Using Sensors to Interdict Nuclear Material Smuggling

David P. Morton
Graduate Program in Operations Research
The University of Texas at Austin, Austin, TX 78712, USA

Feng Pan
Infrastructure and Energy Analysis,
Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Abstract
We describe a stochastic network interdiction model for locating sensors that detect nuclear material. A nuclear material smuggler selects a path through a transportation network that maximizes the probability of avoiding detection. An interdictor installs sensors to minimize that maximum probability. We formulate this problem as a bi-level stochastic mixed-integer program, and then focus on a special case in which the underlying network is bipartite. We show that a class of valid inequalities, called step inequalities, can significantly reduce computational effort.

Keywords
stochastic programming, mixed-integer programming, network interdiction, nonproliferation, nuclear smuggling

1. Introduction
We model two adversaries, an interdictor and an evader, and an underlying network $G(N,A)$ on which the evader travels. In the deterministic version of our 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. The probability that the evader can traverse arc $(i,j) \in A$ undetected is $p_{ij}$ if the interdictor has not installed a sensor on arc $(i,j)$, and this probability is $q_{ij} < p_{ij}$ if the interdictor has installed a sensor on $(i,j)$. An evader can be caught by indigenous law enforcement without detection equipment, and so typically $p_{ij} < 1$. 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 sensors in order to minimize the probability the evader can travel from $s$ to $t$ undetected.

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 install the 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 $p^\omega$, over the population of possible evaders. We call this problem SNIP, for stochastic network interdiction problem. The timing of decisions and realizations in SNIP is as follows. First, the interdictor installs sensors on a subset of the network’s arcs. Then, a random origin-destination pair $(s^\omega,t^\omega)$ for the evader is revealed and the evader selects an $s^\omega$-$t^\omega$ path to maximize the probability of avoiding detection. The evader selects this path with full knowledge of the sensor locations and evasion probabilities.

The study of network interdiction in operations research began in the 1970s. During the Vietnam War, deterministic mathematical programs to disrupt flow of enemy troops and materiel were developed [9, 15]. The problem of maximizing an adversary’s shortest path is considered in [8, 10]. A closely-related problem concerns maximizing the longest path in an adversary’s PERT network [19]. In these linear programs, 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 [2, 6, 14]. Generalizations of the $k$-most-vital arcs problem, and associated solution procedures, are considered in [12]. The interdiction problem of removing arcs to minimize flow in an adversary’s maximum-flow network is considered in [20]. See [21] for game-theoretic approaches to related network interdiction problems, [5] for an interdiction model on a minimum-cost-flow network, and [13] for interdiction models of more general systems.
The above interdiction models are deterministic in the following senses. First, the arc lengths in the shortest-path and PERT problems, and the arc capacities in the maximum-flow problems, are known with certainty. Second, when increasing the length of an arc in the former problems or when removing or decreasing the capacity of an arc in the latter problem, these modifications occur in a deterministic manner, i.e., with certainty. In [7] the work of [20] on interdicting a maximum-flow network is generalized to allow for random arc capacities and random interdiction successes. An interdiction model with uncertain network topology is developed in [11].

2. SNIP Formulation
In this section we formulate SNIP and briefly discuss associated computational issues.

Network and Sets:
\( G(N, A) \) directed network with nodes \( N \) and arcs \( A \)
\( FS(i) \) set of arcs leaving node \( i \)
\( RS(i) \) set of arcs entering node \( i \)
\( AD \subseteq A \) arcs on which sensors may be placed

Data:
\( b \) total budget for installing sensors
\( c_{ij} \) cost of installing a sensor on arc \((i, j) \in AD\)
\( p_{ij} \) probability evader can traverse \((i, j)\) undetected with no sensor installed
\( q_{ij} \) probability evader can traverse \((i, j)\) undetected with a sensor installed

Random Elements:
\((s^\omega, t^\omega)\) realization of random origin–destination pair
\(\omega \in \Omega\) sample point and sample space
\(p^\omega\) probability mass function

Interdictor’s Decision Variables:
\(x_{ij}\) 1 if a sensor installed on arc \((i, j)\) and 0 otherwise

Evader’s Decision Variables:
\(y_{ij}\) positive only if evader traverses \((i, j)\) and no sensor is installed
\(z_{ij}\) positive only if evader traverses \((i, j)\) and a sensor is installed

Boundary Conditions:
\(x_{ij} \equiv 0 \quad (i, j) \notin AD\)
\(z_{ij} \equiv 0 \quad (i, j) \notin AD\)

Formulation:
\[
\min_{x \in X} \sum_{\omega \in \Omega} p^\omega h(x, (s^\omega, t^\omega)),
\] (1)
where \( X \) includes the budget constraint \( \sum_{(i,j) \in AD} c_{ij} x_{ij} \leq b \) and binary restrictions on \( x \), and where

\[
h(x, (s^\omega, t^\omega)) = \max_{y,z} y_{t^\omega}
\]
\[
\sum_{(s^\omega,j) \in FS(s^\omega)} (y_{s^\omega,j} + z_{s^\omega,j}) = 1\) (2a)
\[
\sum_{(i,j) \in FS(i)} (y_{ij} + z_{ij}) = \sum_{(j,i) \in RS(i)} (p_{ji} y_{ji} + q_{ji} z_{ji}), \quad i \in N \setminus \{s^\omega, t^\omega\}\) (2b)
\[
y_{t^\omega} = \sum_{(j,t^\omega) \in RS(t^\omega)} (p_{jt^\omega} y_{jt^\omega} + q_{jt^\omega} z_{jt^\omega})\) (2c)
\[
0 \leq y_{ij} \leq 1 - x_{ij} , \quad (i, j) \in A\) (2d)
\[
0 \leq z_{ij} \leq x_{ij} , \quad (i, j) \in AD.
\) (2e)
The optimal value, \( h(x, (s^\omega, t^\omega)) \), is the conditional probability the smuggler avoids detection, given \((s^\omega, t^\omega)\). The objective function in (1) is the expected value of this conditional evasion probability, where the expectation is taken over all possible origin-destination pairs.

Each link in the network on which a sensor can be placed is modeled as two arcs in parallel. If a sensor is installed, i.e., \( x_{ij} = 1 \), then flow may occur only on the “sensor” arc, through \( z_{ij} \). Conversely, if no sensor is installed then flow can only occur on the “no sensor” arc, via \( y_{ij} \). Flow on arc \((i, j)\) is multiplied by that arc’s gain (either \( p_{ij} \) or \( q_{ij} \)). So, if \( P_{s^\omega, t^\omega} \) is a path from \( s^\omega \) to \( t^\omega \) then
\[
y_{t^\omega} = \prod_{(i,j) \in P_{s^\omega, t^\omega}} \left[ p_{ij}(1 - x_{ij}) + q_{ij}x_{ij} \right]
\]
is the probability that an evader can travel from \( s^\omega \) to \( t^\omega \) on \( P_{s^\omega, t^\omega} \) without being detected. The evader’s goal is to select a path \( P_{s^\omega, t^\omega} \) that maximizes \( y_{t^\omega} \). The evader’s subproblem (2) accomplishes this by forcing one unit of flow out of \( s^\omega \) in (2b), enforcing flow conservation at all intermediate nodes in (2c), defining the flow that reaches \( t^\omega \) as \( y_{t^\omega} \) in (2d) and maximizing that value in the objective function. Flow is forced on the appropriate arc, and incurs the associated gain (actually, loss), by the interdictor’s decision variable \( x_{ij} \) in constraints (2e) and (2f).

When locating the sensors, the interdictor knows: (i) the network topology \( G(N, A) \), (ii) the indigenous detection probability on each arc, (iii) the detection probability given the presence of a sensor, (iv) the budget constraint, (v) the probability distribution governing the random \((s, t)\) pair, and (vi) the method by which the evader will select a path. After an \((s^\omega, t^\omega)\) realization is revealed, the evader selects an \( s^\omega-t^\omega \) path that maximizes the probability evading detection, knowing (i), (ii) and (iii) as well as the sensor locations.

Assuming that the smuggler solves an optimization model to select an \( s^\omega-t^\omega \) path is a behavioral assumption. While it is important to explore other possibilities (see, e.g., [16]), this is beyond the scope of this paper. We note that even in cases when our current assumption is not valid, the optimal value of (1) still provides a potentially useful pessimistic prediction of the evasion probability.

The SNIP model (1) with \( h \) defined in (2) is a bi-level stochastic mixed-integer program. In bi-level programs (e.g., [3, 4]) each player has 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 is trying to minimize that function and the evader is trying to maximize it. The problem 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. We show in [17] how this program can be reformulated as a single large-scale mixed-integer program. Here, we turn to an important special case of SNIP in which the problem can be reduced to a stochastic network interdiction model on a bipartite network.

3. SNIP on a Bipartite Network

Our work is motivated by the Second Line of Defense Program [1] in which potential sensor locations can be restricted to customs checkpoints leaving Russia. When potential sensor locations are limited in this manner, our SNIP model of the previous section can be simplified as described here.

Our underlying network model of Russia has four basic location entities: facilities from which sensitive nuclear material could be stolen, oblasts, i.e., Russian provincial regions, destinations outside Russia where a nuclear smuggler may desire to go, and customs checkpoints 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 sample point \( \omega \in \Omega \) specifies a facility-destination pair. In the SNIP model of Section 2, sensors are installed on arcs and this can be modeled by splitting each customs-checkpoint node into two nodes with an associated arc representing travel through the checkpoint.

The key to simplifying the formulation, when the customs checkpoints of a single country are under consideration, is that on each possible \( s^\omega-t^\omega \) 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 \((s^\omega, t^\omega)\). (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 can receive a sensor. Let $AD^{\omega} = \{(i,j) : (i,j) \in AD, (i,j) \in P^{\omega}\}$ be all such checkpoint arcs for $\omega \in \Omega$. The evader, under scenario $\omega$, must select an $s^{\omega}$-$t^{\omega}$ path, but this now depends on the sensor locations in a much simpler way than in the general model. For each $\omega$ we perform a preprocessing step to compute the value of the maximum-reliability path from $s^{\omega}$ to the tail of each checkpoint arc and the value of the maximum-reliability path from the head of each checkpoint arc to $t^{\omega}$. Call the product of these two probabilities $\gamma^{c}_{k}, k = (i,j) \in AD^{\omega}$. The probability the evader avoids detection under scenario $\omega$ is then

$$h(x,(s^{\omega},t^{\omega})) = \max_{k \in AD^{\omega}} \{\gamma^{c}_{k} p_{k}(1-x_{k}), \gamma^{c}_{k} q_{k} x_{k}\}. \quad (3)$$

By linearizing (3), we can express BiSNIP as the following stochastic mixed-integer program

$$\min_{x,\theta} \sum_{\omega \in \Omega} p^{\omega} \theta^{\omega}$$

s.t. $x \in X$

$$\theta^{\omega} \geq \gamma^{c}_{k} p_{k}(1-x_{k}), \; k \in AD^{\omega}, \omega \in \Omega$$

$$\theta^{\omega} \geq \gamma^{c}_{k} q_{k} x_{k}, \quad k \in AD^{\omega}, \omega \in \Omega. \quad (4)$$

BiSNIP (4) may be visualized on a bipartite network with node sets $\Omega$ and $\bigcup_{\omega \in \Omega} AD^{\omega}$. Arcs $(\omega,k)$ link each facility-destination pair, $\omega \in \Omega$, with its possible intermediate checkpoints, $k \in AD^{\omega}$. Excluding the possibility of being detected at the checkpoint, $\gamma^{c}_{k}$ is the evader’s probability of traveling from $\omega$’s facility to $\omega$’s destination, via $k$, undetected. This probability is multiplied by $q_{k}$ or $p_{k}$ depending on whether or not a sensor is at $k$. We show in [16] that the related decision problem is strongly NP-Complete.

Defining $r^{\omega}_{k} = (\gamma^{c}_{k} p_{k} - \max_{k \in AD^{\omega}} r^{\omega}_{k} q_{k})^{+}$, where $(\cdot)^{+} = \max(\cdot,0)$, model (4) simplifies to

$$\min_{x,\theta} \sum_{\omega \in \Omega} p^{\omega} \theta^{\omega}$$

s.t. $x \in X$

$$\theta^{\omega} \geq r^{\omega}_{k}(1-x_{k}), \; k \in AD^{\omega}, \omega \in \Omega. \quad (5c)$$

The reformulated model (5) is equivalent to (4) in that they have the same set of optimal solutions, although their objective functions differ by a constant. The linear programming relaxation of (5) can be tightened through addition of a class of valid inequalities that we term step inequalities. Let $k_{1}, \ldots, k_{\ell} \in AD^{\omega}$ satisfy 

$$r^{\omega}_{k_{1}} > r^{\omega}_{k_{2}} > \cdots > r^{\omega}_{k_{\ell}}.$$ Then, we define the associated step inequality

$$\theta^{\omega} \geq r^{\omega}_{k_{1}} - (r^{\omega}_{k_{1}} - r^{\omega}_{k_{2}}) x_{k_{1}} - \cdots - (r^{\omega}_{k_{\ell}} - 0) x_{k_{\ell}}. \quad (6)$$

When $\ell = 1$ the one-step inequality (6) is simply an existing constraint in (5c). When $\ell \geq 2$ the step inequalities are, in general, not redundant. The separation problem for step inequalities requires that, given a solution to the linear programming relaxation of (5) we either identify the (most) violated step inequality for each $\omega$ or determine that none are violated. This is done efficiently by solving a longest-path problem on an appropriately defined acyclic network [16]. Moreover, provided $k_{1}$ with the largest $r^{\omega}_{k_{1}}, k \in AD^{\omega}$, is included, [16] shows that (6) is a facet under further mild conditions.

### 4. Computation for BiSNIP

The test problem we consider has 85 facilities, 79 oblasts, 79 customs checkpoints and 9 destinations. (The two 79s are coincidental.) There are 30 oblasts with checkpoints and 34 oblasts with facilities. Facilities within an oblast are aggregated. Maintaining checkpoint integrity is important, so they are not aggregated. After facility aggregation, and allowing all possible facility-destination combinations, the model has $|\Omega| = 34 \cdot 9 = 306$ scenarios. Performing the reduction from SNIP to BiSNIP requires finding the maximum-reliability path from each facility to each customs site and from each customs site to each destination. (In our test problem the latter is trivial as each permissible customs site-destination combination is represented by a single arc.) These $34 \cdot 79 = 2686$ shortest paths are computed in about 3.5 seconds on a 1.7 GHz, Dell Xeon dual-processor machine with 2 Gb of memory. All computations reported here are on this computer.

We assume $c_{ij} = 1$, for all $(i,j) \in AD$, and solve our test problem for various values of the budget $b$. The separation procedure was coded in C++ and the MIP problems were solved with the CPLEX Concert
Table 1: The use of step inequalities leads to a significant reduction in computational effort.

<table>
<thead>
<tr>
<th>b</th>
<th>rel. gap (%)</th>
<th>comp. time</th>
<th>With Step Inequalities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rel. gap (%)</td>
<td>comp. time</td>
<td>no. of ≥ iters.</td>
</tr>
<tr>
<td>10</td>
<td>15.6</td>
<td>19</td>
<td>0.00</td>
</tr>
<tr>
<td>20</td>
<td>21.8</td>
<td>319</td>
<td>0.17</td>
</tr>
<tr>
<td>30</td>
<td>23.7</td>
<td>660</td>
<td>0.06</td>
</tr>
<tr>
<td>40</td>
<td>23.9</td>
<td>539</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>23.8</td>
<td>697</td>
<td>0.03</td>
</tr>
<tr>
<td>60</td>
<td>26.0</td>
<td>2133</td>
<td>0.24</td>
</tr>
<tr>
<td>70</td>
<td>29.0</td>
<td>6310</td>
<td>0.06</td>
</tr>
<tr>
<td>80</td>
<td>30.5</td>
<td>19629</td>
<td>0.26</td>
</tr>
<tr>
<td>90</td>
<td>31.3</td>
<td>6977</td>
<td>0.55</td>
</tr>
<tr>
<td>100</td>
<td>31.2</td>
<td>2628</td>
<td>0.55</td>
</tr>
<tr>
<td>120</td>
<td>28.2</td>
<td>280</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Technology libraries (version 9.0). All MIPs were solved with a relative tolerance of 0.1%. Table 1 displays the computational results for: (i) solving our MIP (5) 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 $100 \cdot \frac{z_{IP} - z_{LP}}{z_{IP}}$, "comp. time" reports total computation time in seconds, "no. of ≥" 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.

5. Conclusions

We have described a stochastic network interdiction model whose solution can be used to select sites to install sensors for detecting smuggled nuclear material. This work is motivated the Department of Energy’s Second Line of Defense Program. Our goal is to minimize the probability that an intelligent and informed smuggler can successfully travel through an underlying transportation network undetected. We focused on the special case of the model in which sensors can only be installed at border crossings of Russia. Our computational experience indicates the benefits of using a new class of valid inequalities called step inequalities.

Acknowledgements

This research was partially supported by the National Science Foundation under grants DMI-0228419 and DMI-0217927.

References


