$. The MIP formulation of this special case of BiSNIP is

$$\begin{aligned} \min_{x,\theta} \quad & \sum_{\omega \in \Omega} \theta^\omega \\ \text{s.t.} \quad & \sum_{c \in AD} x_c \leq b \\ & \theta^\omega \geq r_c^\omega (1 - x_c), \qquad c \in AD^\omega, \quad \omega \in \Omega \\ & x_c \in \{0, 1\}, \qquad c \in AD. \end{aligned} \tag{2.25}$$

Because of the uniform distribution, we have omitted p^ω in the objective function in (2.25) for simplicity. The decision problem of (2.25) is

BiSNIP-DECISION:

INSTANCE: Bipartite graph $B(\Omega, AD, A)$, positive arc gain r_c^ω, $(\omega, c) \in A$, a positive integer b, and a positive real α.

QUESTION: Does there exist a subset $S \subseteq AD$ of size b such that if S is removed from AD, then

$$\sum_{\omega \in \Omega} \theta^{\omega} \leq \alpha,$$

where $\theta^{\omega} = \max\{r_c^{\omega} : (\omega, c) \in A, c \in AD \setminus S\}$?

Theorem 2.8.1. *BiSNIP*-DECISION *is strongly NP-Complete.*

Proof. We first establish that BiSNIP-DECISION belongs to the class NP, and then reduce CLIQUE, which is known to be NP-Complete in the strong sense, to BiSNIP-DECISION. To see that BiSNIP-DECISION belongs to NP, note that a polynomial-length guess for an instance of BiSNIP-DECISION consists of $S \subseteq AD$ with $|S| = b$. To test whether or not such a guess verifies a BiSNIP-DECISION instance as a yes-instance, we can evaluate $\sum_{\omega \in \Omega} \max_{\substack{(\omega,c) \in A \\ c \in AD \setminus S}} \{r_{ij}\}$ and then do a simple comparison. So, the number of steps required to verify a guess is bounded by $O(|\Omega||AD|)$, which proves that BiSNIP-DECISION is in NP.

Next, we transform CLIQUE to BiSNIP-DECISION. We construct a bipartite graph $B^G(\Omega^G, AD^G, A^G)$ from $G(V, E)$ in the statement of CLIQUE as follows. For each node $v \in V$, create a node $v \in AD^G$. For each edge $e \in E$, create a node $e \in \Omega^G$. The bipartite network arcs are constructed as follows: for each edge $e = (u, v) \in E$, create arcs $(e, u) \in A^G$ with $r_u^e = 1$ and $(e, v) \in A^G$ with $r_v^e = 1$. Finally, set the cardinality restriction $b = k$ and objective target $\alpha = |E| - \binom{k}{2}$.

We must show that the original CLIQUE instance is a yes-instance if and only if the transformed BiSNIP-DECISION instance is a yes-instance. In B^G, with $S = \emptyset$

$$\begin{aligned}\sum_{\omega \in \Omega^G} \theta^\omega &= \sum_{\omega \in \Omega^G} \max_{\substack{(\omega,c) \in A^G \\ c \in AD^G}} \{r_c^\omega\} \\ &= \sum_{e=(u,v) \in E} \max\{r_u^e, r_v^e\} \\ &= |E|.\end{aligned}$$

Suppose the BiSNIP-DECISION instance is a yes-instance. Then, there exists a set $S \subseteq AD^G$ with $|S| = k$. After removing S from AD^G,

$$\begin{aligned}\sum_{\omega \in \Omega^G} \theta^\omega &= \sum_{e=(u,v) \in E} \max\{r_u^e, r_v^e : u, v \in AD^G \setminus S\} \\ &= \sum_{e=(u,v) \in E} 1 - \sum_{\substack{e=(u,v) \in E \\ u,v \in S}} 1 \\ &\leq |E| - \binom{k}{2}.\end{aligned}$$

Since BiSNIP-DECISION instance is a yes-instance, the last inequality is valid. This implies that $\sum_{\substack{e=(u,v) \in E \\ u,v \in S}} 1 \geq \binom{k}{2}$. Let $T = \{e = (u, v) \in E : u, v \in S\}$. Since there are at most $\binom{k}{2}$ arcs for a graph with k nodes, $|T| = \binom{k}{2}$ with $|S| = k$. Therefore, S is a clique of size k in graph G, and the original CLIQUE instance is a yes-instance.

Now, suppose the CLIQUE instance is a yes-instance. Let S be a clique of size k. Let $T = \{e = (u, v) \in E : u, v \in S\}$, then $|T| = \binom{k}{2}$. In the induced bipartite graph, B^G,

$$\begin{aligned}
\sum_{\omega \in \Omega^G} \theta^{\omega} &= \sum_{\omega \in \Omega^G} \max_{\substack{(\omega,c) \in A^G \\ c \in AD^G \setminus S}} \{r_c^{\omega}\} \\
&= \sum_{e=(u,v) \in E} 1 - \sum_{e=(u,v) \in T} 1 \\
&= |E| - \binom{k}{2}.
\end{aligned}$$

Therefore, if we remove node set S from AD^G then BiSNIP-Decision is a yes-instance.

Since the foregoing transformation can be accomplished in polynomial time, and all of the data from the transformation is polynomially bounded, we have that BiSNIP-Decision is strongly NP-Complete. □

The theorem establishes that BiSNIP-Decision is strongly NP-Complete and, in this sense we can regard (2.29) as being strongly NP-Complete. BiSNIP is a special case of SNIP, and hence SNIP is strongly NP-Complete. For a direct proof that SNIP, or rather its related decision problem, is NP-Complete we refer to [49].

2.9 Conclusion

In this chapter, we introduced several stochastic network interdiction models to minimize the chance of a successful smuggling attempt. The models differ with respect to the information available to the adversaries, the relative perceptions of the interdictor and the adversary of the network's parameters, and where in the network detectors can be placed. In subsequent chapters, we will discuss solution techniques for BiSNIP and SNIP.

Chapter 3

Solution Methods For BiSNIP

3.1 Introduction

In this chapter, we discuss solution methods for BiSNIP, the special case of SNIP in which detectors can only be installed on the boundary of the region in question. BiSNIP, introduced in Section 2.7, is formulated as the stochastic mixed-integer linear program

$$\min_{x \in X} E[\max_{c \in AD^{\omega}} \{\gamma_c^{\omega} p_c(1 - x_c), \gamma_c^{\omega} q_c x_c\}], \tag{3.1}$$

where $x \in X$ includes the budget constraint and binary restrictions on x, and where γ_c^{ω} is the evasion probability on the most reliable path from ω's origin through border crossing c to ω's destination not including the border crossing evasion probability p_c or q_c on the indigenous border-crossing arc or detector border-crossing arc.

In this chapter, we develop techniques to solve BiSNIP via the branch-and-bound method. We mainly focus on preprocessing techniques, structural properties of the feasible region for BiSNIP, and a class of valid inequalities for tightening the LP relaxation of BiSNIP. In Section 3.2, we introduce basic properties of BiSNIP and explore some preprocessing techniques to reduce the number of constraints and binary variables. Section 3.3 introduces structure dependent valid inequalities, which we term *step inequalities.* We start with the *two-step* inequality, then we derive the step inequality with arbitrary number of steps. We provide conditions under which step inequalities are facet-defining. And, we give an algorithm to detect violated step inequalities and add them in the course of solving BiSNIP. Finally, we provide computational results to show the improvement of applying step inequalities.

3.2 Basic Formulation and Preprocessing

BiSNIP introduced in Section 2.7 has the following MIP formulation,

$$\begin{aligned}
\min_{x,\theta} \quad & \sum_{\omega\in\Omega} p^\omega\theta^\omega \\
\text{s.t.} \quad & \sum_{c\in AD} a_c x_c \le b \\
& \theta^\omega \ge \gamma_c^\omega p_c(1-x_c), \qquad c\in AD^\omega, \quad \omega\in\Omega \\
& \theta^\omega \ge \gamma_c^\omega q_c x_c, \qquad c\in AD^\omega, \quad \omega\in\Omega \\
& x_c \in \{0,1\}, \qquad c\in AD.
\end{aligned} \tag{3.2}$$

Let $\hat{q}^\omega = \max\limits_{c\in AD^\omega} \gamma_c^\omega q_c$, $\omega\in\Omega$. Then, $\hat{q}^\omega$ provides a lower bound for θ^ω. Let

$$r_c^\omega = \begin{cases} \gamma_c^\omega p_c - \hat{q}^\omega & \text{if } \gamma_c^\omega p_c > \hat{q}^\omega \\ 0 & \text{if otherwise,} \end{cases} \tag{3.3}$$

$c\in AD^\omega$. In the context of nuclear smuggling interdiction, $\hat{q}^\omega$ is the optimal evasion probability in scenario ω with monitors being installed at all possible interdictable arcs, and r_c^ω is the marginal increase in the evasion probability due to selecting an uninterdicted path through c over $\hat{q}^\omega$. Here, we assume that $a_c = 1$, $c\in AD$ and that the budget b is a nonnegative integer. Then, BiSNIP becomes

$$
\begin{aligned}
\min_{x,\theta} \quad & \sum_{\omega\in\Omega} p^\omega \theta^\omega \\
\text{s.t.} \quad & \sum_{c\in AD} x_c \le b && && \text{(3.4a)}\\
& \theta^\omega \ge r_c^\omega(1-x_c), && c\in AD^\omega, \quad \omega\in\Omega && \text{(3.4b)}\\
& x_c \in \{0,1\}, && c\in AD. && \text{(3.4c)}
\end{aligned}
$$

By BiSNIP we mean formulation (3.4) in the rest of this dissertation.

Lemma 3.2.1. *With $a_c = 1$, $c \in AD$, formulation (3.2) and (3.4) have the same set of optimal solutions and optimal values that differ by the constant* $\sum_{\omega\in\Omega} p^\omega \hat{q}^\omega$.

Proof. For $\omega \in \Omega$, since $p_c \ge q_c$ and $\gamma_c^\omega \ge 0$, $c \in AD$, $\gamma_c^\omega p_c \ge \gamma_c^\omega q_c$. Since $\hat{q}^\omega = \max_{c\in AD^\omega} \gamma_c^\omega q_c$, $\omega \in \Omega$, constraints

$$\theta^\omega \ge \gamma_c^\omega q_c x_c, \quad c \in AD^\omega, \quad \omega \in \Omega$$

can be replaced by

$$\theta^\omega \ge \hat{q}^\omega, \quad \omega \in \Omega.$$

Let $\hat{\theta}^\omega = \theta^\omega - \hat{q}^\omega$. We have $\hat{\theta}^\omega \ge 0$, and constraint $\theta^\omega \ge \gamma_c^\omega p_c(1-x_c)$ becomes

$$\hat{\theta}^\omega \ge \gamma_c^\omega p_c(1-x_c) - \hat{q}^\omega. \tag{3.5}$$

For a fixed x,

$$\gamma_c^\omega p_c(1-x_c) - \hat{q}^\omega = \begin{cases} \gamma_c^\omega p_c - \hat{q}^\omega & \text{if } x_c = 0 \\ -\hat{q}^\omega & \text{if } x_c = 1. \end{cases}$$

Since $\hat{\theta} \geq 0$, inequality (3.5) can be simplified to

$$\hat{\theta}^\omega \geq r_c^\omega (1 - x_c),$$

where r_c^ω is defined in (3.3). For an arbitrary feasible x, since x is binary, the objective value of (3.4) is

$$\begin{aligned}
\sum_{\omega \in \Omega} p^\omega \max_{c \in AD^\omega} \{r_c^\omega (1 - x_c)\} &= \sum_{\omega \in \Omega} p^\omega \max_{c \in AD^\omega} \{\max\{(\gamma_c^\omega p_c - \hat{q}^\omega), 0\}(1 - x_c)\} \\
&= \sum_{\omega \in \Omega} p^\omega \max_{c \in AD^\omega} \{\max\{\gamma_c^\omega p_c (1 - x_c) - \hat{q}^\omega, 0\}\} \\
&= \sum_{\omega \in \Omega} p^\omega \max\{\max_{c \in AD^\omega} \{\gamma_c^\omega p_c (1 - x_c)\} - \hat{q}^\omega, 0\} \\
&= \sum_{\omega \in \Omega} p^\omega (\max\{\max_{c \in AD^\omega} \{\gamma_c^\omega p_c (1 - x_c)\}, \hat{q}^\omega\} - \hat{q}^\omega) \\
&= \sum_{\omega \in \Omega} p^\omega \max\{\max_{c \in AD^\omega} \{\gamma_c^\omega p_c (1 - x_c)\}, \hat{q}^\omega\} - \sum_{\omega \in \Omega} p^\omega \hat{q}^\omega.
\end{aligned}$$

Therefore, formulations (3.2) and (3.4) have the same set of optimal solutions, and the optimal values of these two problems differ by the constant $\sum_{\omega \in \Omega} p^\omega \hat{q}^\omega$.

□

Let BiSNI be the feasible region of BiSNIP (3.4),

$$\begin{aligned}
\text{BiSNI } = \{(x, \theta) : \quad & \sum_{c \in AD} x_c, \leq b,\ x_c \in \{0,1\},\ c \in AD^\omega,\ \theta^\omega \geq 0,\ \omega \in \Omega, \\
& \theta^\omega \geq r_c^\omega (1 - x_c),\ c \in AD^\omega,\ \omega \in \Omega\},
\end{aligned}$$

and let $\bar{r}^\omega = \max_{c \in AD^\omega} r_c^\omega$.

Proposition 3.2.2.

1. *The dimension of* conv*(BiSNI) is* $|AD| + |\Omega|$ *for* $b = 1, \cdots, n$.

2. *For* $b = 1, \cdots, n$*,* $x_c \geq 0$ *defines a facet for* conv*(BiSNI),* $c \in AD$*.*

3. *For* $b = 2, \cdots, n$*,* $x_c \leq 1$ *defines a facet for* conv*(BiSNI),* $c \in AD$*.*

4. *If* $\rho^\omega \geq \bar{r}^\omega$ *then* $\theta^\omega \leq \rho^\omega$ *defines a facet for* conv*(BiSNI),* $\omega \in \Omega$*.*

5. $\sum_{c \in AD} x_c \leq b$ *defines a facet for* conv*(BiSNI) if* $b = 1, \cdots, n-1$*.*

6. $\theta^\omega \geq r_c^\omega(1 - x_c)$ *defines a facet for* conv*(BiSNI) if and only if* $|AD^\omega| = 1$ *and* $b = 1, \cdots, n-1$*.*

Proof. We only prove part 2 as proofs of the other parts are similar. For $b \in \{1, \cdots, n\}$ and $c \in AD$, we show that inequality $x_c \geq 0$ defines a facet for conv(BiSNI) by constructing $|AD| + |\Omega|$ affinely independent feasible points (v^0, v^ω, $\omega \in \Omega$, and v^i, $i \in AD \setminus \{c\}$) at which $x_c = 0$. All point are in the form of (x, θ), and e_i is the $(|AD| + |\Omega|)$-dimension unit vector which has 1 in the position of element i and 0 elsewhere. Here are $|AD| + |\Omega|$ points,

$$\begin{aligned} v^0 &= \sum_{\omega \in \Omega} \bar{r}^\omega e_\omega, \\ v^\omega &= \sum_{\omega' \in \Omega \setminus \{\omega\}} \bar{r}^{\omega'} e_{\omega'} + (\bar{r}^\omega + 1) e_\omega, \quad \omega \in \Omega, \\ v^i &= \sum_{\omega \in \Omega} \bar{r}^\omega e_\omega + e_i, \quad i \in AD \setminus \{c\}. \end{aligned} \tag{3.6}$$

In all of the above points, $\theta^\omega \geq \bar{r}^\omega$, $\omega \in \Omega$, and there is at most one component of x at 1 with the others of 0. So, these points are in conv(BiSNI). Also, $x_c = 0$ in each point, and so inequality $x_c \geq 0$ holds with equality.

Let $u^k = v^k - v^0$ for $k \in AD \cup \Omega$, then

$$
\begin{aligned}
u^\omega &= e_\omega, \quad \omega \in \Omega, \\
u^i &= e_i, \quad i \in AD \setminus \{c\}.
\end{aligned}
$$

Since the u's are linearly independent points, the points in (3.6) are affinely independent. Therefore, inequality $x_c \geq 0$ defines a facet for conv(BiSNIP).

□

3.3 Step Inequalities

Coefficient reduction, probing and cutting planes are frequently used to preprocess and improve an MIP formulation. Before solving a mixed-integer linear program, we want to tighten its LP relaxation and reduce the integrality gap between the mixed-integer linear program and its LP relaxation so that branch-and-bound methods will require less computational time. In this section, we derive a class of structure-dependent valid inequalities, similar to the star inequality in [3], to tighten the BiSNIP formulation.

3.3.1 BiSNIP's LP Relaxation

To understand how we can tighten BiSNIP's formulation via valid inequalities, we start with the single scenario problem, formulated as follows

$$
\begin{aligned}
z^1(b) = \min_{x,\theta} \quad & \theta \\
\text{s.t.} \quad & \sum_{c \in AD} x_c \le b && (3.7a) \\
& \theta \ge r_c(1 - x_c), \quad c \in AD && (3.7b) \\
& x_c \in \{0, 1\}, \qquad c \in AD. && (3.7c)
\end{aligned}
$$

Without loss of generality, assume $r_{c_1} \ge r_{c_2} \ge \cdots \ge r_{c_{|AD|}} > 0$ and let $r_{c_{|AD|+1}} = 0$. For $b \in \{0, \cdots, |AD|\}$, $z^1(b) = r_{b+1}$. In an optimal solution of (3.7), $x^*_{c_i} = 1$ for $1 \le i \le b$ and $x^*_{c_1} = 0$ for $c_i > b$. The following ordering inequality

$$x_c \ge x_{c'} \text{ if } r_c \ge r_{c'} \tag{3.8}$$

is satisfied at x^* for any pair of $c, c' \in AD$, $c \ne c'$. Let $z^1_{LP}(b)$ be the optimal objective value of the LP relaxation of (3.7). For the given $z^1_{LP}(b)$, we can find an index c_k such that $r_{c_k} > z^1_{LP}(b) \ge r_{c_{k+1}}$, and we can write $z^1_{LP}(b)$ as

$$z^1_{LP}(b) = \frac{k - b}{\sum_{i=1}^{k} \frac{1}{r_{c_i}}}.$$

Then, an optimal solution x^{LP*} of the LP relaxation problem is

$$x^{LP*}_c = \begin{cases} (r_c - z^1_{LP}(b))/r_c & \text{if } r_c > z^1_{LP}(b) \\ 0 & \text{if } r_c \le z^1_{LP}(b). \end{cases}$$

The LP relaxation solution x^{LP*} also satisfies the ordering inequality (3.8). In fact, for any $b < |AD|$, $x^{LP*}_c < 1$ for $c \in AD$. The following example illustrates the difference between objective values of the single scenario problem, (3.7), and its LP relaxation.

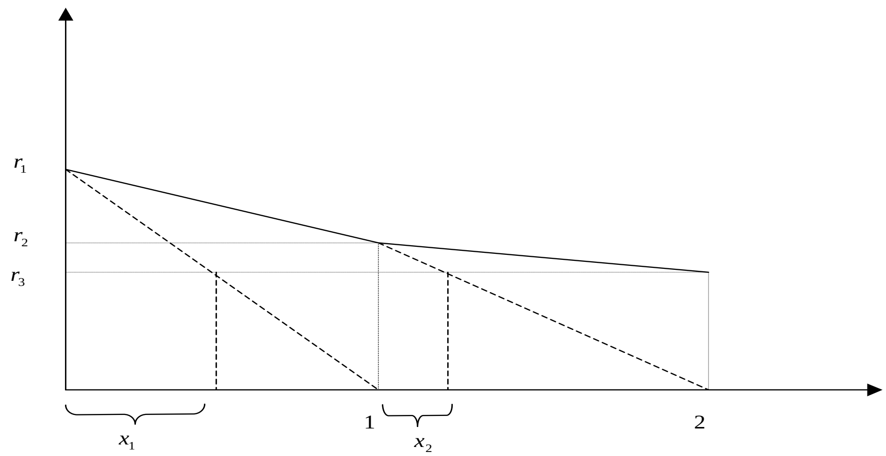

Figure 3.1: The graph shows the difference between the MIP solution and its LP relaxation solution for a single-scenario BiSNIP.

Example 3.3.1. *Let $r_1 > r_2 > r_3 > 0$. We have the following problem*

$$\begin{aligned} \min_{x,\theta} \quad & \theta \\ \text{s.t.} \quad & x_1 + x_2 + x_3 \leq 2 \\ & \theta \geq r_1(1 - x_1) \\ & \theta \geq r_2(1 - x_2) \\ & \theta \geq r_3(1 - x_3) \\ & x_1, x_2, x_3 \in \{0, 1\}. \end{aligned} \tag{3.9}$$

The optimal solution of (3.9) is $(x^*, \theta^*) = (1, 1, 0, r_3)$. But in its LP relaxation, to achieve $\theta = r_3$, we only require that $x_1^{LP} \geq \frac{r_1 - r_3}{r_1}$ and $x_2^{LP} \geq \frac{r_2 - r_3}{r_2}$ (see Figure 3.1). Thus, with $x_1^{LP} + x_2^{LP} = \frac{r_1 - r_3}{r_1} + \frac{r_2 - r_3}{r_2} < 2$, we can already achieve $\theta = r_3$ in the LP relaxation.

□

In the related stochastic version of problem (3.4), let (x^{LP*}, θ^{LP*}) be an optimal solution for the LP relaxation of BiSNIP.

Theorem 3.3.1. *Let (x^{LP*}, θ^{LP*}) be an optimal solution for the LP relaxation of BiSNIP. Then,*

$$x_c^{LP*} = \begin{cases} \max\{\frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega} : \omega \in \Omega, r_c^\omega > \theta^{\omega LP*}\} & \text{if} \quad \exists \omega \in \Omega, r_c^\omega > \theta^{\omega LP*} \\ 0 & \text{if} \quad r_c^\omega \leq \theta^{\omega LP*}, \forall \omega \in \Omega. \end{cases} \tag{3.10}$$

Proof. Without loss of generality, assume $r_c^\omega > 0$, $c \in AD^\omega$ and $\omega \in \Omega$. Suppose there exists $\omega \in \Omega$ and $r_c^\omega > \theta^{\omega LP*}$, and $x_c^{LP*} < \frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega}$. Then, constraint (3.4b) for (ω, c) pair becomes

$$\theta^{\omega LP*} \geq r_c^\omega (1 - x_c^{LP*}) > \theta^{\omega LP*},$$

which contradicts that (x^{LP*}, θ^{LP*}) is a feasible solution. Therefore, $x_c^{LP*} \geq \frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega}$, for $\omega \in \Omega$ with $r_c^\omega > \theta^{\omega LP*}$.

Suppose $\epsilon = x_{c'}^{LP*} - \max\left\{\frac{r_{c'}^\omega - \theta^{\omega LP*}}{r_{c'}^\omega} : \omega \in \Omega, r_{c'}^\omega > \theta^{\omega LP*}\right\} > 0$ for some $c' \in AD$. Let $\hat{\theta} = \theta^{LP*}$, $\hat{x}_c = x_c^{LP*}$, $c \in AD \setminus \{c'\}$, and $\hat{x}_{c'} = x_{c'}^{LP*} - \epsilon$. Then, $(\hat{x}, \hat{\theta})$ is a feasible solution of BiSNIP, and

$$\sum_{c \in AD} \hat{x}_c = b - \epsilon \leq b.$$

For $\delta > 0$, let $\hat{\theta}^{\tilde{\omega}} = \theta^{\tilde{\omega} LP*} - \delta$ for an arbitrary $\tilde{\omega} \in \Omega$. Define $A = \{c' \in AD^{\tilde{\omega}} : r_{c'}^{\tilde{\omega}} > \hat{\theta}^{\tilde{\omega}}\}$, and set $\hat{x}_{c'} = x_{c'}^{LP*} + \frac{\delta}{r_{c'}^{\tilde{\omega}}}$, $c' \in A$. For

$$\delta \leq \frac{\epsilon}{\sum_{c' \in A} \frac{1}{r_{c'}^{\tilde{\omega}}}},$$

$(\hat{x}, \hat{\theta})$ remains a feasible solution of the LP relaxation of BiSNIP and $\sum_{\omega \in \Omega} p^\omega \hat{\theta}^\omega < \sum_{\omega \in \Omega} p^\omega \theta^{\omega LP*}$. This contradicts that (x^{LP*}, θ^{LP*}) is an optimal solution. Therefore, $x_{c'}^{LP*} = \max\left\{\frac{r_{c'}^\omega - \theta^{\omega LP*}}{r_{c'}^\omega} : \omega \in \Omega, r_{c'}^\omega > \theta^{\omega LP*}\right\}$. We can use a similar argument to show that $x_c^{LP*} = 0$ if $r_c^\omega \leq \theta^{\omega LP*}$, $\forall \omega \in \Omega$. $\square$

Corollary 3.3.2. *Let (x^{LP*}, θ^{LP*}) be an optimal solution for the LP relaxation of BiSNIP. Then,*

1. *if $x_c^{LP*} > 0$, then there exists a scenario $\omega \in \Omega$ where $r_c^\omega > \theta^{\omega LP*}$ and $x_c^{LP*} = \frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega}$;*
2. *$x_c^{LP*} = 1$ if and only if $\theta^{\omega LP*} = 0$ for some $\omega \in \Omega$ and $c \in AD^\omega$;*
3. *for $c, c' \in AD$, if $r_c^\omega \geq r_{c'}^\omega$, $\forall \omega \in \Omega$, then $x_c^{LP*} \geq x_{c'}^{LP*}$;*
4. *if $x_c^{LP*} > x_{c'}^{LP*}$, there exists $\omega \in \Omega$ such that $r_c^\omega > r_{c'}^\omega$ and $x_c^{LP*} = \frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega}$;*

Part 1 of Corollary 3.3.2 is a direct implication of Theorem 3.3.1. Part 2 is true because $\frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega} = 1$ if and only if $\theta^\omega = 0$. In part 3, if $r_c^\omega \geq r_{c'}^\omega$, $\forall \omega \in \Omega$, then

$$\max\left\{\frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega} : \omega \in \Omega, r_c^\omega > \theta^{\omega LP*}\right\} > \max\left\{\frac{r_{c'}^\omega - \theta^{\omega LP*}}{r_{c'}^\omega} : \omega \in \Omega, r_{c'}^\omega > \theta^{\omega LP*}\right\}.$$

Finally, part 3 implies part 4. Corollary 3.3.2 provides some helpful results in our development of valid inequalities in later sections.

3.3.2 Two-step Inequality

For each pair (ω, c) where $\omega \in \Omega$ and $c \in AD^{\omega}$, constraint (3.4b) of BiSNIP is simply:

$$\theta^{\omega} \geq r_c^{\omega} - r_c^{\omega} x_c. \tag{3.11}$$

In the above inequality, θ^{ω} can be reduced from r_c^{ω} to 0 in one step in which x_c increases from 0 to 1. So, if we view (3.11) as a *one-step* inequality, we can term the following as a *two-step* inequality,

$$\theta^{\omega} \geq r_c^{\omega} - (r_c^{\omega} - r_{c'}^{\omega})x_c - r_{c'}^{\omega} x_{c'} \tag{3.12}$$

where (ω, c, c') satisfies $\omega \in \Omega$, $c, c' \in AD^{\omega}$ and $r_c^{\omega} > r_{c'}^{\omega}$. In the two-step inequality, θ^{ω} is reduced in two steps, from r_c^{ω} to $r_{c'}^{\omega}$ to 0 by first increasing x_c from 0 to 1 and then increasing $x_{c'}$ from 0 to 1.

Lemma 3.3.3. *Let $\omega \in \Omega$, $c, c' \in AD^{\omega}$ with $r_c^{\omega} > r_{c'}^{\omega}$. Then, the two-step inequality (3.12) is valid for BiSNIP.*

Proof. Let $(\hat{x}, \hat{\theta}) \in$ BiSNI. There are four cases of interest with combinations of different values of $\hat{x}_c$ and $\hat{x}_{c'}$:

1. Assume $\hat{x}_c = \hat{x}_{c'} = 0$. Then, by constraints (3.4b) $\hat{\theta}^{\omega} \geq r_c^{\omega}$, and the right-hand side of (3.12) is $r_c^{\omega} - (r_c^{\omega} - r_{c'}^{\omega})\hat{x}_c - r_{c'}^{\omega}\hat{x}_{c'} = r_c^{\omega}$. Thus, the two-step inequality is valid in case 1.

2. Assume $\hat{x}_c = 1, \hat{x}_{c'} = 0$. Then $\hat{\theta}^{\omega} \geq r_{c'}^{\omega}$, and $r_c^{\omega} - (r_c^{\omega} - r_{c'}^{\omega})\hat{x}_c - r_{c'}^{\omega}\hat{x}_{c'} = r_{c'}^{\omega}$. So that, the two-step inequality is valid.

3. Assume $\hat{x}_c = 0, \hat{x}_{c'} = 1$. Then, $\hat{\theta}^\omega \geq r_c^\omega$, and $r_c^\omega - (r_c^\omega - r_{c'}^\omega)\hat{x}_c - r_{c'}^\omega \hat{x}_{c'} = r_c^\omega - r_{c'}^\omega \leq r_c^\omega$.

4. Assume $\hat{x}_c = 1, \hat{x}_{c'} = 1$. Then, $\hat{\theta}^\omega \geq 0$, and $r_c^\omega - (r_c^\omega - r_{c'}^\omega)\hat{x}_c - r_{c'}^\omega \hat{x}_{c'} = 0$.

Therefore, the two-step inequality (3.12) is valid for BiSNIP.

□

In the next example, we show how we apply two-step inequalities to tighten the LP relaxation formulation of BiSNIP.

Example 3.3.2 (Example 3.3.1 continued). *For $b = 2$, let (x^{LP*}, θ^{LP*}) be an optimal solution of the LP relaxation of (3.9). Note $\theta^{LP*} > 0$, an optimal solution is $x_1^{LP*} = \frac{r_1 - \theta^{LP*}}{r_1}$, $x_2^{LP*} = \frac{r_2 - \theta^{LP*}}{r_2}$, and $x_3^{LP*} = \frac{r_3 - \theta^{LP*}}{r_3}$. Here is a two-step inequality,*

$$\theta \geq r_1 - (r_1 - r_2)x_1 - r_2 x_2, \tag{3.13}$$

where $r_1 - (r_1 - r_2)x_1^{LP*} - r_2 x_2^{LP*} = \theta^{LP*} + \theta^{LP*}(1 - \frac{r_2}{r_1}) > \theta^{LP*}$. In another two-step inequality,

$$\theta \geq r_1 - (r_1 - r_3)x_1 - r_3 x_3,$$

and $r_1 - (r_1 - r_3)x_1^{LP*} - r_3 x_3^{LP*} = \theta^{LP*} + \theta^{LP*}(1 - \frac{r_3}{r_1}) > \theta^{LP*}$. Both inequalities cut off (x^{LP*}, θ^{LP*}) and tighten the LP relaxation of (3.9).

This next lemma shows that there exist two-step inequalities which improve the LP relaxation of BiSNIP.

Lemma 3.3.4. *Let (x^{LP*}, θ^{LP*}) be an optimal solution for the LP relaxation of BiSNIP. If there is a scenario $\omega \in \Omega$ and $c, c' \in AD^\omega$ where $x_c^{LP*} = \frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega}$ and $x_c^{LP*} > x_{c'}^{LP*}$, then the two-step inequality,*

$$\theta^\omega \geq r_c^\omega - (r_c^\omega - r_{c'}^\omega)x_c - r_{c'}^\omega x_{c'},$$

cuts off (x^{LP*}, θ^{LP*}).

Proof. Let $\omega \in \Omega$ and $c, c' \in AD^\omega$, where $x_c^{LP*} = \frac{r_c^\omega - \theta^{\omega LP*}}{r_c^\omega}$ and $x_{c'}^{LP*} < x_c^{LP*}$. Then,

$$\begin{aligned} \theta^{\omega LP*} &< \theta^{\omega LP*} + r_{c'}^\omega(x_c^{LP*} - x_{c'}^{LP*}) \\ &= r_c^\omega(1 - x_c^{LP*}) + r_{c'}^\omega(x_c^{LP*} - x_{c'}^{LP*}) \\ &= r_c^\omega - (r_c^\omega - r_{c'}^\omega)x_c^{LP*} - r_{c'}^\omega x_{c'}^{LP*}. \end{aligned}$$

Therefore, the two-step inequality cuts off (x^{LP*}, θ^{LP*}). □

In BiSNIP, there are up to $\binom{|AD^\omega|}{2}$ possible two-step inequalities, $\omega \in \Omega$. Thus, there are at most $(\sum_{\omega \in \Omega} \binom{|AD^\omega|}{2})$ two-step inequalities for BiSNIP, in contrast to $\sum_{\omega \in \Omega} |AD^\omega|$ original constraints of form (3.4b). Adding all two-step inequalities will increase the size of BiSNIP dramatically. So, we instead iteratively solve the LP relaxation of BiSNIP and carry out the following separation algorithm to identify violated two-step inequalities for each scenario and tighten the formulation.

Algorithm 3.3.1. Separation Algorithm for Two-step Inequalities

Step 0: *Set $k = 0$.*

Step 1: *In the kth iteration, solve LP relaxation of BiSNIP. Let (θ^k, x^k) be an optimal solution.*

Step 2: *For each ω and $c, c' \in AD^\omega$ with $r_c^\omega > r_{c'}^\omega$, calculate $\beta_{c,c'}^\omega = r_c^\omega - (r_c^\omega - r_{c'}^\omega)x_c^k - r_{c'}^\omega x_{c'}^k$, and if $\beta_{c,c'}^\omega > \theta^{\omega,k}$, then add the violated two-step inequality*

$$\theta^\omega \geq r_c^\omega - (r_c^\omega - r_{c'}^\omega)x_c - r_{c'}^\omega x_{c'}$$

to BiSNIP.

Step 3: *If there are violated two-step inequalities in* **Step 2**, *let $k = k + 1$ and go to* **Step 1**. *Otherwise, stop.*

To measure the computational improvement by using two-step inequalities, we use Russia as the country in our specific application of BiSNIP. There are 85 facilities, 263 border crossings, and 9 destinations. Considering the structure of the underlying network, we aggregate the 85 sources into 34 sources. We process the problem as described in Section 2.7 and reduce the problem to (3.2). We use equation (3.3) to calculate r_c^ω, set the installation cost $a_c = 1$, $c \in AD$, and obtain BiSNIP as in (3.4). As the result of preprocessing and simplification, there are 306 scenarios, 226 binary variables, 6164 bipartite network structure constraints and the cardinality constraint from the first stage. There are 107,415 two-step inequalities in total. We start with basic LP relaxation of BiSNIP and apply Algorithm 3.3.1 to iteratively generate violated two-step inequalities and tighten the LP relaxation formulation. We

	Basic Formulation		Tightened Formulation			
b	rel. gap(%)	comp. time	rel. gap(%)	comp. time	sep. time(%)	no. of $\geq$
10	15.60	18	2.97	37	66.22	6143
20	21.84	330	4.50	61	43.25	10098
30	23.66	596	3.96	83	63.1	7179
40	23.95	491	3.79	92	62.57	5300
50	23.82	745	4.06	168	38.86	10919
60	26.03	2257	5.89	289	25.93	5196
70	29.04	7467	8.43	732	11.93	6469
80	30.50	19957	9.03	614	16.44	7289
90	31.27	4431	9.06	423	16.99	10439
100	31.15	4396	8.41	621	10.81	4884
120	28.18	508	5.04	122	71.25	8367

Table 3.1: The table shows the computational effort (in elapsed seconds) required to solve some representative instances of BiSNIP(b) and tightened BiSNIP(b) with two-step inequalities under different values of budget b.

only add two-step inequalities at the root node of the branch-and-bound tree. After no more two-step inequalities can be generated we then solve the tightened LP relaxation as an mixed-integer linear program. The computational results are shown in Table 3.1 with budget b varying from 10 to 120. Our separation procedure is implemented in C++ and the LPs and the mixed-integer linear programs are solved with CPLEX version 8.0. Table 3.1 shows the computational effort (in elapsed seconds) required to solve the instances of the basic BiSNIP formulation and the tightened BiSNIP formulation with two-step inequalities. All test instances are computed in 1.7 GHz, Dell Xeon dual-processor machine with 2 GB of memory. In Table 3.1,"rel. gap(%)" is the gap between the optimal value of BiSNIP, $z^*(b)$, and its LP relaxation as a percentage of $z^*(b)$; "comp. time" is the total computational time in seconds including solving the LP relaxations, generating and adding two-step inequalities, and solving the mixed-integer linear program; "sep. time(%)" is the percentage of total computation time spent generating two-step inequalities; "no. of $\geq$" is the number of two-step inequalities generated. The results show that by adding two-step inequalities, we can reduce the computational time by a factor of roughly 5 up to 30 and that adding inequalities has particularly helped on the most difficult problems.

3.3.3 Step Inequality

Based on our discussion of two-step inequalities in Section 3.3.2, we extend this idea to construct *step inequalities* with an arbitrary number of steps. We also discuss the facet properties of a step inequality and construct the related algorithms for solving BiSNIP by using step inequalities.

Example 3.3.3 (Examples 3.3.1 and 3.3.2 continued). *In (3.9), we can add the following three-step inequality*

$$\theta \geq r_1 - (r_1 - r_2)x_1 - (r_2 - r_3)x_2 - r_3 x_3.$$

For $x_2 > x_3$, the three-step inequality dominates the two-step inequality (3.13) of Example 3.3.2.

□

To facilitate discussion of the step inequality and its properties, we introduce an alternative formulation of BiSNIP. In (3.4), BiSNIP is expressed in the form

$$\begin{aligned}
\min_{x',\theta} \quad & \sum_{\omega\in\Omega} p^\omega \theta^\omega \\
\text{s.t.} \quad & \sum_{c\in AD} x'_c \le b' \\
& \theta^\omega \ge r_c^\omega (1 - x'_c), \qquad c \in AD^\omega, \quad \omega \in \Omega \\
& x'_c \in \{0,1\}, \qquad\qquad c \in AD.
\end{aligned}$$

Let $x_c = 1 - x'_c$, $c \in AD$ and $b = |AD| - b'$. Then, we have the following formulation BiSNIP′

$$\begin{aligned}
\min_{x,\theta} \quad & \sum_{\omega\in\Omega} p^\omega \theta^\omega \\
\text{s.t.} \quad & \sum_{c\in AD} x_c \ge b & \text{(3.14a)} \\
& \theta^\omega \ge r_c^\omega x_c, \qquad c \in AD^\omega, \quad \omega \in \Omega & \text{(3.14b)} \\
& x_c \in \{0,1\}, \qquad c \in AD. & \text{(3.14c)}
\end{aligned}$$

Let BiSNI′ denote the feasible region of BiSNIP′.

Definition 3.3.5. *For $\omega \in \Omega$, an ordered subset $T(\omega)$ is defined as*

$$T(\omega) = \{c_1, c_2, \cdots, c_{l^\omega}\} \subseteq AD^\omega$$

and satisfies the ordering condition

$$r_{c_1}^\omega > r_{c_2}^\omega > \cdots > r_{c_{l^\omega}}^\omega > 0, \tag{3.15}$$

where $r_{c_1}^\omega = \bar{r}^\omega = \max_{c\in AD^\omega} r_c^\omega$.

Definition 3.3.6. *Let* $\underline{r}^\omega$ *be a valid lower bound for* θ^ω*. For a given* $T(\omega)$*, define step size* $d^\omega_{c_i}$ *as*

$$d^\omega_{c_i} = \begin{cases} r^\omega_{c_i} - r^\omega_{c_{i+1}} & \textit{if } i = 1, \cdots, l^\omega - 1 \\ r^\omega_{c_i} - \underline{r}^\omega & \textit{if } i = l^\omega. \end{cases}$$

Note that l^ω depends on ω, but when possible we suppress this for notational simplicity. We define the step inequality on $T(\omega)$ as

$$\theta^\omega \geq \sum_{c \in T(\omega)} d^\omega_c x_c + \underline{r}^\omega. \tag{3.16}$$

Lemma 3.3.7. *Let* $T(\omega)$ *satisfy Definition 3.3.5 and* $d^\omega_{c_i}$ *satisfy Definition 3.3.6. Step inequality (3.16) is valid for BiSNI′.*

Proof. Let $(x, \theta) \in$ BiSNI′, and $S = \{c \in AD : x_c = 1\}$. If $S = \emptyset$ then

$$\theta^\omega \geq \underline{r}^\omega,$$

and (3.16) is valid. Otherwise, let $\hat{r}^\omega = \max_{c \in S} \{r^\omega_c\}$ and

$$j^* \in \operatorname*{argmin}_{1 \,\leq\, i \,\leq\, l^\omega} \{i : c_i \in S \cap T(\omega)\}.$$

Then,

$$\sum_{c_i \in T(\omega)} d^\omega_{c_i} x_{c_i} + \underline{r}^\omega = \sum_{c_i \in S \cap T(\omega)} d^\omega_{c_i} + \underline{r}^\omega \leq r^\omega_{c_{j^*}} \leq \hat{r}^\omega \leq \theta^\omega.$$

Thus, the step inequality is valid for BiSNI′. □

With the step inequality being valid for BiSNIP′, we seek conditions under which the step inequality defines a facet. For notational simplicity, we discuss facet properties of a step inequality for an arbitrary scenario $\hat{\omega}$, and hence the results can hold all scenarios.

For a given b, the wait-and-see problem for scenario $\hat{\omega}$ is

$$
\begin{aligned}
z^{\hat{\omega}}(b) = \min_{x,\theta^{\hat{\omega}}} \quad & \theta^{\hat{\omega}} \\
\text{s.t.} \quad & \sum_{c \in AD^{\hat{\omega}}} x_c \geq b \\
& \theta^{\hat{\omega}} \geq r_c^{\hat{\omega}} x_c, \qquad c \in AD^{\hat{\omega}} \\
& x_c \in \{0,1\}, \qquad c \in AD^{\hat{\omega}}.
\end{aligned}
$$

Let $L(b) = \{c \in AD^{\hat{\omega}} : r_c^{\hat{\omega}} \leq z^{\hat{\omega}}(b)\}$ and $U = AD \setminus AD^{\hat{\omega}}$. We consider three mutually exclusive and exhaustive cases in which we pose additional conditions:

Case 1: $AD^{\hat{\omega}} = AD$.

Case 2: $AD^{\hat{\omega}} \neq AD$, and $|U| \geq b$.

Case 3: $AD^{\hat{\omega}} \neq AD$, and $|U| < b$.

In scenario $\hat{\omega}$, a step inequality is

$$\theta^{\hat{\omega}} \geq \sum_{c_i \in T(\hat{\omega})} d_{c_i}^{\hat{\omega}} x_{c_i} + \underline{r}^{\hat{\omega}}. \tag{3.17}$$

Since BiSNIP′ becomes trivial if $b = 0$ or $b = |AD|$ and becomes infeasible if $b > |AD|$, we assume $0 < b < |AD|$.

Lemma 3.3.8. *Let $\hat{\omega} \in \Omega$, and suppose $AD^{\hat{\omega}} = AD$. The step inequality (3.17) with $\omega = \hat{\omega}$ is facet-defining for conv(BiSNI′) if*

1. $0 < b < |AD|$,

2. $L(b) \cap T(\hat{\omega}) \neq \emptyset$, *and*

3. $\underline{r}^{\hat{\omega}} = z^{\hat{\omega}}(b)$.

Proof. To prove that (3.17) is a facet, it suffices to show that there are $|AD| + |\Omega|$ affinely independent feasible points in BiSNI′ and that (3.17) holds with equality at these points. We construct such points in the form of $a = \binom{x}{\theta} \in \{0,1\}^{|AD|} \times R_+^{|\Omega|}$. Let e_i be the $(|AD| + |\Omega|)$-unit vector with a "1" in the position of element i and zero elsewhere (e.g., for $c \in AD$, e_c has a "1" in the position of x_c). With $\bar{r}^\omega = \max\limits_{c \in AD^\omega} r_c^\omega$, we specify the following $|AD| + |\Omega|$ points,

$$
\begin{aligned}
a^1 &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^\omega e_\omega + \sum_{c \in L(b)} e_c + z^{\hat{\omega}}(b) e_{\hat{\omega}}, && (3.18a)\\
a^\omega &= (\bar{r}^\omega + 1) e_\omega + \sum_{\omega' \in \Omega \setminus \{\omega, \hat{\omega}\}} \bar{r}^{\omega'} e_{\omega'} + z^{\hat{\omega}}(b) e_{\hat{\omega}} + \sum_{c \in L(b)} e_c, \\
&\quad \omega \in \Omega \setminus \{\hat{\omega}\}, && (3.18b)\\
a^{c_i} &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^\omega e_\omega + r_{c_l}^{\hat{\omega}} e_{\hat{\omega}} + \sum_{c \in L(b) \setminus \{c_i\}} e_c + e_{c_l}, \quad c_i \in L(b), && (3.18c)\\
a^{c_j} &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^\omega e_\omega + \sum_{j \le i \le l} e_{c_i} + r_{c_j}^{\hat{\omega}} e_{\hat{\omega}} + \sum_{c \in L(b)} e_c, \quad c_j \in T(\hat{\omega}), && (3.18d)\\
a^{c_k} &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^\omega e_\omega + e_{c_k} + \sum_{j(k) \le i \le l} e_{c_i} + r_{c_{j(k)}}^{\hat{\omega}} e_{\hat{\omega}} + \sum_{c \in L(b)} e_c, \\
&\quad k \in AD \setminus (T(\hat{\omega}) \cup L(b)), && (3.18e)
\end{aligned}
$$

where $j(k) \in \underset{c_i \,\in\, T(\hat{\omega})}{\operatorname{argmax}} \{i : r_{c_i}^{\hat{\omega}} \ge r_{c_k}^{\hat{\omega}}\}$. It is clear that there are at least $|L(b)|$ components of x that are one at each point above. Since $|L(b)| \ge b$, the cardinality constraint (3.14a) is satisfied for each point of (3.18). Since θ^{ω} is set to $\bar{r}^{\omega}$ or $\bar{r}^{\omega}+1$, $\omega \in \Omega \setminus \{\hat{\omega}\}$, constraints (3.14b) are satisfied at points of (3.18) for $\omega \in \Omega \setminus \{\hat{\omega}\}$. Points of (3.18a) and (3.18b) have $x_c = 0$ for $c \in AD \setminus L(b)$, $x_c = 1$ for $c \in L(b)$, and $\theta^{\hat{\omega}} = z^{\hat{\omega}}(b)$. Hence, these points satisfy constraint (3.14b) for scenario $\hat{\omega}$. Thus, points in (3.18a) and (3.18b) are feasible in BiSNIP′. For these points, $\theta^{\hat{\omega}} = z^{\hat{\omega}}(b)$ and so (3.17) holds with equality at (3.18a) and (3.18b). In a^{c_i} of (3.18c), $x_c = 1$ only if $c \in L(b) \cup \{c_l\}$. Since $d^{\hat{\omega}}_{c_l} = r^{\hat{\omega}}_{c_l} - z^{\hat{\omega}}(b)$ and $\theta^{\hat{\omega}} = r^{\hat{\omega}}_{c_l}$ in (3.18c), constraint (3.14b) is satisfied and (3.17) holds with equality. In a^{c_j} of (3.18d),

$$x_c = \begin{cases} 1 & c \in \{c_j, \cdots, c_l\} \\ 1 & c \in L(b) \\ 0 & \text{otherwise} \end{cases}.$$

By the definition of the step inequality,

$$\sum_{c_i \in T(\hat{\omega})} d^{\hat{\omega}}_{c_i} x_{c_i} + z^{\hat{\omega}}(b) = \sum_{j \le i \le l} d^{\hat{\omega}}_{c_i} x_{c_i} + z^{\hat{\omega}}(b) = r^{\hat{\omega}}_{c_j}.$$

Since $\theta^{\hat{\omega}} = r^{\hat{\omega}}_{c_j}$ in (3.18d), we satisfy constraint (3.14b) and have (3.17) with equality. In a^{c_k} of (3.18e),

$$x_c = \begin{cases} 1 & c \in \{c_{j(k)}, \cdots, c_l\} \\ 1 & c = c_k \\ 1 & c \in L(b) \\ 0 & \text{otherwise} \end{cases}.$$

Since $r^{\hat{\omega}}_{j(k)} \ge r^{\hat{\omega}}_{k}$ and $\theta^{\hat{\omega}} = r^{\hat{\omega}}_{j(k)}$, points of (3.18e) are feasible and (3.17) holds with equality. Therefore, points of (3.18) are feasible in BiSNI′, and the step inequality (3.17) holds with equality at each point.

Let $\hat{a}^{\hat{\omega}} = a^{\hat{\omega}} - a^1$ for $\omega \in \Omega \setminus \{\hat{\omega}\}$ and $\hat{a}^c = a^c - a^1$ for $c \in \cup AD$. Then,

$$
\begin{aligned}
\hat{a}^{\omega} &= e_{\omega}, \quad \omega \in \Omega \setminus \{\hat{\omega}\}, \\
\hat{a}^{i} &= r^{\hat{\omega}}_{c_l} e_{\hat{\omega}} + e_{c_l} - e_i, \quad i \in L(b), \\
\hat{a}^{c_j} &= r^{\hat{\omega}}_{c_j} e_{\hat{\omega}} + \sum_{j \le i \le l} e_{c_i}, \quad c_j \in T(\hat{\omega}) \\
\hat{a}^{k} &= r^{\hat{\omega}}_{c_{j(k)}} e_{\hat{\omega}} + e_k + \sum_{j(k) \le i \le l} e_{c_i}, \quad k \in AD \setminus (T(\hat{\omega}) \cup L(b)).
\end{aligned}
\tag{3.19}
$$

The following points are formed via linear combinations of points in (3.19)

$$
\begin{aligned}
\bar{a}^{\omega} &= \hat{a}^{\omega}, \quad \omega \in \Omega \setminus \{\hat{\omega}\}, \\
\bar{a}^{i} &= \hat{a}^{i} - \hat{a}^{c_l} \\
&= -e_i, \quad i \in L(b), \\
\bar{a}^{c_j} &= \hat{a}^{c_j} - \hat{a}^{c_{j+1}} \\
&= (r^{\hat{\omega}}_{c_j} - r^{\hat{\omega}}_{c_{j+1}}) e_{\hat{\omega}} + e_{c_j}, \quad c_j \in T(\hat{\omega}) \\
\bar{a}^{k} &= \hat{a}^{k} - \hat{a}^{j(k)} \\
&= e_k, \quad k \in AD \setminus (T(\hat{\omega}) \cup L(b)).
\end{aligned}
\tag{3.20}
$$

In (3.20), each point has a nonzero component which is zero in all other points, so these points are linearly independent, and hence the points of (3.18) are affinely independent. Therefore, step inequality (3.17) defines a facet for conv(BiSNI′). $\square$

The next two lemmas are for **Cases 2** and **3**. We only list the $||AD|+$

$|\Omega|$ affinely independent feasible points and omit the proofs since they are essentially the same as in the proof of Lemma 3.3.8.

Lemma 3.3.9. *Let $\hat{\omega} \in \Omega$. Suppose $AD^{\hat{\omega}} \neq AD$ and $|U| \geq b$. The step inequality (3.17) with $\omega = \hat{\omega}$ is facet-defining for conv(BiSNI′) if*

1. $0 < b < |AD|$,

2. $\underline{r}^{\hat{\omega}} = 0$.

Proof. With $U = AD \setminus AD^{\hat{\omega}}$ and $|U| \geq b$, we specify the following $|AD| + |\Omega|$ feasible and affinely independent points

$$\begin{aligned}
a^1 &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + \sum_{c \in U} e_c, \\
a^{\omega} &= (\bar{r}^{\omega} + 1) e_{\omega} + \sum_{\omega' \in \Omega \setminus \{\hat{\omega}, \omega\}} \hat{r}^{\omega'} e_{\omega'} + \sum_{c \in U} e_c, \quad \omega \in \Omega \setminus \{\hat{\omega}\}, \\
a^{c_i} &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + \sum_{c \in U \setminus \{c_i\}} e_c + r^{\hat{\omega}}_{c_l} e_{\hat{\omega}} + e_{c_l}, \quad c_i \in U, \\
a^{c_j} &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + \sum_{j \leq i \leq l} e_{c_i} + r^{\hat{\omega}}_{c_j} e_{\hat{\omega}} + \sum_{c \in U} e_c, \quad c_j \in T(\hat{\omega}), \\
a^k &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + e_k + \sum_{j(k) \leq i \leq l} e_{c_i} + r^{\hat{\omega}}_{c_{j(k)}} e_{\hat{\omega}} + \sum_{c \in U} e_c, \\
& \qquad k \in AD^{\hat{\omega}} \setminus T(\hat{\omega}),
\end{aligned} \tag{3.21}$$

where $j(k) \in \underset{c_i \in T(\hat{\omega})}{\operatorname{argmax}} \{i : r^{\hat{\omega}}_{c_i} \geq r^{\hat{\omega}}_k\}$. □

Lemma 3.3.10. *Let $\hat{\omega} \in \Omega$. Suppose $AD^{\hat{\omega}} \neq AD$ and $|U| < b$. The step inequality (3.17) is facet-defining for conv(BiSNI′) if*

1. $0 < b < |AD|$,

2. $L(b - |U|) \cap T(\hat{\omega}) \neq \emptyset$, *and*

3. $\underline{r}^{\hat{\omega}} = z^{\hat{\omega}}(b - |U|)$.

Proof. With $U = AD \setminus AD^{\hat{\omega}}$ and $|U| < b$, $d^{\hat{\omega}}_{c_l} = r^{\hat{\omega}}_{c_l} - z^{\hat{\omega}}(b - |U|)$. We specify the following $|AD| + |\Omega|$ feasible and affinely independent points

$$\begin{aligned}
a^1 &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + \sum_{c \in L(b-\delta) \cup U} e_c, \\
a^{\omega} &= (\bar{r}^{\omega} + 1) e_{\omega} + \sum_{\omega' \in \Omega \setminus \{\omega, \hat{\omega}\}} \bar{r}^{\omega'} e_{\omega'} + \sum_{c \in L(b-\delta) \cup U} e_c, \quad \omega \in \Omega \setminus \{\hat{\omega}\}, \\
a^{c_i} &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + \sum_{c \in L(b-\delta) \cup U \setminus \{c_i\}} e_c + r^{\hat{\omega}}_{c_l} e_{\hat{\omega}} + e_{c_l}, \\
&\qquad c_i \in L(b-\delta) \cup U, \\
a^{c_j} &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + \sum_{j \le i \le l} e_{c_i} + r^{\hat{\omega}}_{c_j} e_{\hat{\omega}} + \sum_{c \in L(b-\delta) \cup U} e_c, \;\; c_j \in T(\hat{\omega}), \\
a^k &= \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{r}^{\omega} e_{\omega} + e_k + \sum_{j(k) \le i \le l^{\hat{\omega}}} e_{c_i} + r^{\hat{\omega}}_{c_{j(k)}} e_{\hat{\omega}} + \sum_{c \in L(b-\delta) \cup U} e_c, \\
&\qquad k \in AD^{\hat{\omega}} \setminus (T(\hat{\omega}) \cup L(b-\delta)),
\end{aligned} \tag{3.22}$$

where $j(k) \in \underset{c_i \in T(\hat{\omega})}{\operatorname{argmax}} \{i : r^{\hat{\omega}}_{c_i} \ge r^{\hat{\omega}}_k\}$. $\square$

In the above lemmas, assumptions are made to keep (3.14) as a feasible and nontrivial optimization problem. Here, we conclude the facet property of step inequality (3.17) in the following theorem.

Theorem 3.3.11. *For* $0 < b < |AD|$*, if step inequality (3.17) is facet-defining for conv(BiSNI′) if*

1. $0 < b < |AD|$,

2. $L(b - |U|) \cap T(\hat{\omega}) \neq \emptyset$, *and*

3. $\underline{r}^{\hat{\omega}} = z^{\hat{\omega}}(b - |U|)$.

where $U = AD \setminus AD^{\hat{\omega}}$.

In BiSNIP$'$, there are at most $\sum_{\omega \in \Omega} 2^{|AD^\omega|-1}$ possible facet-defining step inequalities. Adding all possible step inequalities will increase the size of BiSNIP$'$ dramatically. So, we instead iteratively solve the LP relaxation of BiSNIP$'$ and add step inequalities on an as-needed basis. Let (x^{LP}, θ^{LP}) be a feasible solution for the LP relaxation of BiSNIP$'$. Our separation problem is to search for a step inequality which is most violated at (x^{LP}, θ^{LP}) for each scenario. For $\omega \in \Omega$, without loss of generality, assume that we already have the following order

$$r^\omega_{c_1} \geq r^\omega_{c_2} \geq \cdots \geq r^\omega_{c_{|AD^\omega|}} > 0.$$

We can identify the most violated step inequality by solving the following maximization problem

$$\begin{aligned} \max_{T(\omega) \subseteq AD^\omega} \quad & \sum_{c_i \in T(\omega)} (r^\omega_{c_i} - r^\omega_{c_{i+1}}) x^{LP}_{c_i} \\ \text{s.t.} \quad & T(\omega) \text{ satisfies Definition 3.3.5.} \end{aligned} \tag{3.23}$$

In scenario ω, we construct a network $G(V, E)$ in which $V = \{c_1, \cdots, c_{|AD^\omega|+1}\}$ and there are at most $\binom{n}{2}$ directed arcs. For constraint (3.14b) of each (ω, c_i)

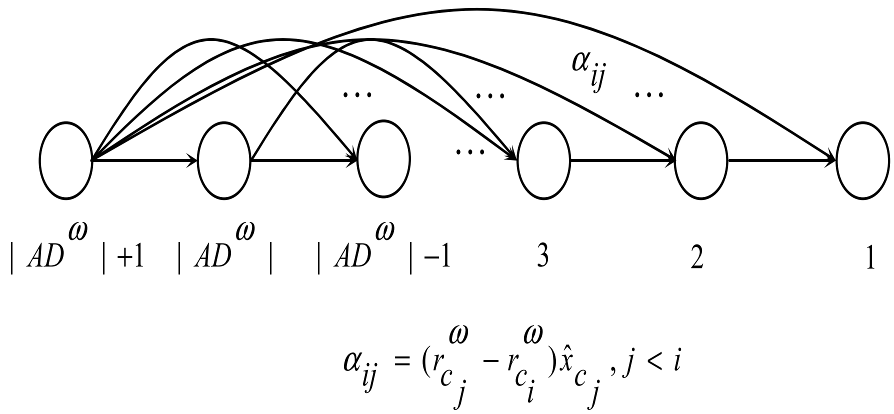

Figure 3.2: The graph shows a network which is constructed according to $r^{\omega}_{c_i}$ in constraints (3.14b) for scenario ω. Node $|AD^{\omega}|+1$ is for $\underline{r}^{\omega}$, which is zero in this graph.

pair, there is a node c_i, and node $c_{|AD^{\omega}|+1}$ is for $\underline{r}^{\omega}$. There is a directed arc from node c_i to c_j if $r^{\omega}_{c_i} \leq r^{\omega}_{c_j}$ (See Figure 3.2). Arc length α_{ij} is defined as

$$\alpha_{i,j} = \begin{cases} (r^{\omega}_{c_j} - r^{\omega}_{c_i})x^{LP}_{c_j} & \text{if } r^{\omega}_{c_i} \leq r^{\omega}_{c_j} \\ 0 & \text{if } r^{\omega}_{c_i} > r^{\omega}_{c_j}, \end{cases}$$

for c_i, $c_j \in AD^{\omega}$, and $\alpha_{i,|AD^{\omega}|+1} = (r^{\omega}_{c_i} - \underline{r}^{\omega})x^{LP}_{c_i}$, for $c_i \in AD^{\omega}$. Since $T(\omega) \in V$, an optimal solution $T(\omega)$ in (3.23) is the set nodes on a longest path from node $|AD^{\omega}|+1$ to node 1 over $G(V, E)$, where the longest path problem is formulated as follows

$$
\begin{aligned}
\max_{s \geq 0} \quad & \sum_{(i,j)\in E} \alpha_{i,j} s_{ij} \\
\text{s.t.} \quad & \sum_{(|AD^\omega|+1,j)\in FS(|AD^\omega|+1)} s_{(|AD^\omega|+1)j} = 1 \\
& \sum_{(i,j)\in FS(i)} s_{ij} - \sum_{(j,i)\in RS(i)} s_{ji} = 0 \quad i \in V \setminus \{1, |AD^\omega|+1\} \\
& \sum_{(i,1)\in RS(1)} s_{i,1} = 1.
\end{aligned}
\tag{3.24}
$$

Since $G(V, E)$ is an acyclic directed network, we can use dynamic programming techniques to solve (3.24). Every node on an optimal path represents a step in an optimal ordered set $T(\omega)$ of (3.23).

Example 3.3.4. *Assume that we have only one scenario in BiSNIP′ with* $|AD| = 5$ *and*

$$(r_1, r_2, r_3, r_4, r_5) = (0.9, 0.8, 0.5, 0.3, 0.1).$$

BiSNIP′ is

$$\begin{aligned}
\min_{x,\theta} \quad & \theta \\
\text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 + x_5 \geq 1 \\
& \theta \geq 0.9x_1 \\
& \theta \geq 0.8x_2 \\
& \theta \geq 0.5x_3 \\
& \theta \geq 0.3x_4 \\
& \theta \geq 0.1x_5 \\
& x_i \in \{0, 1\} \quad i = 1, \cdots, 5.
\end{aligned}$$

The optimal solution of the LP relaxation is

$$(x^{LP*}, \theta^{LP*}) = (0.06280, 0.07064, 0.11302, 0.18838, 0.56515, 0.05615).$$

The separation problem is to find the longest path from node 6 to node 1 in a directed acyclic network. We can visualize the directed network in Figure 3.2. Node 6 has the value of 0, which is $\underline{r}$*, and node i has value* r_i*,* $1 \leq i \leq 5$*. The length of arc* (i, j) *is* $(r_j - r_i)x_j^{LP*}$ *if* $j < i$*. For instance, there is no arc* $(1, 5)$ *since* $r_5 < r_1$*, and* $\alpha_{5,2}$ *is* $(0.8 - 0.1)0.07064 = 0.049448$ *for arc* $(5, 2)$*. A path* $6 \to 5 \to 2 \to 1$ *represents the step inequality*

$$\theta \geq r_5 x_5 + (r_2 - r_5)x_2 + (r_1 - r_2)x_1.$$

After we solve the longest path problem from node 6 to node 1, we generate the most violated step inequality

$$\theta \geq (r_1 - r_2)x_1 + (r_2 - r_3)x_2 + (r_3 - r_4)x_3 + (r_4 - r_5)x_4 + r_5 x_5, \qquad (3.25)$$

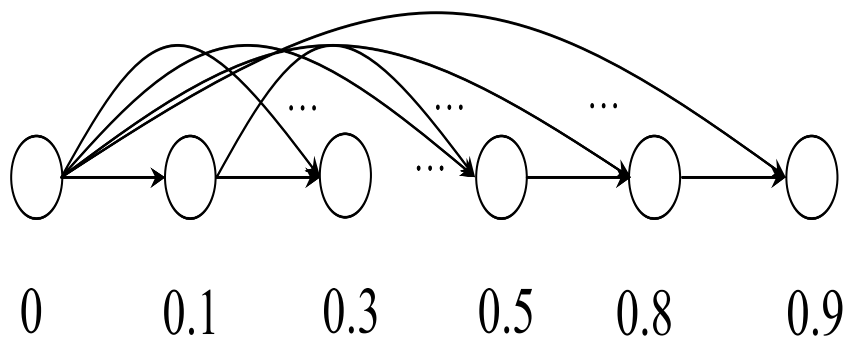

Figure 3.3: The graph shows the separation problem is equivalent to the longest path problem of an acyclic network.

which cuts off the optimal solution (x^{LP*}, θ^{LP*}).

□

We solve (3.24) for each scenario and iteratively generate step inequalities to tighten the BiSNIP′.

Algorithm 3.3.2. Algorithm for Tightening BiSNIP′

Step 1 *Solve the LP relaxation of BiSNIP′ and obtain* (x^{LP*}, θ^{LP*}).

Step 2 *For each scenario, solve (3.24) to generate the most violated step inequality, or determine that none exist.*

Step 3 *If there are step inequalities violated at* (x^{LP*}, θ^{LP*})*, then add those step inequalities to BiSNIP′ and goto* **Step 1**. *Otherwise, stop.*

As the result of Algorithm 3.3.2, we generate all necessary step inequalities to tighten the formulation of BiSNIP′.

Example 3.3.5 (Example 3.3.4 continued). *We add inequality (3.25) to the original problem and solve the LP relaxation again. The new optimal solution is* $(x^{LP*}, \theta^{LP*}) = (0.11111, 0, 0, 0, 0.88889, 0.1)$. *Apply Algorithm 3.3.2 and add a new step inequality*

$$\theta \geq (0.9 - 0.1)x_1 + 0.1x_2$$

to the formulation. Resolving the problem again, we obtain

$$(x^2, \theta) = (0, 0, 0, 0, 1, 0.1)$$

which is also a feasible solution to the original mixed-integer linear program, and there are no additional violated step inequalities. In this example, as the result of Algorithm 3.3.2, we obtain an integer optimal solution.

$\square$

3.4 Computational Results

We use the same computational case as in Section 3.3.2. We apply Algorithm 3.3.2 to add step inequalities prior to applying the branch-and-bound algorithm, and then there are no additional inequalities generated during the branch-and-bound algorithm. Our separation procedure is implemented in C++ and the LPs and the mixed-integer linear programs are solved via

	Basic Formulation		Tightened Formulation			
b	rel. gap. (%)	comp. time	rel. gap. (%)	comp. time	no. of $\geq$	iters.
10	15.6	19	0.00	12	732	8
20	21.8	319	0.17	15	859	9
30	23.7	660	0	8	619	6
40	23.9	539	0.06	14	651	7
50	23.8	697	0	12	570	8
60	26.0	2133	0.03	41	1401	19
70	29.0	6310	0.71	63	1353	18
80	30.5	19629	0.24	63	1156	16
90	31.3	6977	0.06	36	1034	14
100	31.2	2628	0.26	34	808	8
110	30.6	682	0.55	34	739	9
120	28.2	280	0.41	23	558	7

Table 3.2: The table shows the computational effort (in elapsed seconds) required to solve some representative instances of BiSNIP(b) and tightened BiSNIP(b) with step inequalities under different values of the budget b.

CPLEX version 9.0. All mixed-integer programs are solved with relative tolerance level 0.1%. Table 3.2 displays the computational results for: (i) solving our basic MIP formulation directly using CPLEX's branch-and-bound code and (ii) adding violated step inequalities to the initial linear programming relaxation and then proceeding with CPLEX's branch-and-bound code. We added step inequalities until the maximum violation was less than 10^{-6}. In the table, "rel. gap (%)" is $100 \cdot (z_{IP} - z_{LP})/z_{IP}$, "comp. time" reports total computation time in seconds, "no. of $\geq$" reports number of step inequalities generated and "iters." reports the total number of major iterations. The results suggest that step inequalities can significantly reduce required computational effort, particularly on the most challenging instances.

Chapter 4

Solution Methods For SNIP

4.1 Introduction

Chapter 3 discussed properties of BiSNIP, the special case in which our stochastic network interdiction model is on a bipartite network, and developed associated solution methods. This chapter discusses the properties of SNIP on a general network and develops associated solution methods.

Section 4.2 simplifies and preprocesses the basic SNIP formulation. In Section 4.3, we discuss the L-shaped method for SNIP and two techniques to enhance the performance of the L-shaped method. Section 4.4 extends our development of step inequalities from bipartite networks (Section 3.3) to SNIP on a general network. We provide computational experience in Section 4.5 and prove a number of our results in Section 4.6.

4.2 Tightening the Formulation

In Section 2.3, we showed that SNIP, on a general network can be formulated as

$$\begin{aligned}
\min_{x,\pi} \quad & \sum_{\omega \in \Omega} p^\omega \pi^\omega_{s^\omega} && && \\
\text{s.t.} \quad & \sum_{(i,j) \in AD} a_{ij} x_{ij} \le b && && \\
& \pi^\omega_i - p_{ij}\pi^\omega_j \ge 0, && (i,j) \in A \setminus AD^\omega,\ \omega \in \Omega && \\
& \pi^\omega_i - p_{ij}\pi^\omega_j + x_{ij} \ge 0, && (i,j) \in AD^\omega,\ \omega \in \Omega && \text{(4.1a)} \\
& \pi^\omega_i - q_{ij}\pi^\omega_j + (1 - x_{ij}) \ge 0, && (i,j) \in AD^\omega, \omega \in \Omega && \text{(4.1b)} \\
& \pi_{t^\omega} \ge 1, && \omega \in \Omega, && \\
& x_{ij} \in \{0,1\} && (i,j) \in AD &&
\end{aligned}$$

where $AD^\omega \subseteq AD$ is the set of interdictable arcs for scenario ω. We can tighten constraints (4.1a) and (4.1b) by reducing the coefficients of x.

Theorem 4.2.1. *Assume that $0 \le \pi^\omega_i \le 1$, $i \in N, \omega \in \Omega$. Then, the constraints*

$$\pi^\omega_i - p_{ij}\pi^\omega_j + (p_{ij} - q_{ij})\pi^\omega_j x_{ij} \ge 0, \quad (i,j) \in AD^\omega,\ \omega \in \Omega \quad \text{(4.2a)}$$

$$\pi^\omega_i - q_{ij}\pi^\omega_j \ge 0, \quad (i,j) \in AD^\omega,\ \omega \in \Omega \quad \text{(4.2b)}$$

are valid for (4.1) and are stronger than (4.1a) and (4.1b), respectively.

Proof. For $\omega \in \Omega$ and $(i,j) \in AD^\omega$, if $x_{ij} = 0$, constraint (4.2a) is the same as (4.1a), and constraint (4.2b) is vacuous since $p_{ij} > q_{ij}$ and $\pi^\omega_j \ge 0$. If $x_{ij} = 1$,

$$\pi^\omega_i - p_{ij}\pi^\omega_j + (p_{ij} - q_{ij})\pi^\omega_j x_{ij} = \pi^\omega_i - q_{ij}\pi^\omega_j,$$

and (4.2a), (4.2b), and (4.1b) become the same inequality. Thus, (4.2a) and (4.2b) are valid inequalities for (4.1).

With $0 \le \pi_j^\omega \le 1$ and $0 \le q_{ij} \le p_{ij} \le 1$, we have $(p_{ij} - q_{ij})\pi_j^\omega \le 1$. So,

$$\pi_i^\omega - p_{ij}\pi_j^\omega + x_{ij} \ge \pi_i^\omega - p_{ij}\pi_j^\omega + (p_{ij} - q_{ij})\pi_j^\omega x_{ij},$$

and (4.2a) is stronger than (4.1a). Since $1 - x_{ij} \ge 0$, we have

$$\pi_i^\omega - q_{ij}\pi_j^\omega + (1 - x_{ij}) \ge \pi_i^\omega - q_{ij}\pi_j^\omega,$$

and (4.2b) is stronger than (4.1b). □

The assumption $\pi_i^\omega \ge 0$ is satisfied for all feasible solutions, and assumption $\pi_i^\omega \le 1$ is satisfied at any optimal solution. Inequality (4.2a) is nonlinear because of the product $\pi_j^\omega x_{ij}$. We can replace π_j^ω with an upper bound and (4.2a) will remain valid. Setting $x_{ij} = 0$, $(i,j) \in AD$, we can compute $\hat{\pi}_j^\omega$, which is the optimal value of a maximum-reliability path from node j to node t^ω in the uninterdicted network.

Corollary 4.2.2. *Constraints*

$$\begin{aligned}
&\pi_i^\omega - p_{ij}\pi_j^\omega + (p_{ij} - q_{ij})\hat{\pi}_j^\omega x_{ij} \ge 0, \quad && (i,j) \in AD^\omega,\ \omega \in \Omega \\
&\pi_i^\omega - q_{ij}\pi_j^\omega \ge 0, && (i,j) \in AD^\omega,\ \omega \in \Omega
\end{aligned}$$

are valid for (4.1) and stronger than (4.1a) and (4.1b), respectively.

In what follows, we simply substitute 1 for $\hat{\pi}_j^\omega$, $j \in N$ and $\omega \in \Omega$.

Corollary 4.2.3. *Constraints*

$$\pi_i^\omega - p_{ij}\pi_j^\omega + (p_{ij} - q_{ij})x_{ij} \geq 0 \quad (i,j) \in AD^\omega, \omega \in \Omega$$

$$\pi_i^\omega - q_{ij}\pi_j^\omega \geq 0 \quad (i,j) \in AD^\omega, \omega \in \Omega$$

are valid for (4.1) and stronger than (4.1a) and (4.1b), respectively.

Hereafter, by SNIP, we mean

$$
\begin{aligned}
z^* = \min_{x,\pi} & \sum_{\omega \in \Omega} p^\omega \pi_{s^\omega}^\omega \\
\text{s.t.} & \sum_{(i,j) \in AD} a_{ij} x_{ij} \leq b && && \text{(4.3a)} \\
& \pi_i^\omega - p_{ij}\pi_j^\omega \geq 0, && (i,j) \in A \setminus AD^\omega, \omega \in \Omega && \text{(4.3b)} \\
& \pi_i^\omega - p_{ij}\pi_j^\omega + (p_{ij} - q_{ij})x_{ij} \geq 0, && (i,j) \in AD^\omega, \omega \in \Omega && \text{(4.3c)} \\
& \pi_i^\omega - q_{ij}\pi_j^\omega \geq 0, && (i,j) \in AD^\omega, \omega \in \Omega && \text{(4.3d)} \\
& \pi_{t^\omega} \geq 1, && \omega \in \Omega, && \text{(4.3e)} \\
& x_{ij} \in \{0,1\} && (i,j) \in AD, &&
\end{aligned}
$$

and we will develop solution techniques for (4.3).

4.3 L-Shaped Method

4.3.1 Multi-cut L-shaped method

We first form the master program and the subproblem for the L-shaped method. Separating the second-stage problem for each scenario from the deterministic equivalent formulation (4.3), we have

$$\begin{aligned} h(x,(s^\omega,t^\omega)) = \min_{\pi}\ & \pi_{s^\omega} \\ & \pi_i - p_{ij}\pi_j \geq 0 && (i,j) \in A \setminus AD^\omega \\ & \pi_i - p_{ij}\pi_j + (p_{ij} - q_{ij})x_{ij} \geq 0 && (i,j) \in AD^\omega \\ & \pi_i - q_{ij}\pi_j \geq 0 && (i,j) \in AD^\omega \\ & \pi_{t^\omega} \geq 1. \end{aligned} \tag{4.4}$$

The dual of (4.4) is

$$\begin{aligned} & h(x,(s^\omega,t^\omega)) = \\ & \max_{y\geq 0, z\geq 0} \quad y_{t^\omega} - \sum_{(i,j)\in AD^\omega} (p_{ij} - q_{ij}) y_{ij} x_{ij} \\ & \text{s.t.} \quad \sum_{(s^\omega,j)\in FS(s^\omega)} (y_{s^\omega j} + z_{s^\omega j}) = 1 \\ & \qquad \sum_{(i,j)\in FS(i)} (y_{ij} + z_{ij}) - \\ & \qquad \sum_{(j,i)\in RS(i)} (p_{ji} y_{ji} + q_{ji} z_{ji}) = 0, \quad i \in N \setminus \{s^\omega, t^\omega\} \\ & \qquad y_{t^\omega} - \sum_{(j,t^\omega)\in RS(t^\omega)} (p_{jt^\omega} y_{jt^\omega} + q_{jt^\omega} z_{jt^\omega}) = 0. \end{aligned} \tag{4.5}$$

We can write the first-stage problem as

$$\begin{aligned} \min_{x,\theta} \quad & \sum_{\omega\in\Omega} p^\omega \theta^\omega \\ \text{s.t.} \quad & \sum_{(i,j)\in AD} a_{ij} x_{ij} \leq b \\ & \theta^\omega \geq h(x,(s^\omega,t^\omega)), \qquad \omega \in \Omega \\ & x_{ij} \in \{0,1\}, \qquad (i,j) \in AD. \end{aligned} \tag{4.6}$$

With the expression of $h(x, (s^\omega, t^\omega))$ from (4.5), we can write the first-stage problem as

$$
\begin{aligned}
\underline{z} = \min_{x,\theta} \quad & \sum_{\omega \in \Omega} p^\omega \theta^\omega \\
\text{s.t.} \quad & \sum_{(i,j) \in AD} a_{ij} x_{ij} \le b && \text{(4.7a)} \\
& \theta^\omega \ge y_{t^\omega}^k - \left[\sum_{(i,j) \in AD_k^\omega} (p_{ij} - q_{ij}) y_{ij}^{\omega,k} x_{ij} \right], \\
& \qquad \omega \in \Omega, k \in K^\omega && \text{(4.7b)} \\
& x_{ij} \in \{0,1\} \quad (i,j) \in AD,
\end{aligned}
$$

where set K^ω is the index set of extreme points of the feasible region in (4.5). If K^ω indexes all the extreme points of (4.5)'s feasible region then (4.7) is called the full master and $\underline{z} = z^*$. If K^ω is only a subset of these extreme points then (4.7) is called the relaxed master and $\underline{z} \le z^*$. We call (4.7) and (4.5) the master program and subproblem, respectively. Each extreme point of (4.5)'s feasible region is a simple path connecting s^ω and t^ω. In an optimality cut in (4.7b), $y_{t^\omega}^k$ is the evasion probability of the kth path from s^ω to t^ω, and $y_{ij}^{\omega,k} > 0$ if arc (i,j) is on the kth evasion path and is not interdicted. Here, $AD_k^\omega = \{(i,j) \in AD : y_{ij}^{\omega,k} > 0\}$ denotes the set of uninterdicted interdictable arcs on the kth path in scenario ω. The size of full master program can be very large because the number of possible paths for a given network is typically enormous. In the L-shaped method, we iteratively add optimality cuts with the goal of obtaining an optimal solution of the original problem by solving a relaxed master program, which contains a subset of optimality cuts. Next, we state the multi-cut L-shaped algorithm for SNIP.

Algorithm 4.3.1. Multi-cut L-shaped Algorithm for SNIP

Step 0: *Define a tolerance* $\epsilon > 0$. *Set* $k = 1$, $\bar{z} = 1$, *and* $\underline{z} = 0$. *Initialize constraints (4.7b) with* $\theta^\omega \geq 0$.

Step 1: *Solve* k*th relaxed master program (4.7). Obtain optimal objective value* $\underline{z}$ *and optimal solution* (x^k, θ^k).

Step 2: *For each scenario* $\omega \in \Omega$, *solve subproblem (4.5) with* x^k, *and obtain the optimal objective value* $h(x^k, (s^\omega, t^\omega))$ *and an optimal solution* $y^{\omega,k}$. *Let* $\hat{z} = \sum_{\omega \in \Omega} p^\omega h(x^k, (s^\omega, t^\omega))$. *If* $\bar{z} \geq \hat{z}$, *then update the incumbent solution* $\bar{z} = \hat{z}$ *and* $x^* = x^k$.

Step 3: *If* $\bar{z} - \underline{z} \leq \min(|\bar{z}|, |\underline{z}|)\epsilon$, *then stop:* x^* *is a solution with objective function value within* $(100\epsilon)\%$ *of optimum.*

Step 4: *Append the set of optimality cuts*

$$\theta^\omega \geq y^k_{t^\omega} - \sum_{(i,j) \in AD^\omega_k} (p_{ij} - q_{ij}) y^{\omega,k}_{ij} x_{ij}, \quad \omega \in \Omega$$

to master program (4.7). Set $k = k + 1$ *and goto* **Step 1**.

Each iteration includes solving a master program and a subproblem for each $\omega \in \Omega$. The solution of the relaxed master program is a feasible interdiction plan, and the solution of the subproblem is an evasion path, which is the evader's response to the interdiction plan.

Theorem 4.3.1. *Algorithm 4.3.1 terminates in Step 3 at a solution x^* with objective values $\hat{z}$ within $(100\epsilon)\%$ of z^* in a finite number of steps.*

Proof. The proof follows the standard argument (e.g., [58] by Van Slyke and Wets) noting that SNIP has a finite optimal solution and relatively complete recourse. □

4.3.2 Enhanced L-shaped Method

In this section, we introduce two techniques to improve the performance of Algorithm 4.3.1. During each iteration, we generate additional valid cuts for each scenario to improve the approximation of $h(x, (s^\omega, t^\omega))$, and we fix some components of x to solve the master problem (4.7) heuristically.

At each iteration, we generate $|\Omega|$ optimality cuts, but only one cut for each scenario. These cuts form an outer approximation of the recourse function. To generate an optimality cut, we need to solve a master program and a subproblem. The subproblem is a maximum-reliability path problem and can be solved polynomially. But, the master program is a mixed-integer linear program and the computational time can be large. So, it may be better to generate more than one optimality cut for each scenario at each iteration.

At the kth iteration, we solve the subproblem and obtain the optimality cut,

$$\theta^\omega \geq y^k_{t^\omega} - \sum_{(i,j)\in AD^\omega_k} (p_{ij} - q_{ij}) y^{\omega,k}_{ij} x_{ij}, \tag{4.8}$$

which is an evasion path in scenario ω. If this evasion path is interdicted, what is the next optimal evasion path? In Algorithm 4.3.1, we won't answer this question unless we solve another master program. Thus, it may be helpful to generate some additional evasion paths assuming that the current evasion path is interdicted. With AD_k^ω being the set of interdictable arcs on the kth path in scenario ω, we set $x_{ij} = 1$ one at a time for each $(i,j) \in AD_k^\omega$ and resolve the subproblem for scenario ω to generate additional evasion paths. We can then add up to $|AD_k^\omega|$ additional optimality cuts for each scenario. In this way, we aim to reduce the number of iterations in the L-shaped method.

For a given b, the space of feasible interdiction plans can be very large. For example, assuming $a_{ij} = 1, \forall (i,j) \in AD$, there are $\binom{|AD|}{b}$ different interdiction plans. When b is large, it may take many iterations for the L-shaped method to converge. Here, we describe a heuristic method which decreases the computational effort by solving the master program at some iterations by fixing a subset of the binary variables to 1. At the kth iteration, we solve the master program and obtain an optimal solution (x^k, θ^k). Solving the subproblem, we generate an optimality cut with the evasion probability $y_{t^\omega}^k$ for each $\omega \in \Omega$. In the next algorithm, we describe a technique to fix a subset of components of x to 1 for iteration $k+1$.

Algorithm 4.3.2. Algorithm for Fixing Variables

Step 0: *Input: the master program solution* (x^k, θ^k), *the subproblem solution* $y_{t^\omega}^k$, $\omega \in \Omega$. *Set* $\delta > 0$.

Step 1: *Fix variable* x_{ij} *at 1 if* $x_{ij}^k = 1$.

Step 2: *For* $(i,j) \in AD$ *with* $x_{ij} = 1$, *if there exists* $\omega \in \Omega$ *and* $l \in \{1, \cdots, k\}$ *such that* $(i,j) \in AD_l^\omega$ *and* $y_{t^\omega}^l \leq \delta y_{t^\omega}^k$, *free variable* x_{ij}.

In Algorithm 4.3.2, we separate optimality cuts into one group with large evasion probabilities and one group with small evasion probabilities for each scenario. As the number of iterations increases, we are more likely to fix x_{ij} to 1 if (i,j) is on a path with large evasion probabilities and to search for the rest of interdiction plan in the group with small evasion probabilities. By fixing some components of x, we can keep x^{k+1} from moving too far away from x^k and this may help to eliminate the well-known "bang–bang" effect, which can slow convergence of the L-shaped method.

Note $\underline{z}$ is not a valid lower bound on z^* if some x's are fixed in this manner. Thus, we can only terminate in **Step 3** in Algorithm 4.3.1 when we free all components of x and solve the unrestricted master program.

Algorithm 4.3.3. Enhanced Multi-cut L-shaped Algorithm for SNIP

Step 0: *Define a tolerance* $\epsilon > 0$. *Set* $k = 1$, $\bar{z} = 1$, *and* $\underline{z} = 0$. *Initialize constraints (4.7b) with* $\theta^\omega \geq 0$.

Step 1: *Solve* k*th relaxed master program (4.7). Obtain the optimal objective value* $\underline{z}$ *and an optimal solution* (x^k, θ^k).

Step 2: *For each scenario* $\omega \in \Omega$, *solve subproblem (4.5) with* x^k, *and obtain the optimal objective value* $\hat{z}(x^k, (s^\omega, t^\omega))$ *and an optimal solution*

$y^{\omega,k}$. Let $\hat{z} = \sum_{\omega \in \Omega} p^\omega h(x^k, (s^\omega, t^\omega))$. If $\bar{z} \geq \hat{z}$, then update the incumbent solution $\bar{z} = \hat{z}$ and $x^* = x^k$.

Step 2.1: *Apply Algorithm 4.3.2 to fix components of x.*

Step 3: *If $\bar{z} - \underline{z} \leq \min(|\bar{z}|, |\underline{z}|)\epsilon$ and the unrestricted master program was solved, then stop: x^* is a solution with objective function value within $(100\epsilon)\%$ of optimum. If $\bar{z} - \underline{z} \leq \min(|\bar{z}|, |\underline{z}|)\epsilon$ and the master program was solved with some variables fixed, free all components of x and go to* **Step 4**.

Step 4: *Append the set of optimality cuts*

$$\theta^\omega \geq y^k_{t^\omega} - \sum_{(i,j) \in AD^\omega_k} (p_{ij} - q_{ij}) y^{\omega,k}_{ij} x_{ij}$$

to master problem (4.7) for each scenario $\omega \in \Omega$.

Step 5: *For each $(i,j) \in AD^\omega_k$, set $\hat{x}^k = x^k$ and $\hat{x}^k_{ij} = 1$, and solve subproblem $h(\hat{x}^k, (s^\omega, t^\omega))$ and add the optimality cut to (4.7). Repeat the process for each scenario. Set $k = k + 1$ and goto* **Step 1**.

4.4 Step Inequality

The master problem (4.7) is a mixed-integer linear program, and the bulk of the computational effort in solving SNIP via the L-shaped method is in solving (4.7). In Section 3.3, we introduced step inequalities to tighten the LP relaxation formulation of BiSNIP, and we illustrated the benefit of these facet-defining inequalities in reducing computational time. In this section, we extend the notion of step inequalities to SNIP.

In the master problem, there is a budget constraint (4.7a) and a set of optimality cuts (4.7b). Each optimality cut

$$\theta^\omega \geq y^k_{t^\omega} - \sum_{(i,j)\in AD^\omega_k} (p_{ij} - q_{ij}) y^{\omega,k}_{ij} x_{ij}$$

contains an evasion path, where $y^k_{t^\omega}$ is the overall evasion probability and if $y^{\omega,k}_{ij} > 0$ then arc (i,j) is on this optimal path. Let

$$T^\omega = \{l_1, l_2, \cdots, l_L\} \subseteq K^\omega$$

be an ordered index set with

$$y^{l_1}_{t^\omega} > y^{l_2}_{t^\omega} > \cdots > y^{l_L}_{t^\omega},$$

and $y^{l_1}_{t^\omega} = \max_{i\in K^\omega} y^i_{t^\omega}$. The intuition behind the step inequality for SNIP is that a smuggler chooses a path with the largest evasion probability unless the path is interdicted, then the smuggler chooses the second best path and so on. We have two types of step inequalities in different algebraic forms. The TYPE-I step inequality is defined as

$$\theta^\omega \geq y^{l_1}_{t^\omega} - \sum_{l_i\in T^\omega} (y^{l_i}_{t^\omega} - y^{l_{i+1}}_{t^\omega})(\sum_{(i,j)\in AD^\omega_{l_i}} x_{ij}), \tag{4.9}$$

where $y^{l_{L+1}}_{t^\omega} = 0$.

Theorem 4.4.1. *The* TYPE-I *step inequality (4.9) is valid for master program (4.7).*

Proof. Let $(\hat{x}, \hat{\theta})$ be a feasible solution of (4.7). For $\omega \in \Omega$, let $W^\omega = \{k : y^k_{t^\omega} > \hat{\theta}^\omega\}$. For $k \in W^\omega$, $\exists (i,j) \in AD^\omega_k$ such that $\hat{x}_{ij} = 1$. Then,

$$\begin{aligned} y^{l_1}_{t^\omega} - \sum_{l_i \in T^\omega} (y^{l_i}_{t^\omega} - y^{l_{i+1}}_{t^\omega})(\sum_{(i,j)\in AD^\omega_{l_i}} \hat{x}_{ij}) &= y^{l_1}_{t^\omega} - \sum_{l_i \in W^\omega \cap T^\omega} (y^{l_i}_{t^\omega} - y^{l_{i+1}}_{t^\omega})(\sum_{(i,j)\in AD^\omega_{l_i}} \hat{x}_{ij}) \\ &\quad - \sum_{l_i \in T^\omega \setminus W^\omega} (y^{l_i}_{t^\omega} - y^{l_{i+1}}_{t^\omega})(\sum_{(i,j)\in AD^\omega_{l_i}} \hat{x}_{ij}) \\ &\leq y^{l_1}_{t^\omega} - \sum_{l_i \in W^\omega \cap T^\omega} (y^{l_i}_{t^\omega} - y^{l_{i+1}}_{t^\omega}) \qquad \text{(4.10a)} \\ &\quad - \sum_{l_i \in T^\omega \setminus W^\omega} (y^{l_i}_{t^\omega} - y^{l_{i+1}}_{t^\omega})(\sum_{(i,j)\in AD^\omega_{l_i}} \hat{x}_{ij}) \\ &\leq \hat{\theta}^\omega, \end{aligned}$$

where the last inequality holds since $\hat{\theta}^\omega \geq y^k_{t^\omega}$ for $k \in T^\omega \setminus W^\omega$ and the telescoping terms in (4.10a) collapse to such a $y^k_{t^\omega}$ (or to zero if $W^\omega = T^\omega$). Thus, the TYPE-I step inequality is valid for (4.7). □

In the last theorem, we showed that the TYPE-I step inequality is valid (4.7) , which includes the full master program if K^ω denotes all extreme points of (4.5), $\omega \in \Omega$. Next, we provide an example for the TYPE-I step inequality.

Example 4.4.1. *Consider the following problem*

$$
\begin{aligned}
\min_{x,\theta} \quad & \theta \\
\text{s.t.} \quad & \sum_{1 \le i \le 8} x_i \le 4 \\
& \theta \ge 0.9 - 0.9x_1 - 0.9x_2 \\
& \theta \ge 0.8 - 0.9x_2 - 0.8x_3 \\
& \theta \ge 0.7 - 0.8x_1 - 0.7x_4 \\
& \theta \ge 0.6 - 0.6x_5 - 0.7x_6 \\
& \theta \ge 0.5 - 0.5x_7 - 0.6x_8 \\
& \theta \ge 0 \\
& x_i \in \{0, 1\} \quad i = 1, \cdots, 8.
\end{aligned}
$$

We assume that $\theta \ge 0$ since the evasion probability of any path is nonnegative. Let $K = \{k_1, k_2, k_3, k_4, k_5\}$, index of the five respective optimality cuts, $T = \{l_1 = k_1, l_2 = k_3, l_3 = k_5\}$, and $AD = \{1, \cdots, 8\}$. Then, Type-I *step inequality defined on T is*

$$\theta \ge 0.9 - (0.9 - 0.7)(x_1 + x_2) - (0.7 - 0.5)(x_1 + x_4) - (0.5 - 0)(x_7 + x_8). \quad (4.11)$$

We construct a set of nine feasible points, $a^1, \cdots, a^9$, of the form (x, θ) as follows

	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9
x_1	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*1*	*1*
x_2	*0*	*1*	*1*	*0*	*1*	*0*	*0*	*1*	*1*
x_3	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*0*	*0*
x_4	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*0*
x_5	*0*	*1*	*0*	*0*	*1*	*1*	*0*	*1*	*1*
x_6	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*0*
x_7	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*0*
x_8	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*1*
θ	*0.9*	*0.5*	*0.7*	*0.9*	*0.5*	*0.9*	*0.9*	*0*	*0*

These nine points are feasible and affinely independent. The TYPE-I *step inequality (4.11) holds with equality at each of the nine points. The dimension of the feasible region of the above optimization problem is 9. Therefore, (4.11) defines a facet for the convex hull of the feasible region. Because there are two* x *components in each "step", the step size between steps can be doubled. If* $x_1 = 1$, $x_2 = 1$ *and all other* x *components are zero, in (4.11) we have* $\theta \geq 0.3$ *instead of* $\theta \geq 0.5$.

□

The general proof of the facet property for the TYPE-I step inequality consists of wearisome notation and an enumeration of affinely independent points. We defer the proof to the appendix at the end of this chapter.

We define the TYPE-II step inequality for SNIP on T^ω as

$$\theta^\omega \geq y_{t^\omega}^{l_1} - (y_{t^\omega}^{l_1} - y_{t^\omega}^{l_2})v_{l_1}^\omega - (y_{t^\omega}^{l_2} - y_{t^\omega}^{l_3})v_{l_2}^\omega - \cdots - (y_{t^\omega}^{l_L} - 0)v_{l_L}^\omega, \tag{4.12}$$

where $v^{\omega}_{l_k}$ is an auxiliary variable defined by the constraints

$$v^{\omega}_{l_k} \leq \sum_{(i,j)\in AD^{\omega}_{l_k}} x_{ij} \tag{4.13a}$$

$$0 \leq v^{\omega}_{l_k} \leq 1. \tag{4.13b}$$

If $x_{ij} = 0$, $\forall (i,j) \in AD^{\omega}_{l_k}$ then constraints (4.13) force $v^{\omega}_{l_k} = 0$. If $x_{ij} = 1$ for some $(i,j) \in AD^{\omega}_{l_k}$ then $v^{\omega}_{l_k}$ may take a positive value and θ^{ω} may become less than $y^{l_k}_{t^{\omega}}$. In the kth iteration, there are at most k new variables for each scenario and $|\Omega| \times k$ new variables in total. We don't add variable $v^{\omega}_{l_k}$ and the associated constraint (4.13) to (4.7) unless we add a Type-II step inequality where $v^{\omega}_{l_k}$ is in.

Theorem 4.4.2. *The* Type-II *step inequality (4.12)-(4.13) is valid for master program (4.7).*

Proof. Let $(\hat{x}, \hat{\theta})$ be a feasible solution of (4.7). For $\omega \in \Omega$, let $W^{\omega} = \{k : y^{k}_{t^{\omega}} > \hat{\theta}^{\omega}\}$. For $k \in W^{\omega}$, $\exists (i,j) \in AD^{\omega}_{k}$ such that $\hat{x}_{ij} = 1$. So, $v^{\omega}_{k} = 1$ satisfies (4.13) for $k \in W^{\omega}$. Then,

$$\begin{aligned}
y^{l_1}_{t^{\omega}} - \sum_{l_i \in T^{\omega}} (y^{l_i}_{t^{\omega}} - y^{l_{i+1}}_{t^{\omega}}) v^{\omega}_{l_i} &= y^{l_1}_{t^{\omega}} - \sum_{l_i \in W^{\omega} \cap T^{\omega}} (y^{l_i}_{t^{\omega}} - y^{l_{i+1}}_{t^{\omega}}) v^{\omega}_{l_i} \\
&\quad - \sum_{l_i \in T^{\omega} \setminus W^{\omega}} (y^{l_i}_{t^{\omega}} - y^{l_{i+1}}_{t^{\omega}}) v^{\omega}_{l_i} \\
&\leq y^{l_1}_{t^{\omega}} - \sum_{l_i \in W^{\omega} \cap T^{\omega}} (y^{l_i}_{t^{\omega}} - y^{l_{i+1}}_{t^{\omega}}) \qquad (4.14a) \\
&\quad - \sum_{l_i \in T^{\omega} \setminus W^{\omega}} (y^{l_i}_{t^{\omega}} - y^{l_{i+1}}_{t^{\omega}}) v^{\omega}_{l_i} \\
&\leq \hat{\theta}^{\omega},
\end{aligned}$$

where the last inequality holds since $\hat{\theta}^{\omega} \geq y^{k}_{t^{\omega}}$ for $k \in T^{\omega} \setminus W^{\omega}$ and the telescoping terms in (4.14a) collapse to such a $y^{k}_{t^{\omega}}$ (or to zero if $W^{\omega} = T^{\omega}$). Thus, the Type-II step inequality is valid for SNIP. □

Before we proceed to further discussions on step inequalities. We use the following example to illustrate the difference between the two types of step inequalities.

Example 4.4.2 (Example 4.4.1 continued). *The* Type-II *step inequality defined on T is*

$$\theta \geq 0.9 - (0.9 - 0.7)v_{l_1} - (0.7 - 0.5)v_{l_2} - (0.5 - 0)v_{l_3}, \tag{4.15}$$

with additional constraints

$$\begin{array}{ll} v_{l_1} \leq x_1 + x_2, & 0 \leq v_{l_1} \leq 1; \\ v_{l_2} \leq x_1 + x_4, & 0 \leq v_{l_2} \leq 1; \\ v_{l_3} \leq x_7 + x_8, & 0 \leq v_{l_3} \leq 1. \end{array}$$

With $x_1 = 1$ and $x_2 = 1$, we have $v_1 = 1$ and $v_2 = 1$. Then in the (4.15), $\theta \geq 0.5$ while $\theta \geq 0.3$ with the Type-I *step inequality in Example 4.4.1. In this sense, the* Type-II *step inequality is tighter than the* Type-I *step inequality.*

□

The formal proof of the facet property for the Type-II step inequality is also in the appendix at the end of this chapter. In our computation, we will only implement the Type-II step inequality, and we use the term step inequality to refer to the Type-II step inequality hereafter. We apply step inequalities in the course of the L-shaped method to tighten the master program

(4.7) and reduce the computing time of solving the MIP master program. We solve the LP relaxation of (4.7) and obtain an optimal solution (x^{LP}, θ^{LP}). Let $\hat{v}_{l_k}^{\omega} = \min\{\sum_{(i,j)\in AD_{l_k}^{\omega}} x_{ij}^{LP}, 1\}$. Then, we solve a separation problem to find T^{ω} and the associated TYPE-II step inequality which is most violated at the current solution. Assume that we are at the Kth iteration and without loss of generality, we have the following order

$$y_{t^{\omega}}^{l_1} \geq y_{t^{\omega}}^{l_2} \geq \cdots \geq y_{t^{\omega}}^{l_K}.$$

The separation problem for $\omega \in \Omega$ can formulated as

$$\min_{T^{\omega}\subseteq\{l_1,\cdots,l_K\}} \sum_{l_k\in T^{\omega}} (y_{t^{\omega}}^{l_k} - y_{t^{\omega}}^{l_{k+1}})\hat{v}_{l_k}^{\omega}, \tag{4.16}$$

and this problem is similar to the separation problem of the step inequality for BiSNIP in Section 3.3.3. We construct a network $G(V, E)$ in which $V = \{l_1, \cdots, l_{K+1}\}$ and there are at most $\binom{K+1}{2}$ directed arcs. In scenario ω, for each optimality cut (4.7b) of (ω, l_i) pair, there is a node l_i in $G(V, E)$, and node l_{K+1} is for 0. There is a directed arc from node l_i to l_j if $y_{t^{\omega}}^{l_i} \leq y_{t^{\omega}}^{l_j}$. Arc length β_{ij} is defined as

$$\beta_{i,j} = \begin{cases} (y_{t^{\omega}}^{l_j} - y_{t^{\omega}}^{l_i})\hat{v}_{c_j}^{\omega} & \text{if } y_{t^{\omega}}^{l_i} \leq y_{t^{\omega}}^{l_j} \\ 0 & \text{if } y_{t^{\omega}}^{l_i} > y_{t^{\omega}}^{l_j}, \end{cases}$$

for l_i, $l_j \in \{l_1, \cdots, l_K\}$, and $\beta_{i,K+1} = y_{t^{\omega}}^{l_i}\hat{v}_{l_i}^{\omega}$, for $l_i \in \{l_1, \cdots, l_K\}$. We can identify an optimal solution of (4.16) by solving the shortest path problem from node l_{K+1} to node l_1 over $G(V, E)$. The nodes on a shortest path from node l_{K+1} to node l_1 represent the steps in T^{ω} on which the most violated step inequality may be defined. Since all arc lengths are nonnegative, this shortest path problem can be solved polynomially.

We solve the separation problem before we apply the branch-and-bound algorithm to the master program. We iteratively solve the LP relaxation of (4.7) and generate step inequalities to tighten the LP relaxation of (4.7). Then, we solve the resulting tightened master problem. We add **Step 1'** before **Step 1** in Algorithm 4.3.1 and Algorithm 4.3.3.

Step 1': Solve the LP relaxation of the kth master problem (4.7). Solve the shortest path problem and obtain an optimal solution of (4.16) for each scenario. Repeat until there are no violated step inequalities. Then, go to **Step 1**.

In the next section we investigate the benefits of enhancing Algorithm 4.3.1 with step inequalities as well as our additional valid inequality generation and heuristic method of the master mixed-integer linear program.

4.5 Computational Results

We test our solution methods for SNIP on a general network using a test problem of 85 facilities, 93 provinces, 320 border checkpoints, and 12 destinations. A transportation network connecting these entities includes air, land, and water transportation modes. There are 830 nodes (85+93+320$\times$ 2 +12) and 2645 arcs in our overall network. After preprocessing, there are 456 scenarios. We set $a_{ij} = 1$, $\forall (i,j) \in AD$ so that the budget constraint becomes a cardinality constraint.

We consider four variants of this problem. In the first denoted SNIP1, p_{ij}s are independent and UNIFORM(0.5,1) (uniformly distributed between 0.5 and 1), for $(i,j) \in A \setminus AD$; p_{ij}s are independent and UNIFORM(0.3,0.6) and q_{ij}s are independent and UNIFORM(0.1,0.3) for $(i,j) \in AD$. In the second variant of the test problem, denoted SNIP2, p_{ij} takes on the same value as in SNIP1 and then we set $q_{ij} = 0.5p_{ij}$. SNIP3 and SNIP4 are identical to SNIP2 except that we set $q_{ij} = 0.1p_{ij}$ and $q_{ij} = 0$, respectively.

We test four approaches: (1) Directly solve the deterministic equivalent formulation (DEF), i.e., model (4.1), using branch-and-bound; (2) Algorithm 4.3.1, i.e., the basic L-shaped method (LS); (3) Algorithm 4.3.1 with **Step 1'**, i.e., the L-shaped method with the step inequality (LSSI); (4) Algorithm 4.3.3 with **Step 1'**, i.e., the L-shaped method enhanced with both step inequalities and additional cut-generation and heuristic techniques (LSSI+). All four algorithms require solving mixed-integer linear programs. To do so, we use CPLEX version 8.0 in its default mode with a relative stopping criterion of 1%. When running the L-shaped method we also terminated with a relative error of 1%, i.e., $\epsilon = 0.01$ in Algorithms 4.3.1 and 4.3.3. The computations are on a 2.8 GHz Dell Xeon dual-processor machine with 1GB memory. We generate five independent instances of SNIP1 and then use these as the basis for five instances each of SNIP2–SNIP4. For our four algorithms we compare the average and maximal computational times in the following tables.

	DEF	LS	LSSI	LSSI+
b	t. time	t. time(40–95%)	t. time(9–46%)	t. time(2–12%)
30	(29.0, 49.6)	(12.4, 25.4)	(3.9, 6.1)	(1.5, 1.9)
40	(55.6, 97.5)	(28.3, 40.3)	(3.3, 3.9)	(1.9, 2.5)
50	(60.6, 161.6)	(25.5, 36.5)	(3.2, 4.5)	(2.1, 2.8)
60	(67.8, 88.3)	(62.6, 121.9)	(3.7, 4.7)	(2.0, 2.8)
70	(35.4, 54.2)	(24.3, 53.2)	(3.4, 4.0)	(2.2, 3.1)
80	(9.7, 15.3)	(3.9, 4.4)	(2.7, 3.3)	(2.0, 2.3)
90	(5.1, 10.5)	(3.2, 5.5)	(2.4, 3.4)	(1.9, 2.3)

Table 4.1: The table shows the average and maximal computational effort (in elapsed minutes) required to solve five instances of SNIP1 by DEF, LS, LSSI, and LSSI+ under different values of the budget b.

Table 4.1 contains the average and maximal computational effort over five instances for SNIP1. In $(\cdot, \cdot)$, the first number is the average computation time and the second number is the maximal computation time. The "t. time" is the overall time for solving SNIP1 by each of the four methods, and "xx-xx%" in the parentheses is the rough range for the percentage of the average overall time spent solving the MIP master program in methods LS, LSSI and LSSI+. The results indicate that the L-shaped method alone reduces the computation time, but our step inequalities further reduce the computational effort, especially for solving the master program. The additional cut-generation and heuristic methods for fixing variables also reduce the average overall computation time and MIP time, which is no more than 16 seconds for any of the budget values.

Table 4.2 is analogous to Table 4.1, except that it reports our results

	DEF	LS	LSSI	LSSI+
b	t. time	t. time(10–86%)	t. time(4–33%)	t. time(1–23%)
30	(29.3, 69.3)	(9.0, 25.1)	(2.3, 3.9)	(1.4, 1.9)
40	(54.3, 98.4)	(14.4, 26.8)	(2.7, 4.7)	(1.7, 2.2)
50	(36.4, 54.0)	(11.2, 21.9)	(2.2, 3.2)	(1.9, 3.1)
60	(36.9, 61.0)	(8.9, 15.1)	(3.0, 4.4)	(1.7, 2.3)
70	(11.7, 22.1)	(5.2, 9.1)	(3.0, 4.4)	(2.2, 3.2)
80	(3.4, 6.2)	(2.8, 4.5)	(2.7, 4.2)	(1.7, 2.2)
90	(0.8, 1.3)	(1.8, 3.3)	(2.6, 4.4)	(1.7, 1.8)

Table 4.2: The table shows the average and maximal computational effort (in elapsed minutes) required to solve some representative instances of SNIP2 by LS, LSSI, and LSSI+ under different values of the budget b.

for SNIP2. LS performs better in SNIP2 than in SNIP1. LS, LSSI and LSSI+ reduce the computational time in all cases except for $b = 90$ in which DEF solves quickly. The improvement due to the special purpose L-shaped method with the step inequalities and further enhancements is significant. For example, average computation time to solve the MIP master program is less than one minute for LSSI and LSSI+ for all values of the budget.

Table 4.3 contains the computational results for SNIP3, where $q = 0.1p$. Here, the solution times for DEF and LS are over the 10 hours in most of the cases. LS doesn't reduce the computational time to 10 hours or less except for $b = 30$. SNIP3 is more difficult to solve than SNIP1 and SNIP2, and the value of the special purpose L-shaped method with the step inequalities and further enhancements is even more pronounced. In LSSI, average times for all of the budget values are within two hours except for $b = 90$. Additional

	DEF	LS	LSSI	LSSI+
b	t. time	t. time(96–100%)	t. time(56–92%)	t. time(28–74%)
30	(301.7, 437.7)	(75.5, 102.0)	(9.6, 17.1)	(5.0,7.7)
40	(505.6, $\times$)	($\times$, $\times$)	(42.8, 82.9)	(7.9, 10.7)
50	(579.7, $\times$)	($\times$, $\times$)	(54.2, 131.4)	(10.1, 13.2)
60	($\times$, $\times$)	($\times$, $\times$)	(107.5, 287.8)	(11.9, 15.9)
70	($\times$, $\times$)	($\times$, $\times$)	(108.1, 252.3)	(10.8, 21.7)
80	($\times$, $\times$)	($\times$, $\times$)	(85.6, 128.3)	(15.9, 20.1)
90	($\times$, $\times$)	($\times$, $\times$)	(189.9, 376.9)	(26.2, 71.6)

Table 4.3: The table shows the average and maximal computational effort (in elapsed minutes) required to solve five instances of SNIP3 by LS, LSSI, and LSSI+ under different values of the budget b. $\times$ means computational time is more than 10 hours.

cut-generation and heuristic methods of fixing variables reduce the average computational time to under sixteen minutes except for $b = 90$.

In Table 4.4, we test our methods for $q = 0$. The computation times of DEF and LS are similar to SNIP3. The step inequality alone reduces the computation time but the performance is worse than in SNIP1, SNIP2 and SNIP3. With the enhanced L-shaped method and step inequalities, the average time of LSSI+ is within 1 hour for all of budget values.

4.6 Appendix

In this appendix, we show the facet properties for two types of step inequalities with $a_{ij} = 1$ in (4.7). For $a_{ij} = 1$, the budget constraint (4.7a) is

	DEF	LS	LSSI	LSSI+
b	t. time	t. time(97–100%)	t. time(64–95%)	t. time(40–85%)
30	(245.0, 352.8)	(118.6, 167.0)	(14.4, 19.3)	(6.7, 11.4)
40	(543.0, $\times$)	(531.1, $\times$)	(46.8, 71.8)	(14.1, 23.2)
50	($\times$, $\times$)	($\times$, $\times$)	(124.2, 247.8)	(57.0, 131.7)
60	($\times$, $\times$)	($\times$, $\times$)	(185.9, 372.7)	(33.7, 55.5)
70	($\times$, $\times$)	($\times$, $\times$)	(269.8, $\times$)	(56.1, 141.4)
80	($\times$, $\times$)	($\times$, $\times$)	(206.5, 369.3)	(18.0, 45.1)
90	($\times$, $\times$)	($\times$, $\times$)	(349.1, $\times$)	(49.0, 103.8)

Table 4.4: The table shows the average and maximal computational effort (in elapsed minutes) required to solve some representative instances of SNIP4 by LS, LSSI, and LSSI+ under different values of the budget b. $\times$ means computational effort is more than 10 hours.

simply a cardinality constraint and the master program (4.7) becomes

$$\begin{aligned}
\underline{z} = \min_{x,\theta} \quad & \sum_{\omega \in \Omega} p^{\omega} \theta^{\omega} \\
\text{s.t.} \quad & \sum_{(i,j) \in AD} x_{ij} \leq b && \text{(4.17a)} \\
& \theta^{\omega} \geq y^{k}_{t^{\omega}} - [\sum_{(i,j) \in AD^{\omega}_{k}} (p_{ij} - q_{ij}) y^{\omega,k}_{ij} x_{ij}], \\
& \quad \omega \in \Omega, k \in K^{\omega} && \text{(4.17b)} \\
& x_{ij} \in \{0,1\} \quad (i,j) \in AD,
\end{aligned}$$

Let *conv*(SNI) be the convex hull of the feasible region of (4.17). Set $T^{\omega} = \{l_1, \cdots, l_L\} \subseteq K^{\omega}$ is an ordered index set satisfying

$$y^{l_1}_{t^{\omega}} > y^{l_2}_{t^{\omega}} > \cdots > y^{l_L}_{t^{\omega}},$$

and $y^{l_1}_{t^{\omega}} = \max_{k \in K^{\omega}} y^{k}_{t^{\omega}}$. Before we proceed to prove the facet property of the step inequality defined on T^{ω}, we introduce the following notation and function.

Definition 4.6.1.

1. *Set* $AD^{\omega}_{T^{\omega}}$ *is defined as* $\bigcup_{i \in T^{\omega}} \{AD^{\omega}_{i}\}$.

2. *Function* $s : AD^{\omega}_{T^{\omega}} \to \{1, \cdots, L\}$ *is defined as*

$$s(c) = \underset{1 \leq i \leq L}{\operatorname{argmin}} \{y^{l_i}_{t^{\omega}} : l_i \in T^{\omega}, \;\; c \in AD^{\omega}_{l_i}\}.$$

Map $r : T^{\omega} \to AD^{\omega}$ *assigns an element* $c \in AD^{\omega}_{l_i}$ *to* l_i *if* $s(c) = l_i$, $l_i \in T^{\omega}$.

3. *For* $l_i \in T^{\omega}$, *set* S^i *is defined as a subset of* $\bigcup \{AD^{\omega}_{k} : k \in K^{\omega}, y^{k}_{t^{\omega}} > y^{l_{i+1}}_{t^{\omega}}\}$ *and* S^i *satisfies*

   (a) $|S^i| \leq b - 1$,

   (b) $S^i \cap (\bigcup_{i \leq j \leq L} AD^{\omega}_{l_j}) = \emptyset$,

   (c) $y^{k}_{t^{\omega}} - \sum_{c \in S^i \cap AD^{\omega}_{k}} (p_c - q_c) y^{\omega,k}_{c} \leq y^{l_{i+1}}_{t^{\omega}}$, $\forall k \in K^{\omega} \setminus \{l_i\}$ *such that* $y^{k}_{t^{\omega}} \geq y^{l_{i+1}}_{t^{\omega}}$.

Example 4.6.1 (Example 4.4.1 continued). *Here is the problem from Ex-*

ample 4.4.1,

$$\begin{aligned}
\min_{x,\theta} \quad & \theta \\
\text{s.t.} \quad & \sum_{1\le i\le 8} x_i \le 4 \\
& \theta \ge 0.9 - 0.9x_1 - 0.9x_2 \\
& \theta \ge 0.8 - 0.9x_2 - 0.8x_3 \\
& \theta \ge 0.7 - 0.8x_1 - 0.7x_4 \\
& \theta \ge 0.6 - 0.6x_5 - 0.7x_6 \\
& \theta \ge 0.5 - 0.5x_7 - 0.6x_8 \\
& \theta \ge 0 \\
& x_i \in \{0,1\} \quad i = 1, \cdots, 8.
\end{aligned}$$

We assume that $\theta \ge 0$ since the evasion probability of any path is nonnegative. Let $K = \{k_1, k_2, k_3, k_4, k_5\}$, index of the five respective optimality cuts, $T = \{l_1 = k_1, l_2 = k_3, l_3 = k_5\}$, and $AD = \{1, \cdots, 8\}$. Then,

1. *Set $AD_T = \{1, 2, 4, 7, 8\}$ contains components of x which have positive coefficients in those optimality cuts from T.*

2. *For function s, $s(1) = 2$ because x_1 is in inequality k_3, which is indexed l_2 in T. For rest of elements in AD_T, $s(2) = 1$, $s(4) = 2$, $s(7) = 3$, and $s(8) = 3$. For function r, $r(l_1) = 2$, $r(l_2) = 1$, and $r(l_3) = 7$.*

3. *For map r, $r(l_1) = 2$, $r_(l_2) = 1$, and $r(l_3) = 7$. There can be other ways to assign values to l_i, $l_i \in T^\omega$.*

4. For l_1, $S^1 = \{3\}$; for l_2, $S^2 = \{2,5\}$; for l_3, $S^3 = \{1,3,5\}$.

□

For $T^\omega \subseteq K^\omega$, TYPE-I step inequality is defined in Section 4.4 as

$$\theta^\omega \geq y_{t^\omega}^{l_1} - \sum_{l_i \in T^\omega} (y_{t^\omega}^{l_i} - y_{t^\omega}^{l_{i+1}})(\sum_{(i,j) \in AD_{l_i}^\omega} x_{ij}). \tag{4.18}$$

where $y_{t^\omega}^{l_{L+1}} = 0$. Let $\bar{y}_{t^\omega} = \max_{k \in K^\omega} y_{t^\omega}^k$ for $\omega \in \Omega$. We construct $|\Omega| + |AD|$ points in the form of (x, θ) for an arbitrary scenario $\hat{\omega}$,

$$a^1 = \sum_{\omega \in \Omega} \bar{y}_{t^\omega} e_\omega \tag{4.19a}$$

$$a^\omega = (\bar{y}_{t^\omega} + 1)e_\omega + \sum_{\omega' \in \Omega \setminus \{\omega\}} \bar{y}_{t^\omega} e_{\omega'}, \quad \omega \in \Omega \setminus \{\hat{\omega}\} \tag{4.19b}$$

$$a^i = \sum_{\omega \in \Omega} \bar{y}_{t^\omega} e_\omega + e_i, \quad \forall i \in AD \setminus AD_{T^{\hat{\omega}}}^{\hat{\omega}} \tag{4.19c}$$

$$a^c = \sum_{\omega \in \Omega \setminus \{\hat{\omega}\}} \bar{y}_{t^\omega} e_\omega + y_{t^{\hat{\omega}}}^{l_{s(c)+1}} e_{\hat{\omega}} + e_c + \sum_{j \in S^{s(c)}} e_j, \quad c \in AD_{T^{\hat{\omega}}}^{\hat{\omega}}, \tag{4.19d}$$

where $S^{s(c)}$ is defined according to part 3. in Definition 4.6.1. Let $z_b^{\hat{\omega}}$ be the optimal objective value of the wait-and-see problem of (4.17) for scenario $\hat{\omega}$ with budget b.

Lemma 4.6.2. *For a given T^ω, all points in (4.19) are feasible for* conv*(SNI) if there exists S^i, $\forall l_i \in T^\omega$.*

Proof. We prove the feasibility by showing all points satisfy the cardinality constraint (4.17a) and optimality cuts (4.17b).

For $\omega \in \Omega \setminus \{\hat{\omega}\}$, θ^{ω} takes value of $\bar{y}_{t^{\omega}}$ or $\bar{y}_{t^{\omega}} + 1$ in points from (4.19). Therefore, optimality cuts (4.17b) of all scenarios except scenario $\hat{\omega}$ are satisfied at these points. Next, we show that optimality cuts in scenario $\hat{\omega}$ are satisfied at these points. At points of a^1, a^{ω} and a^i, $\theta^{\hat{\omega}} = \bar{y}_{t^{\hat{\omega}}} \geq y^k_{t^{\hat{\omega}}}$, $\forall k \in K^{\hat{\omega}}$, so optimality cuts are satisfied. In a^c, since $x_c = 1$, $y^{l_{s(c)}}_{t^{\omega}} - y^{\hat{\omega}, l_{s(c)}}_{c} \leq y^{l_{s(c)+1}}_{t^{\omega}}$. By the definition of $S^{s(c)}$, $y^k_{t^{\omega}} - \sum_{(i,j) \in S^{s(c)} \cap AD^{\omega}_k} y^{\hat{\omega},k}_{ij} \leq y^{l_{s(c)+1}}_{t^{\omega}}$, $\forall k \in K^{\omega} \setminus \{l_{s(c)}\}$ with $y^k_{t^{\omega}} > y^{l_{s(c)+1}}_{t^{\omega}}$. Hence, optimality cuts (4.17b) are satisfied at points of a^c in scenario $\hat{\omega}$.

Cardinality constraint (4.17a) is also satisfied in these points because:

1. in a^1, $\sum_{j \in AD} x_j = 0$;
2. in a^{ω}, $\sum_{j \in AD} x_j = 0$, $\omega \in \Omega \setminus \{\hat{\omega}\}$;
3. in a^i, $\sum_{j \in AD} x_j = 1$, $i \in AD \setminus AD^{\hat{\omega}}_{T^{\hat{\omega}}}$;
4. in a^c, $\sum_{j \in AD} x_j = |S^{s(c)}| + 1 \leq b$, $c \in AD^{\hat{\omega}}_{T^{\hat{\omega}}}$.

Therefore, points of (4.19) are feasible for *conv*(SNI). □

Lemma 4.6.3. *For a given T^{ω},* TYPE-I *inequality holds with equality at points of (4.19) if for $l_i \in T^{\omega}$, if there exists S^i with $|S^i \cap AD^{\omega}_{l_k}| = 1$, $1 \leq k < i$.*

Proof. For points of a^1, a^{ω} and a^i, $x_{ij} = 0$, $\forall (i,j) \in AD^{\hat{\omega}}_{T^{\hat{\omega}}}$, and $\theta^{\omega} = \bar{y}_{t^{\hat{\omega}}}$. So, the inequality is at equality.

Let $\hat{c} \in AD^{\hat{\omega}}_{T^{\hat{\omega}}}$. By the definition of $S^{s(\hat{c})}$,

$$y^k_{t^{\hat{\omega}}} - \sum_{(i,j)\in S^{s(c)}\cap AD^{\hat{\omega}}_k} y^{\hat{\omega},k}_{ij} \leq y^{l_{s(c)+1}}_{t^{\hat{\omega}}},$$

$\forall k \in K^{\hat{\omega}} \setminus \{l_{s(c)}\}$ with $y^k_{t^{\hat{\omega}}} > y^{l_{s(c)+1}}_{t^{\hat{\omega}}}$. In $a^{\hat{c}}$, $x_{\hat{c}} = 1$ and $|S^{s(\hat{c})} \cap AD^{\hat{\omega}}_{l_i}| = 1$, $1 \leq i < s(c)$. The right-hand side of the step inequality becomes

$$\begin{aligned}
& y^{l_1}_{t^{\hat{\omega}}} - \sum_{l_i \in T^{\hat{\omega}}} (y^{l_i}_{t^{\hat{\omega}}} - y^{l_{i+1}}_{t^{\hat{\omega}}})(\sum_{(i,j)\in AD^{\hat{\omega}}_{l_i}} x_{ij}) \\
= \; & y^{l_1}_{t^{\hat{\omega}}} - \sum_{1\leq i<s(\hat{c})} (y^{l_i}_{t^{\hat{\omega}}} - y^{l_{i+1}}_{t^{\hat{\omega}}})(\sum_{(i,j)\in S^{s(\hat{c})}\cap AD^{\hat{\omega}}_i} x_{ij}) - (y^{l_{s(\hat{c})}}_{\hat{\omega}} - y^{l_{s(\hat{c})+1}}_{\hat{\omega}})x_{\hat{c}} \\
= \; & y^{l_{s(\hat{c})+1}}_{\hat{\omega}}.
\end{aligned}$$

TYPE-I inequality holds with equality at points of a^c. Therefore, TYPE-I inequality holds with equality at each point of (4.19). $\square$

Lemma 4.6.4. *Points of (4.19) are affinely independent.*

Proof. Let $b^j = a^j - a^1$ for $j \in (\Omega \setminus \{\hat{\omega}\}) \cup AD$, then

$$b^\omega = a^\omega - a^1 = e_\omega \quad \forall \omega \in \Omega \setminus \{\hat{\omega}\} \tag{4.20a}$$

$$b^i = a^i - a^1 = e_i \quad \forall i \in AD \setminus AD^{\hat{\omega}}_{T^{\hat{\omega}}} \tag{4.20b}$$

$$\begin{aligned} b^c = a^c - a^1 = \; & (y^{l_{s(c)+1}}_{t^{\hat{\omega}}} - \bar{y}_{t^{\hat{\omega}}})e_{\hat{\omega}} + e_c + \\ & \sum_{j\in S^{s(c)}} e_j \quad \forall c \in AD^{\hat{\omega}}_{T^{\hat{\omega}}}. \end{aligned} \tag{4.20c}$$

We use the following steps to further simplify b^c,

Step 0 Let $j = 1$. For each $c \in AD^{\hat{\omega}}_T$ such that $s(c) = 1$, let $\bar{b}^c = b^c - \sum_{k\in S^j} b^k$.

Step 1 Let $j = j + 1$. If $j > L$, stop.

Step 2 For each $c \in AD^{\hat{\omega}}_{T^{\hat{\omega}}}$ such that $s(c) = j$, let $\bar{b}^c = b^c - \sum_{k \in S^j \setminus AD^{\hat{\omega}}_{T^{\hat{\omega}}}} b^k - \sum_{c' \in S^j \cap AD^{\hat{\omega}}_{T^{\hat{\omega}}}} \bar{b}^{c'}$. Goto **Step 1**.

As the result of the above algorithm, we have

$$\bar{b}^c = d^{s(c)} e^{\hat{\omega}} + e_c,$$

where $d^{s(c)}$ is a finite real number. Points of $\bar{b}^c$ are obtained as linear combinations of points from (4.20). Points of $\bar{b}^c$ together with points of b^ω and b^i are linearly independent. Hence, points in (4.19) are affinely independent. □

Theorem 4.6.5. *For a given T^ω,* Type-I *step inequality (4.18) is facet-defining for* conv(SNI) if for $l_i \in T^\omega$, there exists S^i with $|S^i \cap AD^\omega_{l_k}| = 1$, $1 \le k < i$.

Proof. By Lemma 4.6.2, Lemma 4.6.3 and Lemma 4.6.4, step inequality (4.18) is facet-defining for *conv(SNI)*. □

The Type-II step inequality is

$$\theta^\omega \ge y^{l_1}_{t^\omega} - \sum_{l_i \in T^\omega} (y^{l_i}_{t^\omega} - y^{l_{i+1}}_{t^\omega}) v^\omega_{l_i}, \tag{4.21}$$

with auxiliary constraints

$$\begin{aligned} v^\omega_k &\le \sum_{c \in AD^\omega_k} x_c \quad && \forall k \in T^\omega \\ v^\omega_k &\le 1 && \forall k \in T^\omega \\ v^\omega_k &\ge 0 && \forall k \in T^\omega. \end{aligned}$$

Let SNI' be the feasible region of (4.17) with auxiliary variables and constraints.

Theorem 4.6.6. *For a given T^{ω},* Type-II *step inequality (4.21) is facet-defining for* conv(SNI') if for $l_i \in T^{\omega}$, there exists S^i.

Proof. The proof is similar to the proof of Theorem 4.6.5. $\square$

Chapter 5

Conclusion

This dissertation has developed a class of stochastic network interdiction models. In these models the interdictor installs detectors on arcs in a network subject to a budget constraint. Then the smuggler's random origin-destination pair is revealed and the smuggler selects a maximum-reliability path in the residual network. The interdictor's goal is to minimize the reliability of the smuggler's maximum-reliability path. The work in this dissertation was motivated by the Second Line of Defense (SLD) Program, which is a cooperative program between the US DOE and the Russian Federation State Customs Committee. The SLD Program aims to minimize the risk of illicit trafficking of nuclear material, equipment and technology.

Chapter 2 began with our basic model in which an informed evader has full knowledge of the interdicted network and of the interdiction probabilities. Then, we developed models for an uninformed smuggler and for the case where the population of potential smugglers consists of a mixture of both informed and uninformed types. Next, we developed a more general model which includes these special cases and more general situations in which the interdictor and evader have differing perceptions of the network parameters, i.e., the evasion probabilities on each arc. Finally, motivated by a special case in which detectors can be placed only on the border of a single country or region, we reduced our basic model to a stochastic network interdiction model on a bipartite network. We showed that the basic model and more specifically, even its special case on a bipartite network, has an NP-Complete decision problem. We view this result as justifying our pursuit of our basic network interdiction models as stochastic mixed-integer programs.

In Chapter 3 we focused on solution methods for the bipartite stochastic network interdiction model. We simplified the model via preprocessing and developed a class of valid inequalities designed to tighten the linear-programming relaxation of the mixed-integer program. We termed our valid inequalities as step inequalities and provided mild conditions under which they are facets. Given a solution to the linear-programming relaxation we provided an efficient separation algorithm that either identifies a violated step inequality or shows that all step inequalities are satisfied. Computational results on a test problem derived from installing detectors at customs checkpoints in Russia demonstrated the value of step inequalities, decreasing solution times from hours to minutes in some cases.

Chapter 4 develops solution methods for the basic stochastic network interdiction model defined on a general network, e.g., when we consider multiple regions or countries and detectors can be installed on both the "boundary" and "interior" of that network. We tightened the model through coefficient reduction and other preprocessing. We then adapted the L-shaped decomposition method to solve the model. We extended the notion of the step inequality from bipartite networks to the model on a general network, deriving two types of step inequalities that exploit special structures in L-shaped optimality cuts. We formed a separation algorithm for the general network case, analogous to that for the bipartite network. We showed that both types of step inequalities are facets. Finally, we developed computational enhancements that generated additional optimality cuts and heuristically solved the master program. Computational tests demonstrated the value of our special-purpose solution methods.

The main contributions of this dissertation lie in both introducing a new class of stochastic network interdiction models which have important real-world applications and in developing solution methods which exploit special structures in our basic model. Our interdiction models vary with respect to the information known by the evader and with respect to the interdictor's and evader's perceptions of key network parameters. An important part of our special-purpose solution methods was the introduction of so-called step inequalities, which are valid inequalities designed to tighten the linear programming relaxation of the basic models. The step inequality is facet-defining when we assume that the budget constraint is simply a cardinality constraint.

Many questions remain for future research topics. How can we incorporate changes in information perceived by the evader depending on the interdiction plan? How can we incorporate smuggling rings in a model? What other types of notions of risk should be incorporated? How sensitive is the proposed interdiction plan to inaccuracies in the network parameters? On the algorithmic side, can the step inequality be generalized to a wider range of problems? What type of approximations are appropriate when the problem size becomes large due to growth in the network and growth in the number of scenarios? Can effective heuristics be developed? Can effective solution techniques be developed for the perception-based models? To what extent can interdiction models play a larger role in models for homeland security?

Bibliography

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, Upper Saddle River, NJ, 1993.

[2] N. Assimakopoulos. A network interdiction model for hospital infection control. *Computers in Biology and Medicine*, 17:413–522, 1987.

[3] A. Atamtürk, G.L Nemhauser, and M.W.P Savelsbergh. The mixed vertex packing problem. *Mathematical Programming*, 89:35–53, 2000.

[4] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324, 1993.

[5] E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42:1229–1246, 1996.

[6] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8:146–164, 1975.

[7] E. Balas. The prize collecting traveling salesman problem. *Networks*, 19:621–636, 1989.

[8] M.O. Ball, B.L. Golden, and R.V. Vohra. Finding the most vital arcs in a network. *Operations Research Letters*, 8:73–76, 1989.

[9] J.F. Bard and J.T. Moore. A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, 11:281–292, 1990.

[10] J.F. Bard. *Practical Bilevel Optimization: Algorithms and Applications.* Kluwer Academic Publishers, Dordrecht, 1998.

[11] C. Barnhart, C.A. Hane, and P.H. Vance. Integer multicommodity flow problems. In P.M. Pardalos, D.W. Heam, and W.W. Hager, editors, *Lecture Notes in Economics and Mathematical Systems*, 450, pages 17–31. Springer, Berlin, 1996.

[12] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

[13] A. Bar-Noy, S. Khuller, and B. Schieber. The complexity of finding most vital arcs and nodes. Technical Report CS-TR-35-39, Computer Science Department, University of Maryland, 1995.

[14] P. Bauer. The circuit polytope: facets. *Mathematics of Operations Research*, 22:110–145, 1997.

[15] E.M.L. Beale. On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society*, 17B:173–184, 1955.

[16] O. Ben-Ayed. Bi-level linear programming. *Computers and Operations Research*, 20:485–501, 1993.

[17] J.R. Birge and F.V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34:384–392, 1988.

[18] J.R. Birge and F.V. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.

[19] M.S. Chern and K.C. Lin. Interdicting the activities of a linear program–a parametric analysis. *European Journal of Operational Research*, 86:580–591, 1995.

[20] H.W. Corley and D.Y. Sha. Most vital links and nodes in weighted networks. *Operations Research Letters*, 1:157–160, 1982.

[21] K.J. Cormican, D.P. Morton, and R.K. Wood. Stochastic network interdiction. *Operations Research*, 46:184–197, 1998.

[22] H. Crowder, E.L. Johnson, and M.W. Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31:803–834, 1983.

[23] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson. Solution of a large-scale traveling salesman problem. *Operations Research*, 2:393–410, 1954.

[24] G.B. Dantzig. Linear programming under uncertainty. *Management Science*, 1:197–206, 1955.

[25] J. Desrochers, Y. Dumas, M.M. Solomon, and F. Soumis. Time constrained routing and scheduling. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Handbook in Operations Research and Management Science*, 8, pages 35–140. Elsevier, 1995.

[26] D.R. Fulkerson and G.C. Harding. Maximizing the minimum source-sink path subject to a budget constraint. *Mathematical Programming*, 13:116–118, 1977.

[27] P.M. Ghare, D.C. Montgomery, and T.M. Turner. Optimal interdiction policy for a flow network. *Naval Research Logistics Quarterly*, 18:37–45, 1971.

[28] B.L. Golden. A problem in network interdiction. *Naval Research Logistics Quarterly*, 25:711–713, 1978.

[29] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

[30] R.E. Gomory. An algorithm for integer solutions to linear programs. In R.L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, 1963.

[31] R. Hemmecke, R. Schultz, and D.L. Woodruff. Interdicting stochastic networks. In D.L. Woodruff, editor, *Network Interdiction and Stochastic Integer Programming*, pages 69–84. Kluwer Academic Publishers, Boston, 2002.

[32] K.L. Hoffman and M.W. Padberg. Improving LP-representations of zero-one linear programs for branch-and-bound. *ORSA Journal on Computing*, 3:121–134, 1991.

[33] E. Israeli and R.K. Wood. Shortest-path network interdiction. *Networks*, 40:97–111, 2002.

[34] E.L. Johnson. Modeling and strong linear programs for mixed integer programming. In S.W. Wallace, editor, *Algorithms and Model Formulations in Mathematical Programming*, 51, pages 1–41. NATO ASI, 1989.

[35] P. Kall and S.W. Wallace. *Stochastic Programming.* John Wiley & Sons, Chichester, 1994.

[36] H. Kopka and P.W. Daly. *A Guide To Latex.* Addison-Wesley, Dorchester, 3rd edition, 1994.

[37] A.H. Land and A.G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[38] K. Malik, A.K. Mittal, and S.K. Gupta. The k-most vital arcs in the shortest path problem. *Operations Research Letters*, 8:223–227, 1989.

[39] R.K. Martin. *Large Scale Linear and Integer Optimization: A United Approach*. Kluwer Academic Publishers, Boston, 1999.

[40] A.W. McMasters and T.M. Mustin. Optimal interdiction of a supply network. *Naval Research Logistics Quarterly*, 17:261–268, 1970.

[41] D.P. Morton and R.K. Wood. Restricted-recourse bounds for stochastic linear programming. *Operations Research*, 47:943–956, 1999.

[42] G.L. Nemhauser and L.E. Trotter, Jr. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6:48–61, 1974.

[43] G.L. Nemhauser and G.S. Sigismondi. A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of Operations Research Society*, 43:443–457, 1992.

[44] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.

[45] M.W. Padberg and G. Rinaldi. Optimization of a 532-city symmetric traveling salesman problem. *Operations Research Letters*, 6:1–7, 1987.

[46] M.W. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33:60–100, 1991.

[47] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

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

[49] F. Pan, W. Charlton, and D.P. Morton. Stochastic network interdiction of nuclear material smuggling. In D.L. Woodruff, editor, *Network Interdiction and Stochastic Integer Programming*, pages 1–19. Kluwer Academic Publishers, 2002.

[50] A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht, 1995.

[51] B.K. Reed. Models for proliferation interdiction response analysis. Master's thesis, Operations Research Department, Naval Postgraduate School, Monterey, California, 1994.

[52] R.T. Rockafellar and R.J-B Wets. A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming. *Mathematical Programming Study*, 28:63–93, 1986.

[53] R.T. Rockafellar and R.J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[54] A. Ruszczyński. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35:309–333, 1986.

[55] M.W.P. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45:831–841, 1997.

[56] K Shimizu, Y Ishizuka, and J.F. Bard. *Nondifferentiable and Two-Level Mathematical Programming*. Kluwer Academic Publishers, Boston, 1997.

[57] P.H. Vance, C. Barnhart, E.L. Johnson, and G.L. Nemhauser. Solving binary cutting stock problems by column generation and branch-and-bound. *Computational Optimization and Applications*, 3:111–130, 1994.

[58] R. Van Slyke and R.J-B Wets. *L*-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[59] A.R. Washburn and R.K. Wood. Two-person zero-sum games for network interdiction. *Operations Research*, 43:243–251, 1994.

[60] R. Weismantel. On the 0/1 knapsack polytope. *Mathematical Programming*, 77:49–68, 1997.

[61] R.D. Wollmer. Removing arcs from a network. *Journal of the Operations Research Society of America*, 12:934–940, 1964.

[62] L.A. Wolsey. *Integer Programming*. John Wiley & Sons, Inc., New York, 1998.

[63] D.L. Woodruff, editor. *Network Interdiction and Stochastic Integer Programming*. Kluwer Academic Publishers, Boston, 2003.

[64] R.K. Wood. Deterministic network interdiction. *Mathematical and Computer Modeling*, 17:1–18, 1993.

Vita

Feng Pan was born to Yizhi Zhang and Naide Pan on May 11, 1971 in Shanghai, China. After his high school, Feng moved to New Orleans, LA with his parents. Feng obtained a BS degree in Applied Mathematics at the University of Wisconsin at Madison in May 1994 and a MA degree in Applied Mathematics at the University of Texas at Austin in May 1998. Feng continued on to pursue a Ph.D in Operations Research/Industrial Engineering graduate program. Feng has been married to Mika Hirai since November 18, 1999.

Permanent address: 2906 Arrowhead Dr., Sugar Land, TX 77479

This dissertation was typeset with LaTeX[†] by Feng Pan.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.