LA-UR- 08-6551

Approved for public release; distribution is unlimited.

Title: INTERDICTION OF A MARKOVIAN EVADER

Author(s): ARIC HAGBERG
ALEXANDER GUTFRAIND
DAVID IZRAELEVITZ
FENG PAN

Intended for: PROCEEDINGS:
ALENEX 09 WORKSHOP ALGORITHM ENGINEERING AND EXPERIMENTS

Los Alamos
NATIONAL LABORATORY
EST. 1943

Interdiction of a Markovian Evader

Alexander Gutfraind* Aric Hagberg† David Izraelevitz‡ Feng Pan§

Abstract

Network interdiction is a combinatorial optimization problem on an activity network arising in a number of important security-related applications. It is classically formulated as a bilevel maximin problem representing an "interdictor" and an "evader". The evader tries to move from a source node to the target node along the shortest or safest path while the interdictor attempts to frustrate this motion by cutting edges or nodes. The interdiction objective is to find the optimal set of edges to cut given that there is a finite interdiction budget and the interdictor must move first. We reformulate the interdiction problem for stochastic evaders by introducing a model in which the evader follows a Markovian random walk guided by the least-cost path to the target. This model can represent incomplete knowledge about the evader and the graph as well as partial interdiction. We formulate the optimization problem for this model and show how, by exploiting topological ordering of the nodes, one can achieve an order-of-magnitude speedup in computing the objective function. We also introduce an evader-motion-based heuristic that can significantly improve solution quality by providing a global view of the network to approximation methods.

1 Introduction

Mathematical modeling of network interdiction was originally introduced in the study of military supply chains and interdiction of transportation networks [12, 16]. The problem is currently studied in different classes of networks and in a variety of contexts, and finds applications in countering of nuclear proliferation programs [17], control of infectious diseases [20], and disruption of terrorist networks [10]. The underlying networks may represent transportation networks, but more generally may be social or activity networks. Recent interest in the problem has been in part due to the threat of smuggling of nuclear materials and devices [18]. In the case of nuclear smuggling, interdiction might correspond to the installation of special radiation-sensitive detectors along the selected transportation edges.

The problem is often posed in terms of two agents called "interdictor" and "evader" where the evader attempts to minimize some objective function in the network, *e.g.* distance, cost, or risk when traveling from network location s to location t, while the interdictor attempts to limit success by removing network nodes or edges. The interdictor has limited resources and can thus only remove a finite set of nodes or edges. In the simplest formulation, the interdictor seeks to identify a set of edges (or nodes) on the network whose removal maximizes the cost of the least-cost path from a source to a destination node, while the evader seeks to find and traverse the best unimpeded path. This interdiction problem is known as the "most vital edges" (or "most vital nodes") problem [9] and it has been shown to be NP-hard [3] and hard to approximate [6]. Methods for solving network interdiction problems have included exact algorithms for solving integer programs, such as branch-and-bound, as well as decomposition methods to rebuild the network by iteratively adding relevant paths to reduce the size of both the underlying network and the number of binary decision variables. A more recent approach, based on structure-dependent cutting planes, exploits the relationship between the ordered set of evading paths and binary interdiction variables [19].

A common assumption in many studies is that there is perfect knowledge about hard-to-compute network parameters, such as the cost to the evader to traverse a network edge in terms of resource consumption or probability of detection. However, it is clear that the evader, and, to a lesser extent, the interdictor, have unreliable and incomplete information about the network and edge weights. This undermines the classical approach which assumes that the evader follows just the optimal (*e.g.* least-cost) evasion path because the constitution of this path can be highly sensitive to network parameters. Indeed, under uncertainty the evader can only be described in probabilistic terms. By constructing such probabilistic evader models one can expect to develop more robust interdiction solutions. Any type of uncertainty places the interdiction problem within stochastic optimization, where one seeks to find those edges that are vital *on average*. This problem of stochastic interdiction has been the focus of a number of recent studies [17, 1, 5, 14, 21].

Failure to account for evader uncertainty can lead to suboptimal decisions, namely, solutions that do not maximize the expected cost of the evader to reach the target. In the network in Fig. 1 there are four paths from the source to the target: through nodes $1,2,3$ and the direct path $(0,5)$ with costs $9,8,8$ and 8.01, respectively. If we can only remove one edge the solution in the classical formulation is to remove edge $(4,5)$ which increases the path length from 8.0 to 8.01. Now suppose an evader would take, with equal probability, any path whose cost is ≤ 1.25 times the cost of the least-cost path. In other words, the evader is unable to determine exactly which one of those paths has the least-cost. If so, interdiction at $(4,5)$ would actually *decrease* the expected cost from ≈ 8.25 to 8.01. This is because it would

*Center for Applied Mathematics, Cornell University, Ithaca, New York 14853 (ag362@cornell.edu).

†Mathematical Modeling and Analysis, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545 (hagberg@lanl.gov).

‡Los Alamos National Laboratory, Los Alamos, New Mexico 87545 (izraelevitz@lanl.gov).

§Risk Analysis and Decision Support Systems, Los Alamos National Laboratory, Los Alamos, New Mexico 87545 (fpan@lanl.gov).

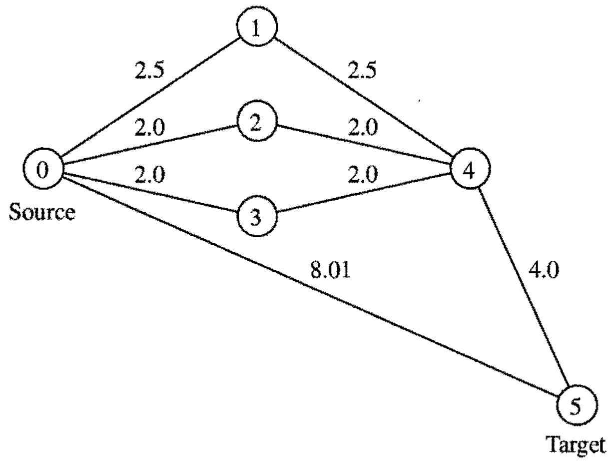

Figure 1: Example network where the shortest path interdiction formulation produces a suboptimal solution when interdicting a single edge. Interdicting that edge $(4,5)$ decreases the expected path cost. Interdicting any one of $(0,2)$, $(2,4)$,$(0,3)$, or $(3,4)$ increases the expected path cost.

cut off the costly path through node 1. The optimal choice is actually to interdict any one of $(0,2)$, $(2,4)$, $(0,3)$, or $(3,4)$. This choice would increase the expected cost from ≈ 8.25 to ≈ 8.33.

In this paper we propose a Markovian network interdiction framework which can capture a wide range of network behaviors by the evader (Sec. 2). We then specialize the general framework to a simple model based on low-level evader decision-making processes (Sec. 3). Finally we develop efficient heuristic algorithm for the interdiction problem using predicted evader motion (Sec. 4).

2 The general interdiction model

Our formulation is a stochastic generalization of the classical max-min shortest path interdiction problem [12, 16]. In the classical formulation an evader wants to traverse a network from an origin s to a destination t. Let p be some path between s and t in a graph $G(N,E)$ with the set of nodes N and the set of weighted edges E. Let $c(p)$ be the path cost computed by summing the costs on each of the edges in p.[1] The cost C_{ij} of an edge (i,j) is used here interchangeably with edge weight. The costs are assumed to be given in the problem and may depend on direction (in the case that $G(N,E)$ is a digraph).

The network interdiction strategy is represented by choosing elements r from the feasible interdiction set R which is typically a subset of the edge set E with a limited size B. We set the value of $r_{ij}=1$ if edge (i,j) is interdicted, and $r_{ij}=0$ otherwise. Let $D_{ij}\geq 0$ be the added cost of traversing (i,j) when it is interdicted. When the value of D_{ij} is very large all paths avoid the interdicted edge (i,j) (assuming that there is an alternative path) which effectively removes the edge (i,j) from the graph.

In the classical model, the evader only travels on least-cost paths, and is fully aware of interdiction decisions. We denote the increased cost of traversing path p given an interdiction strategy r as $c_r(p)$. Thus if r intersects p, then typically $c_r(p) >= c(p)$. Given the set of edges E_p on the path $p\in PT$, $c_r(p)=c(p)+\sum_{(i,j)\in E_p} D_{ij}r_{ij}$ and we need to solve the optimization problem

$$\max_{r\in R}\min_{p\in PT} c_r(p)\,. \tag{1}$$

This formulation is for edge interdiction but a similar problem can be considered for node interdiction by introducing node costs D_i.

A stochastic version of the interdiction problem can be constructed by supposing that an evader may take any path from s to t, according to some probability distribution, rather than always choosing the least-cost path. Randomness in evader path decision could be caused by uncertainty about interdiction decisions r or network costs, mistaken cost computations, or possibly by intent to increase unpredictability. The path p becomes a random variable distributed as $Pr(p)$, and the expected cost of traveling from s to t is then

$$E(c)=\sum_{p\in PT} Pr(p)c(p)\,. \tag{2}$$

The interdiction problem becomes

$$\max_{r\in R}\sum_{p\in PT} Pr(p|r)c_r(p)\,, \tag{3}$$

where $Pr(p|r)$ is now the probability of traversing a path given the interdiction set r and note that $Pr(p|r)$ implicitly contains the evader's strategy. The classical optimization problem (1) is clearly just a special instance of (3) when the expectation is conditioned on traversal of only least-cost paths.

2.1 Markovian evaders In order to compute $Pr(p|r)$ values it is necessary to develop stochastic evader models. In the following we consider, for simplicity, the problem of a Markovian evader. Complete information about such an evader is encoded in a distribution of starting nodes, a, and a Markovian transition probability matrix, $\mathbf{P}$. An element P_{ij} of this matrix is the probability that an evader at node i will move along edge (i,j). The distribution of starting nodes is assumed to be given and independent of the interdiction strategy r, while the $\mathbf{P}$ matrix is assumed to be determined as soon as the graph and r are known. This Markovian assumption simplifies cost computations as we show below

[1] The additivity of costs is natural for resources such as money or time, but it also holds for simple models of risk. Suppose cost $c(p)$ measures the probability of detection on p, which is 1 less the probability of evasion on p. If $q(p)$ is the probability of evasion on this path then set $c(p)=-\log(1-q(p))$. If the probabilities of detection on all edges of p are independent, then the probability of detection on p, $1-q(p)$, is just an exponential of negative the sum of the costs C_{ij} along each edge on p.

but, in general, the evader may have transition probabilities node i to node j that depend not only on the current evader location and target but also on the history of previous moves.

To compute the expected path cost $E(p)$ with a given a and $\mathbf{P}$ we first construct a list of paths that reach t in order of increasing length.[2] It is convenient in the following calculations to introduce a new matrix $\mathbf{M}$ to be the same as the transition matrix $\mathbf{P}$, but with a single entry M_{tt} set to 0. This $\mathbf{M}$ can be interpreted as an evader model where the evader is removed from the network when reaching the target t. Suppose a path from s to t is specified by the edge sequence $(s,a),(a,b),...,(z,t)$ (see for example Fig. 2). The conditional probability that the evader will traverse this path is $M_{sa}M_{ab}...M_{zt}$. The cost accumulated along this path is

$$C_{sa} + C_{ab} + ... + C_{zt}\,, \tag{4}$$

where C_{ij} would include the cost of passing an interdicted edge if (i,j) is interdicted. Let π_n be the probability vector whose j coordinate is the probability that a path of length n begins at s and ends at j. Thus, π_n is the sum of the probabilities of all paths of length n that end at j. Since $\mathbf{M}$ defines a Markov chain, $\pi_n = \pi_{n-1}\mathbf{M} = \pi_0\mathbf{M}^n = a\mathbf{M}^n$, where a is the distribution over the starting nodes. If all paths must begin at s, then a is just the unit vector in the s direction.

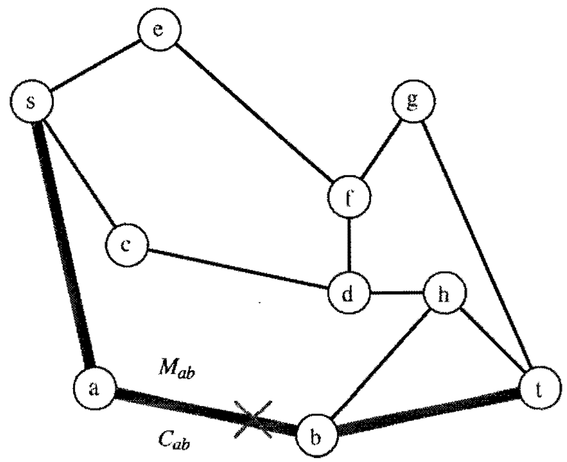

Figure 2: The evader's goal is to travel from node s to node t and the interdictor's task is to remove edges (or nodes) to increase the expected path traversal cost to the evader. The probability of the evader crossing the marked edge (a,b) is M_{ab} with an associated traversal cost of C_{ab}. The costs and probabilities are set by the network structure and the evader model.

Let the vector h_n represent the expected accumulated costs for paths of length n. The j coordinate $[h_n]_j$ is the expected cost accumulated by a path with length n that terminates at j

$$[h_n]_j = \sum_i [h_{n-1}]_i M_{ij} + [\pi_{n-1}]_i M_{ij} C_{ij}\,. \tag{5}$$

The entire vector can be written as

$$h_n = h_{n-1}\mathbf{M} + \pi_{n-1}(\mathbf{C}\odot\mathbf{M})\,, \tag{6}$$

where $\mathbf{C}\odot\mathbf{M}$ is the matrix formed by element-wise "Hadamard" multiplication of $\mathbf{C}$ and $\mathbf{M}$. The expected accumulated cost of paths of any length is given by $h := \sum_{n=0}^{\infty} h_n$, when the sum converges. Noting that $\sum_{n=0}^{\infty}\pi_n = \sum_{n=0}^{\infty}\pi_0\mathbf{M}^n = a(\mathbf{I}-\mathbf{M})^{-1}$, and summing Eq. (6) over all n gives

$$h = a(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{C}\odot\mathbf{M})(\mathbf{I}-\mathbf{M})^{-1}\,, \tag{7}$$

where $\mathbf{I}$ is the identity matrix. Equation 7 is key to our approach: its t^{th} element, $[h]_t$, expresses in closed form the expected cost of paths starting at s and ending at t given the evader's movement model. Each part of Eq. (7) formula has an intuitive meaning: the vector $a(\mathbf{I}-\mathbf{M})^{-1}$ is the expected number of times that each of the nodes is visited by the evader when starting at a distribution a [13, p.419]; the vector $a(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{C}\odot\mathbf{M})$ is the expected cost of reaching each of the nodes from their immediate predecessor nodes; and h gives the expected cost of reaching each of the nodes from the starting distribution a.

The interdiction objective, according to our approach, is to maximize $[h]_t$. Because the interdiction variable r affects the costs and then the matrix $\mathbf{M}$ this results in the nonlinear optimization problem

$$\max_{r\in R}\left[a(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{C}\odot\mathbf{M})(\mathbf{I}-\mathbf{M})^{-1}\right]_t\,. \tag{8}$$

This expression can be generalized for the case of multiple evaders where evader k has certain probability $w^{(k)}$ of occurring ($\sum_k w^{(k)} = 1$), as well as a distinctive source distribution $a^{(k)}$, target node $t^{(k)}$ and transition matrix $\mathbf{M}^{(k)}$. The generalized objective would be a weighted sum of Eq. (7).

3 Least-cost-guided evader model

In order to solve interdiction problems, it is necessary to develop a concrete model of evader behavior, namely, to specify $\mathbf{M}$, and this is our next task. In general, model design for the Markovian evader is constrained by the requirement that $\sum_{n=0}^{\infty} h_n$ converges. Fortunately, it is sufficient to know that any node is visited at most a finite number of times (because $\|h\| \leq \left\|a(\mathbf{I}-\mathbf{M})^{-1}\right\|_1 \max_{(i,j)\in E} C_{ij}$.) Hence, the sum converges if under the model $a(\mathbf{I}-\mathbf{M})^{-1}$ is well-defined. For a general source distribution a this corresponds to the existence of $(\mathbf{I}-\mathbf{M})^{-1}$. This is guaranteed if the target node is an absorbing state of the Markov chain defined by $\mathbf{P}$, namely if: (a) upon reaching the target the evader never leaves and (b) for each source site, the target node is

[2] The length of a path p is the number of edges in the path, while the cost $c(p)$ of a path is the sum of the costs of the edges. These are equal only if the cost of each edge is unity.

reached with non-zero probability after finitely many steps [13, Sec.11.2][3].

Recall that stochasticity in evader motion may have two causes: first, the evader has limited information about the network topology, interdiction decisions and the costs/risks along alternative paths and second, the evader may be intentionally trying to make unpredictable moves to make the task of the interdictor harder. In order to capture the former stochasticity, suppose that the errors of the evader are random rather than systematic. This implies that the network known to us gives an average of the networks that the evader might perceive. If so, the probability M_{ij} that a cost- or risk-minimizing evader at node i would traverse $i \longrightarrow j$ *increases* with the probability of (successful) evasion on the shortest path to the target through this edge, q_{ij}. Suppose then that M_{ij} is found by taking a monotonically increasing function $\phi : q_{ij} \longrightarrow \mathbb{R}$, and setting $M_{ij} \propto \phi(q_{ij})$. One choice is to assume that an evader would choose edge (i, j) with probability *proportional* to q_{ij}, or more generally, proportional to a positive power of q_{ij}

$$M_{ij} \propto \left(\frac{q_{ij}}{q_{i*}}\right)^{\lambda}, \tag{9}$$

where $\lambda > 0$ is a parameter, $q_{i*} = \max_j q_{ij}$ is the probability of evasion if the shortest path from i to the target is followed, and the constant of proportionality is found from $\sum_j M_{ij} = 1$ (for an illustration, see Fig.3.) When $\lambda \longrightarrow \infty$ the evader moves deterministically along the least-risk path and when $\lambda \longrightarrow 0$ the motion is perfectly random. The least-risk path has the highest probability, but the difference with other paths vanishes as $\lambda \longrightarrow 0$. Hence, the model can be called the "least-cost-guided evader". The parameter λ represents the precision of the information the evader has about the graph and interdiction decisions.

In many cases the values q_{ij} and q_{i*} are not known directly but are instead found by relating them to edge costs and applying Dijkstra's algorithm. One approach is to find the cost of the path through j, z_{ij}, and the cost of the least-cost path from i, z_{i*} (see Eq. 4.) Then the probabilities of evasion may be computed from the cost by the relation $q_{ij} = e^{-z_{ij}}$. Substitution into Eq. 9 yields

$$M_{ij} \propto e^{-\lambda(z_{ij} - z_{i*})}. \tag{10}$$

Notice that although M_{ij} values depend on the cost of least-cost paths, this dependence is smooth rather than a step function of the classical evader model. This model is similar to one developed for routing in ad-hoc wireless networks. In that application $\mathbf{M}$ is used to determine where to transmit a message when the final destination cannot be reached directly [4].

[3]The $\mathbf{M}$ matrix here is the $\mathbf{Q}$ matrix in Grinstead and Snell's formulation [13] except for a small detail: $\mathbf{Q}$ includes only the transitions between non-absorbing states, but here $\mathbf{M}$ does include the absorbing state - the target node t. As a compensation, we impose $\mathbf{M}_{tt} = 0$ implying that the evader is removed from the graph upon reaching t.

3.1 Model with no backtracking A useful variant of this model is to make the reasonable assumption that evaders never backtrack, that is, move away from the target node t or move to an already-visited node. Hence, we assume that there is zero probability of motion through (i, j) if i is at least as close to the target as j as defined by $c(i) \leq c(j)$, where $c(i)$ and $c(j)$ are the smallest costs of paths to the target from nodes i and j computed using Eq. (4). This assumption implies that the evader would never cross a node or an edge twice. Consequently the set of nodes becomes a partially ordered set and as a result, there exists a relabeling σ (*i.e.* a basis) of the nodes such that if $c(i) > c(j)$ then $\sigma(i) > \sigma(j)$. A simple (non-unique) procedure is to take the target node as $\sigma(t) = 0$ and then rank the nodes in the order of their distance *i.e.* cost along least-cost path to t, breaking ties arbitrarily. Computationally, this is the same as the order they are reached by Dijkstra's algorithm that starts at t. The transition probability becomes (where α is the normalization)

$$M_{ij} = \begin{cases} \alpha e^{-\lambda(z_{ij} - z_{i*})} & c(i) > c(j) \\ 0 & c(i) \leq c(j) \end{cases}.$$

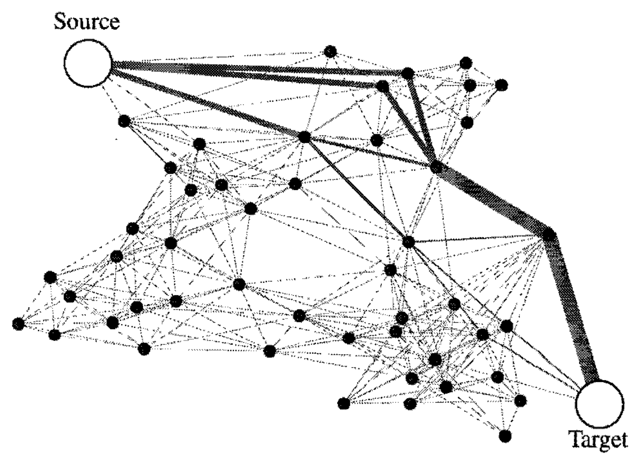

Figure 3: Illustration of the evader model in Eq.(10) on a 50-node network. The probabilities that an evader would pass any one of the edges are indicated by the width of the (red) edges. The thin (gray) edges have probability zero and thus not traversed at all. The evader has $\lambda = 0$ and does not backtrack.

This no-backtracking assumption implies that all paths must reach the target after at most $|N| - 1$ steps, where $|N|$ is the number nodes in G, and hence $\mathbf{M}$ becomes nilpotent of power $|N| - 1$. Moreover, by labeling the nodes from 0 up in order of increasing cost, $\mathbf{M}$ can be written as a lower-triangular matrix with zero diagonal. For example, if the evader traverses a 2x3 grid graph with the target in any corner node then one possible σ gives the matrix

$$\mathbf{M} = \begin{pmatrix} 0 & & & & & \\ 1 & 0 & & & & \\ 1 & 0 & 0 & & & \\ & 1 & 0 & 0 & & \\ & .5 & .5 & 0 & 0 & \\ & & & .5 & .5 & 0 \end{pmatrix}.$$

This special structure of $\mathbf{M}$ facilitates an order-of magnitude speedup in the computation of Eq. 7. For a general $\mathbf{M}$, computing $a(\mathbf{I}-\mathbf{M})^{-1}$ involves Gaussian elimination at a cost of $\frac{2}{3}|N|^3$. For a nilpotent lower-triangular $\mathbf{M}$ the cost falls to $|N|^2$ since we can use backward-forward substitutions instead of Gaussian elimination. The cost of computing the objective function Eq. 7 is also expected to drop to $O(|N|^2)$ despite the need to reorder the matrix $\mathbf{C}$ when the nodes are relabeled.

4 Solving the Markovian interdiction problem

The challenge of network interdiction consists of developing both realistic models and tractable algorithms. The above evader model adds realism to interdiction but does not reduce the computational complexity of interdiction. The general model is computational hard because in the limit of $\lambda \to 0$, the model reduces to the least-cost interdiction problem which is NP-Hard [2, 3] and also hard to approximate [6]. Therefore, in this section we discuss approximation algorithms. We consider general purpose algorithms such as simulated annealing, as well as heuristics based on evader motion and network structure.

4.1 Local search algorithms Local search algorithms have been successfully used to solve many combinatorial optimization problems. In general, local search algorithms take a random solution (or a population thereof) and improve it by a series of incremental changes. Some of the the most frequently used local search algorithms are simulated annealing (SA), genetic algorithm (GA), Tabu search (TS) [15]. In addition to those we consider here a randomized greedy algorithm (RGA) which constructs a solution incrementally, (one edge at a time). At each increment in the RGA an edge is added to the interdiction set which is the best of a random sample of edges (see Alg. 1). The sample size L is typically much smaller than the graph size because of the high cost of computing the change Δ_S in the objective function. By running the local algorithms several times it is possible to improve the likelihood of finding the optimal set.

We compared the local search algorithms on a set of 50 network interdiction problems. In each problem the underlying network structure was constructed randomly using a 100-node Geographical Threshold Graph (GTG) model. The GTG model has been used as a model for wireless networks and may be appropriate as a model for random transportation networks [7]. We set the threshold parameter in the GTG model at $\theta = 30$ to produce sparse graphs but connected graphs; resulting in graphs of around 800 edges.

Algorithm 1 RGA construction of the interdiction set S with budget B and sample size L

```
S ← ∅
while B > 0 do
    W ← {L random elements from E \ S}
    for all e_i ∈ W do
        Δ_S(e_i) := h(S ∪ {e_i}) − h(S)
    S ← S ∪ {argmax_{e_i∈W} Δ_S(e_i)}, resolving ties arbitrarily.
    B ← B − 1
Output(S)
```

To make the comparison fair, all algorithms were allowed to perform the same number of cost evaluations and all had their parameters tuned. To compare solutions across several different networks with different interdiction complexities, the solution on each network was normalized by the solution found on this network with a benchmark algorithm. For the benchmark, we used a variant of RGA, namely RGA with $L = |E|$ so the "sample" contains all of the network's edges. This exhaustive search allows this algorithm to find solutions superior to anything the four algorithms can find (at a vast increase in computational cost.)

As can be seen in Fig. 4, the four algorithms gave comparable performance. In another experiment with a small sample of other types of graphs, it was found that only *SA*, *RG*, and *RGA* gave comparable performance, while the performance of the genetic algorithm was significantly inferior in percentage and statistical senses to the other three algorithms (data not shown). The relatively high variance in all heuristics is likely due to sensitivity of the solution quality to randomness in sample choice. In absolute sense, the performance of all four algorithms was relatively poor. The average solution was 83% of the benchmark algorithm, which itself likely falls short of the optimum (possibly by a large gap).

In general local search algorithms are not a promising approach for large problems. Given the exponential size of the solution space it is unlikely that they would be able to explore even a small fraction of it in a reasonable number of optimization runs. Moreover, the solution space is quite rugged: there are synergies (non-linear gains) when multiple edges are interdicted on paths going to a single target.

4.2 Graph heuristics The poor performance of the local search algorithms suggests that fast high-quality approximation algorithms can only come from more specialized solvers that exploit the structure of the interdiction problem. We now discuss two heuristics, RGAH-flow and RGAH-betweenness, that build on top of the RGA algorithm above. The idea in both is to make the RGA algorithm above more efficient; instead of randomly sampling from $E \setminus S$ we choose edges from that set that are ranked highly by a heuristic as described in Alg.2. In Alg.2, $H_S(e_i)$ is the value of the heuristic (*e.g.* edge betweenness of e_i) on the network G modified by interdicting the set S. We give two examples below of heuristics that hold promises for our interdiction problem.

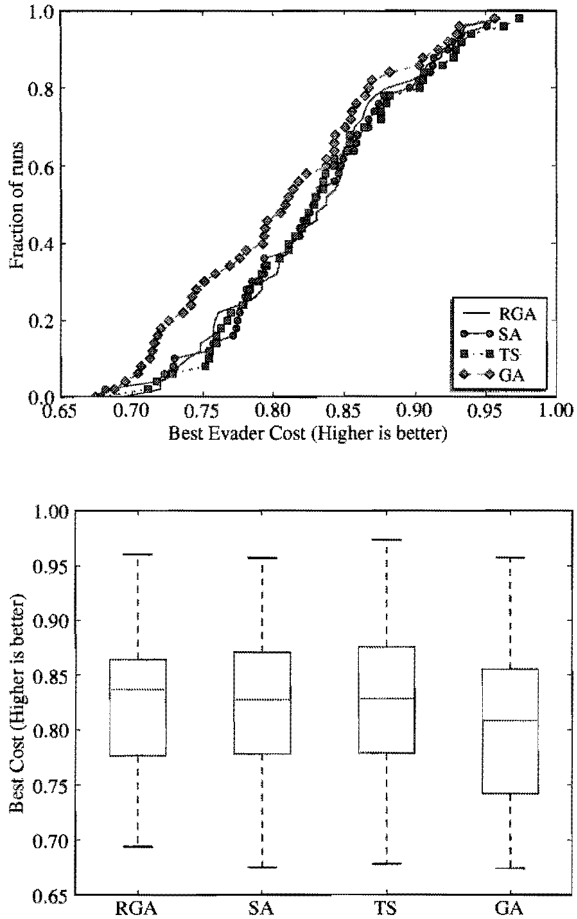

Figure 4: A comparison of of local search algorithms on sample interdiction problems. (top) The cumulative distribution function of the highest evader cost for 50 runs of each algorithm. The overlap between the data series indicates that no algorithm stochastically dominates the other three. (bottom) The distribution of best evader costs. The box contains runs from the lower to upper quartile values of the data. The high overlap between the boxes indicates that all of the algorithms have comparable performance. In all runs, the edge costs were $\mathbf{C}_{ij} = 1$ and interdiction doubled them to 2 ($\mathbf{D}_{ij} = 1$.) There were two evaders with uniformly distributed probability of starting from the set of source nodes. For half the problems we used $\lambda = 1$ and the other $\lambda = 1000$. The interdictor budget was $B = 6$ and the sample size was $L = 20$. The evader cost in both graphs was scaled by the solution of a benchmark algorithm (see text).

Algorithm 2 RGAH with budget B and sample size L

$S \leftarrow \emptyset$
while $B > 0$ **do**
  $W \leftarrow \{\lfloor \frac{L}{2} \rfloor$ random elements from $E \setminus S\}$
  $W \leftarrow R \cup \{e_i \,|\, e_i \in$ top $\lfloor \frac{L-1}{2} \rfloor$ elements ranked by $H_S(e_i)\}$
  for all $e_i \in W$ **do**
    $\Delta_S(e_i) := h(S \cup \{e_i\}) - h(S)$
  $S \leftarrow S \cup \{\text{argmax}_{e_i \in W} \Delta_S(e_i)\}$, resolving ties arbitrarily.
  $B \leftarrow B - 1$
Output(S)

Notice that Alg.2 selects some but not all of the edges using the heuristic. Allowing some edges to be selected randomly ensures that the algorithm is not deterministic: determinism would have prevented it from exploring the entire set of feasible solutions even in principle (for complete exploration replace argmax with stochastic selection). Also, the reason RGAH is allowed only $L-1$ edges rather than L in the sample was to keep the comparison of the algorithms fair, namely, to ensure that the basic algorithm (RGA) and its heuristic-modified versions (RGAH) have about the same asymptotic computational cost (more on the cost of the heuristic below.) Thus, if the solutions found by RGAH have better quality than those of RGA then it is because the heuristics are able to identify interdiction locations that are better than random.

The RGA does not provide performance guarantees. Indeed, on some graphs its solution is arbitrarily poor as a fraction of the optimal solution. Nevertheless, RGA performed as well as the other local search algorithms and it is probably the simplest of them all so it is a logical foundation for building heuristic algorithms. It would be interesting to incorporate heuristics into algorithms other than RGA.

4.2.1 Evader flow One heuristic approach is to exploit the fact that edges likely to be traversed by the evader are also likely to be good interdiction locations. Interdiction of such edges will compel the evader to take alternative paths which might be considerably more costly. Another argument in support of this heuristic is from the reverse: it would be wasteful to interdict a low-likelihood edge *i.e.* an edge not likely to be traversed by the evader because such an interdiction is not going to increase the evader's cost. Naturally any heuristic based on evader motion in the current graph has limited ability to predict evader motion on the graph after edges have been interdicted; it may be possible for the evader to take an alternative path with little or no cost penalty. Fortunately, this is not a serious problem since within RGAH, the heuristic is only the first computational step. After selection of the top likelihood edges (and the random sample), we compute the gain from interdicting each one of them, and then take the one with the best gain (that is, greatest increase in evader cost.)

For the flow heuristic, we rank each edge (i,j) based on the the expected number of times the evader is likely to traverse it, $a(\mathbf{I}-\mathbf{M})^{-1}M_{ij}$. These values can be computed for all edges in the network simultaneously at a cost of $O(|N|^2)$. In contrast, to compute the evader cost in Eq. (7) for all edges would cost $O(|E|\cdot|N|^2)$. The λ value in the heuristic was set to be the same as of the evader itself. In general, it is possible that at least on some graphs setting λ to be lower than the evader's own would give better exploration of the solution space and hence better results.

4.2.2 Betweenness centrality Another heuristic approach is to use network betweenness centrality, originally introduced to measure the importance of nodes in a social networks [11], to rank the edges. Betweenness centrality is the fraction of shortest paths (or least-cost paths in weighted networks) between all pairs of nodes in a network that cross a given node (or edge). This metric identifies those edges that are critical to connectivity within a network because they participate in a large number of least-cost paths linking nodes on a network, such as a bridge edge that joins two graph components. Interdiction of such edges is likely to significantly increase the evader's cost. This heuristic is less specific than the direct evader motion (the flow) because it takes all pairs of nodes and not just the sources and target of the evader. Yet, it may be more robust to the evader bypassing an interdicted edge. The computational cost of this heuristic is $O(|N||E|+|N|^2\log|N|)$ [8] which is comparable to the flow heuristic.

4.2.3 Performance of heuristic algorithms For the comparison of the RGAH algorithms we used the same 50 example interdiction problems introduced earlier. The results of the comparison show that the RGAH with flow heuristic stochastically dominates the other algorithms and provides considerable improvement in solution quality as shown in Fig. 5. This advantage was maintained in both $\lambda=0$ and $\lambda=1000$ cases (data not shown). The gain from the heuristic is significant: the normalized cost of RGAH with the flow heuristic is very close to 1.0 which implies that it finds solutions almost indistinguishable from solutions found by an algorithm using $39.5 = 790/20$ times as many cost evaluations. In contrast, the betweenness heuristic does not provide any significant gain in performance.

5 Further work

The evader model developed here is a first step toward a more refined model that more closely ties evader motion with its computational and informational constraints. Research into more refined models promises further gains in computational performance and realism. The algorithm performance data

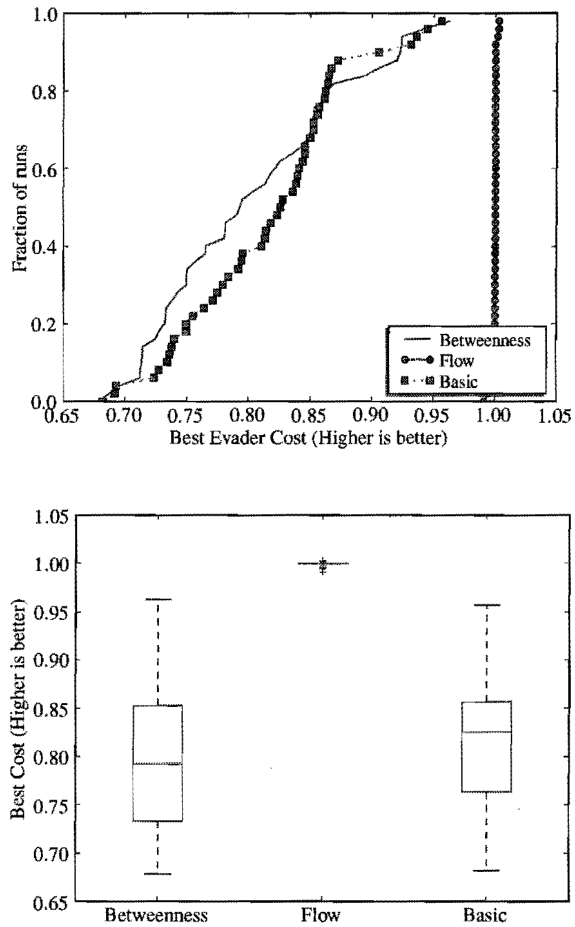

Figure 5: Comparison between the basic RGA algorithm and two variants which use a heuristic to choose edges. (top) The cumulative distribution function of the highest evader cost for 50 runs of each algorithm. (bottom) The distribution of best evader costs. The box contains runs from the lower to upper quartile values of the data. A one-tailed t-test supports the advantage of RGAH-flow over the other two algorithms with $p<0.0001$ and $p<0.0001$ for RGAH-betweenness and basic RGA, respectively. The scale is normalized as in Fig. 4.

suggests that heuristics can be very valuable for solving the interdiction problem. However, it would be extremely useful for network interdiction and other applications to construct a method for identifying edges that would be *vital*, that is, cannot be cheaply bypassed by the evader. This could potentially eliminate the requirement of computing the objective function and keeping the sample size small. Another important objective would be to develop better theoretical understanding of the performance of the evader flow heuristic and to bound its performance.

Acknowledgements

AG would like to thank David Shmoys for suggesting the use of randomized greedy algorithm for interdiction, and Vadas Gintautas for countless fruitful discussions. Part of this work was funded by the Department of Energy at Los Alamos National Laboratory under contract DE-AC52-06NA25396 through the Laboratory Directed Research and Development Program.

References

[1] MICHAEL P. ATKINSON, ZHENG CAO, AND LAWRENCE M. WEIN, *Optimal stopping analysis of a radiation detection system to protect cities from a nuclear terrorist*, Risk Analysis, 28 (2008), pp. 353–371.

[2] MICHAEL O. BALL, BRUCE L. GOLDEN, AND RAKESH V. VOHRA, *Finding the most vital arcs in a network*, Oper Res Lett, 8 (1989), pp. 73–76.

[3] A. BAR-NOY, S. KHULLER, AND B. SCHIEBER, *The complexity of finding most vital arcs and nodes*, tech. report, University of Maryland, College Park, MD, USA, 1995.

[4] CHRISTOPHER L. BARRETT, STEPHAN J. EIDENBENZ, LUKAS KROC, MADHAV MARATHE, AND JAMES P. SMITH, *Parametric probabilistic routing in sensor networks*, Mobile Networks and Applications, 10 (2005), pp. 529–544.

[5] H. BAYRAK AND M.D. BAILEY, *Shortest path network interdiction with asymmetric information*, Networks, 52 (2008), pp. 133–140.

[6] E. BOROS, K. BORYS, AND V. GUREVICH, *Inapproximability bounds for shortest-path network intediction problems*, tech. report, Rutgers University, Piscataway, NJ, USA, 2006.

[7] MILAN BRADONJIĆ AND JOSEPH S. KONG, *Wireless ad hoc networks with tunable topology*, in Forty-Fifth Annual Allerton Conference, UIUC, Illinois, USA, 2007.

[8] U. BRANDES, *A faster algorithm for betweenness centrality*, Journal of Mathematical Sociology, 25 (2001), pp. 163–177.

[9] H. W. CORLEY AND D. Y. SHA, *Most vital links and nodes in weighted networks.*, Oper Res Lett, 1 (1982), pp. 157 – 160.

[10] JONATHAN D. FARLEY, *Breaking Al Qaeda cells: A mathematical analysis of counterterrorism operations (a guide for risk assessment and decision making)*, Studies in Conflict and Terrorism, 26 (2003), pp. 399–411.

[11] L. C. FREEMAN, *A set of measures of centrality based on betweenness*, Sociometry, 40 (1977), pp. 35 – 41.

[12] P. M. GHARE, D. C. MONTGOMERY, AND W. C. TURNER, *Optimal interdiction policy for a flow network*, Naval Research Logistics Quarterly, 18 (1971), p. 37.

[13] CHARLES M. GRINSTEAD AND J. LAURIE SNELL, *Introduction to Probability*, Amer Math Soc, USA, second revised ed., Jul 1997.

[14] U. JANJARASSUK AND J. LINDEROTH, *Reformulation and sampling to solve a stochastic network interdiction problem*, Networks, (2008). to appear.

[15] JAMES P. KELLY, *Meta-Heuristics: Theory and Applications*, Kluwer Academic Publishers, Norwell, MA, USA, 1996.

[16] A. W. MCMASTERS AND T. M. MUSTIN, *Optimal interdiction of a supply network*, Naval Research Logistics Quarterly, 17 (1970), pp. 261–268.

[17] D. P. MORTON, F. PAN, AND K. J. SAEGER, *Models for nuclear smuggling interdiction*, IIE Transactions, 39 (2007), pp. 3–14.

[18] F. PAN, W. CHARLTON, AND D. P. MORTON, *Interdicting smuggled nuclear material*, in Network Interdiction and Stochastic Integer Programming, D.L. Woodruff, ed., Kluwer Academic Publishers, Boston, 2003, pp. 1–19.

[19] F. PAN AND D. P. MORTON, *Minimizing a stochastic maximum-reliability path*, Networks, 52 (2008), pp. 111–119.

[20] B. POURBOHLOUL, L.A. MEYERS, D.M. SKOWRONSKI, M. KRAJDEN, D.M. PATRICK, AND R.C. BRUNHAM, *Modeling control strategies of respiratory pathogens*, Emerg Infect Dis, 11 (2005), pp. 1246–56.

[21] DANIEL REICH AND LEO LOPES, *The most likely path.* preprint, 2008.