LA-UR- 10-00541

Approved for public release; distribution is unlimited.

Title: Matrix Interdiction Problem

Author(s): Shiva Kasiviswanathan, CCS-3
Feng Pan, D-6

Intended for: Paper, Lecture Notes on Computer Science

Los Alamos NATIONAL LABORATORY
EST. 1943

Matrix Interdiction Problem

Shiva Kasiviswanathan* Feng Pan†

Abstract

In the matrix interdiction problem, a real-valued matrix and an integer k is given. The objective is to remove k columns such that the sum over all rows of the maximum entry in each row is minimized. This combinatorial problem is closely related to bipartite network interdiction problem which can be applied to prioritize the border checkpoints in order to minimize the probability that an adversary can successfully cross the border. After introducing the matrix interdiction problem, we will prove the problem is NP-hard, and even NP-hard to approximate with an additive n^γ factor for a fixed constant γ. We also present an algorithm for this problem that achieves a factor of $(n-k)$ multiplicative approximation ratio.

1 Introduction

In this paper, we introduce a combinatorial optimization problem, named *matrix interdiction*. The input to a matrix interdiction problem consists of a real valued matrix of dimension $m \times n$ and an integer $k \leq n$. The objective is to remove k columns from the matrix such that the sum over all rows of the maximum entry in each row is minimized. This combinatorial problem is closely related to a bipartite network interdiction problem in which limited resources are allocated to reduce the probability of nuclear material trafficking. The matrix interdiction problem turns out to be NP-hard, in fact it turns out that it is even NP-hard to get an additive n^γ approximation (for a fixed constant γ). Nevertheless, we show a simple greedy algorithm running in linear in the size of the input matrix can get a multiplicative $(n-k)$-approximation factor for the matrix interdiction problem.

Network interdiction is an active research area in operations research. Network interdiction problems can be viewed as a *Stackelberg game* on a network. There are two competitors, an evader and an interdictor, and the two competitors compete on an objective with opposing interests. The interdictor interdicts the network by modifying node and edge attributes on a network, and these modifications are usually constrained by limited resources. The evader then optimizes over the residual network. The origins of the network interdiction work can be traced back to 1970s when *minimizing maximum flow* models [12, 17] were developed to disrupt flow of enemy troops and supplies in the Vietnam War. Discrete version of maximum flow interdiction considers removing edges [25, 26] and is NP-hard. An another type of network interdiction problem is the *shortest path interdiction* where the goal is given that only a fixed number of edges (or nodes) can be removed, is to decide which set of edges (or nodes) to be removed so as to produce the largest increase in the shortest path between a source and a destination [11, 13, 16]. This problem is also known as the *most vital edges* (also *most vital nodes*) problem [6] and is also NP-hard [3].

There are a wide range applications of network interdiction models, including detecting drug smuggling [23], analyzing power grid vulnerability [22], and fighting infectious disease in hospital [2].

*CCS-3, Los Alamos National Laboratory, kasivisw@lanl.gov.

†D-6, Los Alamos National Laboratory, fpan@lanl.gov.

(a) (b)

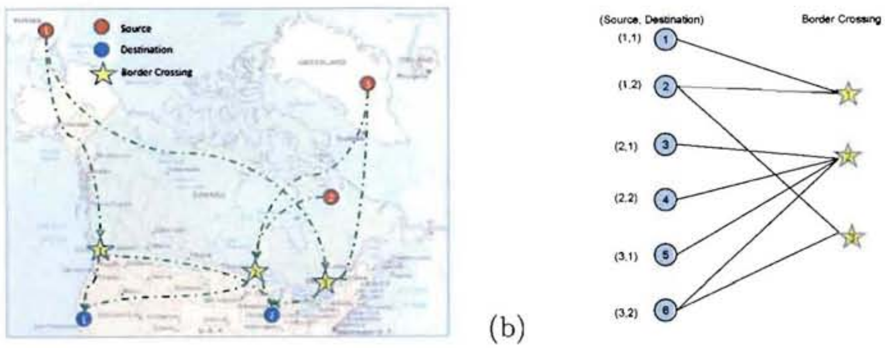

Figure 1: Bipartite network interdiction for border control (a) 3 sources and 2 destinations for smuggling attempts; 3 border crossings (b) equivalent bipartite network

Some recent network interdiction research has been motivated by homeland security applications. Researchers [24, 4] investigated how to allocate resources at individual ports to stop the illegal trafficking of WMDs (weapons of mass destruction). For the application of detecting smuggled nuclear material, interdiction models that *minimize maximum-reliability paths* on a transportation network [20, 18, 19] have been used to prioritize border crossings for the installation of radiation monitors. Stochastic models were developed to capture the uncertainties in the source-destination pairs [18], smuggling paths [14, 10], and physical characteristics of the detectors [9]. Minimizing maximum reliability path on general network can be applied to the cases where there are multiple layers of borders, but again this problem is NP-hard [19, 20]. For a single layer of border, the problem can be formulated as interdiction on bipartite network, which as we show in Section 2 is closely related to the matrix interdiction problem.

Many network interdiction problems are formulated as stochastic integer programs, and solution methods mainly involve the techniques from mixed integer programming. Benders decomposition is an efficient method to decompose the interdiction problem to smaller subproblems [8, 16], and valid inequalities are derived to strengthen the formulation [16, 21]. Approximation algorithms have been developed for some special types of the network interdiction problem like the maximum flow interdiction problem [5] and the graph matching interdiction problem [27].

The outline for the remaining paper is a follows. In Section 2, we will formally define the matrix interdiction problem and explain its relation to the bipartite network interdiction. A proof of NP-hardness is given in Section 3. In Section 4, we show the inapproximability result for the matrix interdiction problem, and in Section 5, we describe a greedy approximation algorithm for the matrix interdiction problem.

2 Matrix Interdiction

Let $[n]$ denote the set $\{1, \ldots, n\}$. For a matrix M of dimension $m \times n$, let $M_{i,j}$ denote the (i,j)th (ith row and jth column) entry of M. For a set $J \in [n]$, let $M|_J$ denote the submatrix of M obtained by picking only the columns of M present in J. Define,

$$val(M|_J) = \sum_{i=1}^{m} \max_{j \in J}\{M_{i,j}\}.$$

Definition 1 (Matrix Interdiction Problem). *Let M be an $m \times n$ matrix with entries from $\mathbb{R}$. Let $\mathcal{M}_s$ be the set of all submatrices of M with dimension $m \times n - k$. The matrix interdiction problem* *is to select a submatrix* $M^* \in \mathcal{M}_s$ *such that*

$$M^* = \text{argmin}_{J \subseteq [n], |J| = n-k} \left\{ \sum_{i=1}^{m} \max_{j \in J} \{M_{i,j}\} \right\}.$$

In other words, the matrix interdiction problem is to find an element M^* *from* $\mathcal{M}_s$ *with the property that*

$$val(M^*) = \min_{M_z \in \mathcal{M}_s} \{val(M_z)\}.$$

Bipartite Network Interdiction Problem. In the bipartite network interdiction problem, there is a single layer of border crossing (B) which separates sources (S) and destinations (T). An evader attempts to travel from a source to a destination, and any source-destination route will go through one and only one border crossing. Evasion may happen between any source-destination (s-t) pair, and every s-t pair is weighted probabilistically as p_{st}. At each edge, there is the edge reliability defined as the probability of traversing the edge without being captured, and the evader will use a route with the maximum reliability. For a triplet (s, b, t) where $s \in S$, $b \in B$ and $t \in T$, we can calculate the maximum reliability of a path from s through b to t and denote it as r_{sbt}. Interdiction of a border crossing b means to strengthen the security at the location, and as the result, $r_{sbt} = 0$ for all $s \in S$ and $t \in T$. The bipartite network structure is a result of the triplet by representing source-destination pairs as one set of nodes and border crossings as the other set of nodes. An edge in the bipartite network implies that there exists a path connecting the triplet. For example, in Figure 1, there are three sources, three border crossings, and two destinations. To go from source 1 to destination 1 the evader has to go through crossing 1, therefore, there is an edge between source-destination pair (1,1) and crossing 1. While, between source 1 and destination 2 the evader has the option of using either crossing 1 or 3, therefore, there are edges between source-destination pair (1,2) and crossings 1 and 3. Figure 1(b) shows the bipartite network. The maximum reliability for the triplet (1,1,2) is calculated by multiplying the maximum reliability between source 1 and crossing 1 and between crossing 1 and destination 2. A budgetary constraint limits the interdiction to k crossings, and the objective of the interdiction is to select k crossings in order to minimize the expected maximum probability of successful evasion between any pair of source and destination. This problem can be formulated as a bi-level integer program:

$$\min_{|X|=k, X \in \{0,1\}^{|B|}} \sum_{(s,t) \in S \times T} p_{st} \max_{b \in B} r_{sbt}(1 - x_b). \tag{1}$$

For more details on the bipartite network interdiction, see [19, 18].

We can construct a matrix M from bipartite network interdiction problem as follows. The dimension of M is set as $n = |B|$ and $m = |S| \times |T|$, and the entry $m_{ij} = r_{sjt} p_{st}$ where i is the node index in the bipartite network for source-destination pair (s, t). With this construction, the entries of M are positive real values between 0 and 1, and the optimal solution of the matrix interdiction problem for input M is exactly the k optimal border crossings to be interdicted in (1). Also, the two problems will also have the same optimal objective values. This leads to the following theorem.

Theorem 2. *Bipartite network interdiction problem is a special case of the matrix interdiction problem.*

3 NP-hardness Result

In this section, we show that the matrix interdiction problem is NP-hard. Thus, assuming P $\neq$ NP there exists no polynomial time algorithm that can exactly solve the matrix interdiction problem.

For establishing the NP-hardness we reduce the clique problem to the matrix interdiction problem. The clique problem is defined as follows.

Definition 3 (Clique Problem [7]). *Let $G = (V, E)$ be an undirected graph where V is the set of vertices and E is the set of edges of G. For a subset $S \subseteq V$, we let $G(S)$ denote the subgraph of G induced by S. A clique C is a subset of V such that the induced graph $G(C)$ is complete (i.e., $\forall u, v \in C$ an edge exists between u and v). The clique problem is the optimization problem of finding a clique of maximum size in the graph. As a decision problem, it requires us to decide whether there exists a clique of a given size k in the graph.*

Reduction from Clique to Matrix Interdiction Consider a graph $G = (V, E)$ with $|E| = m$ and $|V| = n$. We construct a matrix $M = M(G)$ of dimension $m \times n$ as follows: The rows of M correspond to the edges of G and columns of M correspond to the vertices of G. Let $e_1, \ldots, e_m$ be the edges of G. Now for every $l \in [m]$ consider the edge e_l, and let u and v be the end points of e_l. In the lth row of M add 1 in the columns corresponding to u and v, all other entries of the lth row are 0.

Notice that M has exactly two 1's in each row, and all the remaining entries of M are 0. Now,

$$val(M) = \sum_{i=1}^{m} \max_{j \in [n]} \{M_{i,j}\} = \sum_{i=1}^{m} 1 = m.$$

That is each edge in G contributes 1 to $val(M)$. Now if there exists a clique C of size k in G, then we can delete the columns of M corresponding to vertices in C and we obtain a submatrix M^* with $val(M^*) = m - \binom{k}{2}$ (because by deleting the columns corresponding to C, contribution to $val(M^*)$ will be 0 for the $\binom{k}{2}$ rows of M corresponding to the $\binom{k}{2}$ edges in C). Similarly, if the output to the matrix interdiction problem is a matrix M^* and if $val(M^*) > m - \binom{k}{2}$ then there exists no clique of size k in G and if $val(M^*) = m - \binom{k}{2}$ then there exists a clique of size k in G. The following two lemmas formalize the observations explained above.

Lemma 1. *Consider a graph G, and let $M = M(G)$ be the matrix as defined above. If there exists a clique C of size k in G, then there exists a submatrix M^* of M such that $val(M^*) = val(M) - \binom{k}{2}$ and M^* is a (optimum) solution to the matrix interdiction problem. Otherwise, if there exists no clique of size k in G then any (optimum) solution to the matrix interdiction problem will have a value strictly greater than $val(M) - \binom{k}{2}$.*

Proof. To show the first part of the lemma notice that each row of M contributes 1 to $val(M)$, or in other words each edge in G contributes 1 to $val(M)$. Consider a row of M, let us assume it corresponds to some edge (u, v) in G. Now notice that to obtain M^* if one only deletes the column corresponding to u or the column corresponding to v then the contribution of this row to $val(M^*)$ still remains 1 (because the row has two 1's and only one of these gets removed). So to reduce the contribution of this row to 0 one needs to delete columns corresponding to both u and v.

A clique C of size k has exactly $\binom{k}{2}$ edges between the vertices in C. Therefore, by deleting the columns corresponding to vertices in C, one can create a submatrix M^* of dimension $m \times n - k$ with $val(M^*) = val(M) - \binom{k}{2}$. We now argue that M^* is a (optimum) solution to the matrix interdiction problem. Consider a set $J \in [n], |J| = n - k$ and let $\bar{J} = [n] - J$. By deleting the columns of M which are in $\bar{J}$ to create $M|_J$ reduces $val(M)$ by the number of edges present between the vertices corresponding to entries in $\bar{J}$ (entries in $\bar{J}$ are the columns that are deleted from M to create $M|_J$ and each column corresponds to a vertex in G). In other words, $val(M|_J) = val(M) - e(\bar{J})$, where

$e(\bar{J})$ is the number of edges in G that are present between the vertices corresponding to entries in $\bar{J}$. Since, for any $\bar{J}$, $e(\bar{J}) \leq \binom{k}{2}$, therefore for all J,

$$val(M|_J) \geq val(M) - \binom{k}{2}.$$

Therefore, M^* whose value equals $val(M) - \binom{k}{2}$ is a (optimum) solution to the matrix interdiction problem.

To show the second part of the lemma, notice that if there exists no clique of size k in G, then for all $\bar{J}$,

$$val(M|_J) = val(M) - e(\bar{J}) > val(M) - \binom{k}{2},$$

as in the absence of a clique of size k, $e(\bar{J})$ is always less than $\binom{k}{2}$. □

Lemma 2. *Consider a graph G, and let $M = M(G)$ be the matrix as defined above. Let M^* be a (optimum) solution to the matrix interdiction problem with input M. Then if $val(M^*) = val(M) - \binom{k}{2}$ then there exists a clique of size k in G, and otherwise there exists no clique of size k in G.*

Proof. From Lemma 1, we know that $val(M^*) \geq val(M) - \binom{k}{2}$. Let $\bar{J}$ be the set of columns deleted from M to obtain M^*. If $val(M^*) = val(M) - e(\bar{J}) = val(M) - \binom{k}{2}$, then the vertices corresponding to entries in $\bar{J}$ form a clique of size k (as $e(\bar{J}) = \binom{k}{2}$). If $val(M^*) > val(M) - \binom{k}{2}$, then there exists no clique of size k in G because if there did exist a clique of size k in G then one can delete the columns corresponding to the vertices in the clique to obtain a matrix M_z with

$$val(M_z) = val(M) - \binom{k}{2} < val(M^*)$$

, a contradiction to the optimality of M^*. □

Theorem 4. *The matrix interdiction problem is NP-hard.*

Proof. The clique problem is NP-complete [7]. Lemmas 1 and 2 show a polynomial time reduction from the clique problem to the matrix interdiction problem. Therefore, the matrix interdiction problem is NP-hard. □

4 Inapproximability Result

In this section, we show that there exists a fixed constant γ such that the matrix interdiction problem is NP-hard to approximate to within an additive n^γ factor. More precisely, we show that assuming P $\neq$ NP there exists no polynomial time approximation algorithm for the matrix interdiction problem that can achieve better than an n^γ additive approximation. Note that this statement is stronger than Theorem 4. Whereas, Theorem 4 shows that assuming P $\neq$ NP there exists no polynomial time algorithm that can solve the matrix interdiction problem exactly, this inapproximability statement shows that unless P = NP it is not even possible to design a polynomial time algorithm which gives close to an optimum solution for the matrix interdiction problem.

To show the inapproximability we use a reduction similar to that in the previous section. It will be convenient to use a variant of the clique problem known as the k-clique.

Definition 5 (k-clique Problem). *In the k-clique problem the input consists of a positive integer k and a k-partite graph G (that is a graph that can be partitioned into k disjoint independent sets) along with its k-partition. The goal is to find the largest clique in G. The k-clique(G) is defined as ℓ/k, where ℓ is the size of the largest clique in G.*

Since in a k-partite graph G a clique can have at most one vertex in common with an independent set, the size of the largest clique in G is at most k. Therefore, k-clique$(G) \leq 1$.

Theorem 6 (Arora *et al.* [1]). *There exists a fixed $\delta > 0$ such that approximating the k-clique problem to within a multiplicative factor of n^δ is NP-hard.*

Proof. The proof presented in [1] (see also Chapter 10 in [15]) proceeds by showing a polynomial time reduction τ from the *SAT* problem (the problem of determining whether the variables of a Boolean formula can be assigned in a way that makes the formula satisfiable) to the k-clique problem. The reduction τ ensures for all instances I of SAT:

$$\begin{aligned} \text{If } I \text{ is satisfiable} &\Rightarrow \quad k\text{-clique}(\tau(I)) = 1, \\ \text{If } I \text{ is not satisfiable} &\Rightarrow \quad k\text{-clique}(\tau(I)) \leq \tfrac{1}{n^\delta}. \end{aligned}$$

Since, SAT is a NP-complete problem, therefore approximating the k-clique problem to within a multiplicative factor of n^δ is NP-hard (because if one can approximate the k-clique problem to within a multiplicative n^δ factor, then one can use τ to solve the SAT problem in polynomial time). □

The following lemma relates the problem of approximating the k-clique to approximating the matrix interdiction problem.

Lemma 3. *Let G be a k-partite graph and δ be the constant from Theorem 6. Let $M = M(G)$ be a matrix created from G as defined in Section 3. Let M^* be a (optimum) solution to the matrix interdiction problem with input M. Then*

$$\begin{aligned} \textit{If } k\textit{-clique}(G) = 1 &\Rightarrow \quad val(M^*) = val(M) - \tbinom{k}{2}, \\ \textit{If } k\textit{-clique}(G) \leq \tfrac{1}{n^\delta} &\Rightarrow \quad val(M^*) \leq val(M) - n^\delta \tbinom{k/n^\delta}{2} - \tbinom{n^\delta}{2}\left(\left(\tfrac{k}{n^\delta}\right)^2 - 1\right). \end{aligned}$$

Proof. If k-clique$(G) = 1$, then the size of the largest clique in G is k, and by deleting the columns corresponding to the vertices in this clique we get a submatrix M^* with $val(M^*) = val(M) - \binom{k}{2}$.

If the k-clique$(G) \leq 1/n^\delta$, then the size of the largest clique in G is at most k/n^δ. The maximum reduction to $val(M)$ occurs when there are n^δ cliques each of size k/n^δ and one deletes the k columns corresponding to the vertices appearing in all these cliques. Each clique has $\binom{k/n^\delta}{2}$ edges within itself. Since there are n^δ such cliques, this gives a total of $n^\delta\binom{k/n^\delta}{2}$ edges within the cliques. There are also edges across these n^δ cliques. Now across any two cliques there are at most $(k/n^\delta)^2 - 1$ edges, and since there are at most $\binom{n^\delta}{2}$ such pairs of cliques, this gives a total of $\binom{n^\delta}{2}((k/n^\delta)^2 - 1)$ edges across the cliques. In this case,

$$val(M^*) \leq val(M) - n^\delta \binom{k/n^\delta}{2} - \binom{n^\delta}{2}\left(\left(\frac{k}{n^\delta}\right)^2 - 1\right).$$

□

Theorem 7. *There exists a fixed constant γ, such that the matrix interdiction problem is NP-hard to approximate within an additive factor of n^γ.*

Proof. From Theorem 6, we know there exists a constant δ such that it is NP-hard to approximate the k-clique problem to within a multiplicative n^δ factor. From Lemma 3, we know that for a k-partite graph G, there exists a matrix $M = M(G)$ such that

$$\text{If } k\text{-clique}(G) = 1 \;\Rightarrow\; val(M^*) = val(M) - \left(\frac{k^2}{2} - \frac{k}{2}\right),$$
$$\text{If } k\text{-clique}(G) \le \frac{1}{n^\delta} \;\Rightarrow\; val(M^*) \le val(M) - \left(\frac{k^2}{2n^\delta} - \frac{k}{2}\right) - \left(\frac{k^2}{2} - \frac{k^2}{2n^\delta} - \frac{n^{2\delta}}{2} + \frac{n^\delta}{2}\right).$$

By comparing the above two equations, we see that if we can approximate the matrix interdiction problem with an $n^{2\delta}/2 - n^\delta/2$ additive factor, then we can approximate the k-clique problem to within an n^δ multiplicative factor. Since, the latter is NP-hard, it implies that an additive $n^{2\delta}/2 - n^\delta/2$ approximation of the matrix interdiction problem is also NP-hard. Setting γ such that, $n^\gamma = n^{2\delta}/2 - n^\delta/2$ proves the lemma. □

5 Greedy Approximation Algorithm

In this section, we present a greedy algorithm for the matrix interdiction problem that achieves an $(n-k)$ approximation factor. The input to the greedy algorithm is a matrix M of dimension $m \times n$ with entries from $\mathbb{R}$. The output of the algorithm is a matrix M_g. The running time of the algorithm is linear in the size of the input matrix.

ALGORITHM GREEDY(M)

1. For every $j \in [n]$, compute $c_j = \sum_{i=1}^{m} M_{i,j}$, i.e., c_j is the sum of the entries in the jth column.
2. Pick the top k columns ranked according to the column sums.
3. Delete the k columns picked in Step 2 to create a submatrix M_g of M.
4. Output M_g.

Theorem 8. *Algorithm Greedy is a multiplicative $(n-k)$ approximation algorithm for the matrix interdiction problem. More precisely, the output M_g of Greedy(M) satisfies the following inequality*

$$val(M_g) \le (n-k)val(M^*),$$

where M^* is a (optimum) solution to the matrix interdiction problem with input M.

Proof. Let $Soln \subseteq [n], |Soln| = n-k$ be the set of $n-k$ columns present in M_g. Let $Opt \subseteq [n], |Opt| = n-k$ be the set $n-k$ columns present in M^*. Now,

$$\begin{aligned}
val(M_g) &= \sum_{i=1}^{m} \max_{j\in Soln}\{M_{i,j}\} \le \sum_{i=1}^{m}\sum_{j\in Soln} M_{i,j} \\
&= \sum_{j\in Soln}\sum_{i=1}^{m} M_{i,j} \le \sum_{j\in Opt}\sum_{i=1}^{m} M_{i,j} \\
&= \sum_{i=1}^{m}\sum_{j\in Opt} M_{i,j} \le \sum_{i=1}^{m}(n-k)\max_{j\in Opt}\{M_{i,j}\} \\
&= (n-k)\sum_{i=1}^{m}\max_{j\in Opt}\{M_{i,j}\} \\
&= (n-k)val(M^*).
\end{aligned}$$

The second inequality follows because the Greedy algorithm deletes the k-most columns ranked according to column sums. The third inequality follows because for any real vector $v = (v_1, \ldots, v_{n-k})$, $\sum_{p=1}^{n-k} v_p \leq (n-k)\max\{v\}$.

The above argument shows that the Greedy algorithm achieves an $(n-k)$ multiplicative approximation factor for the matrix interdiction problem. □

6 Conclusion

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):555, 1998.

[2] N Assimakopoulos. A network interdiction model for hospital infection control. *Comput Biol Med*, 17(6):413–22, 1987.

[3] A. Bar-Noy, S. Khuller, and B. Schieber. The complexity of finding most vital arcs and nodes. Technical report, University of Maryland, College Park, MD, USA, 1995.

[4] E. Boros, L. Fedzhora, P.B. Kantor, K. Saeger, and P. Stroud. Large scale lp model for finding optimal container inspection strategies. *Naval Research Logistics Quarterly*, 56(5):404–420, 2009.

[5] Carl Burch, Robert Carr, Sven Krumke, Madhav Marathe, Cynthia Phillips, and Eric Sundberg. A decomposition-based pseudoapproximation algorithm for network flow inhibition. In *Network Interdiction and Stochastic Integer Programming*, volume 22 of *Operations Research/Computer Science Interfaces*, pages 51–68. Springer US, 2003.

[6] H. W. Corley and D. Y. Sha. Most vital links and nodes in weighted networks. *Operations Research Letters*, 1(4):157 – 160, Sep 1982.

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms.* The MIT press, 2001.

[8] Kelly J. Cormican, David P. Morton, and R. Kevin Wood. Stochastic network interdiction. *Oper. Res.*, 46(2):184–197, 1998.

[9] N. Dimitrov, D. P. Michalopoulos, D. P. Morton, M. V. Nehme, F. Pan, E. Popova, E. A. Schneider, and G. G. Thoreson. Network deployment of radiation detectors with physics-based detection probability calculations. *Annals of Operations Research*, 2009.

[10] Nedialko B. Dimitrov and David P. Morton. Combinatorial design of a stochastic markov decision process. In *Operations Research and Cyber-Infrastructure*, volume 47 of *Operations Research/Computer Science Interfaces*, pages 167–193. Springer US, 2009.

[11] D. R. Fulkerson and Gary C. Harding. Maximizing the minimum source-sink path subject to a budget constraint. *Mathematical Programming*, 13(1):116–118, December 1977.

[12] P. M. Ghare, D. C. Montgomery, and W. C. Turner. Optimal interdiction policy for a flow network. *Naval Research Logistics Quarterly*, 18(1):37, 1971.

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

[14] Alexander Gutfraind, Aric Hagberg, and Feng Pan. Optimal interdiction of unreactive markovian evaders. In *CPAIOR '09: Proceedings of the 6th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 102–116, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] D.S. Hochbaum, editor. *Approximation algorithms for NP-hard problems.* PWS Publishing Co., Boston, MA, USA, 1997.

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

[17] A. W. McMasters and T. M. Mustin. Optimal interdiction of a supply network. *Naval Research Logistics Quarterly*, 17(3):261, 1970.

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

[19] F. Pan. *Stochastic Network Interdiction: Models and Methods.* PhD dissertation, University of Texas at Austin, Operations Research, 2005.

[20] F. Pan, W. Charlton, and D. P. Morton. Interdicting smuggled nuclear material. In D.L. Woodruff, editor, *Network Interdiction and Stochastic Integer Programming*, pages 1–19. Kluwer Academic Publishers, Boston, 2003.

[21] F. Pan and D. P. Morton. Minimizing a stochastic maximum-reliability path. *Networks*, 2008. Online - *to appear in print.*

[22] J. Salmeron, K. Wood, and R. Baldick. Worst-case interdiction analysis of large-scale electric power grids. *Power Systems, IEEE Transactions on*, 24(1):96–104, 2009.

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

[24] Lawrence M. Wein, Alex H. Wilkins, Manas Baveja, and Stephen E. Flynn. Preventing the importation of illicit nuclear materials in shipping containers. *Risk Analysis*, 26(5):1377–1393, 2006.

[25] Richard Wollmer. Removing Arcs from a Network. *OPERATIONS RESEARCH*, 12(6):934–940, 1964.

[26] R. K. Wood. Deterministic network interdiction. *Mathematical and Computer Modoling*, 17:1–18, 1997.

[27] Rico Zenklusen. Matching interdiction, 2008.