Published in final edited form as:
Ann Oper Res. 2014 November 1; 222(1): 341–359. doi:10.1007/s10479-013-1372-x.

Evader Interdiction: Algorithms, Complexity and Collateral Damage

Matthew P. Johnson[1,*], **Alexander Gutfraind**[2], and **Kiyan Ahmadizadeh**[3]

[1]Pennsylvania State University

[2]University of Illinois at Chicago

[3]Cornell University

Abstract

In network interdiction problems, evaders (e.g., hostile agents or data packets) are moving through a network toward targets and we wish to choose locations for sensors in order to intercept the evaders. The evaders might follow deterministic routes or Markov chains, or they may be reactive, i.e., able to change their routes in order to avoid the sensors. The challenge in such problems is to choose sensor locations economically, balancing interdiction gains with costs, including the inconvenience sensors inflict upon innocent travelers. We study the objectives of (1) maximizing the number of evaders captured when limited by a budget on sensing cost and, (2) capturing all evaders as cheaply as possible.

We give algorithms for optimal sensor placement in several classes of special graphs and hardness and approximation results for general graphs, including evaders who are deterministic, Markov chain-based, reactive and unreactive.

A similar-sounding but fundamentally different problem setting was posed by Glazer and Rubinstein where both evaders and innocent travelers are reactive. We again give optimal algorithms for special cases and hardness and approximation results on general graphs.

Keywords

network interdiction; bridge policy; submodular set cover; Markov chain; minimal cut; four color theorem

1 Introduction

In network interdiction problems, one or more *evaders* (e.g., smugglers or terrorists, or hostile data packets) travel through a network, beginning at some initial locations and attempting to reach some targets. Our goal is to stop them. We do so by placing *sensors* on nodes in hopes that most or all of the evaders will pass by a sensor and thus be captured (or *intercepted*) before reaching their destinations (formal problem definitions are given in the "Preliminaries" section below).

*This work was performed in part while visiting Los Alamos National Laboratory.

We take as a given the evader movement dynamics, which may be either deterministic (each evader specified by a route from source to target) or stochastic, e.g. each evader specified by a Markov chain whose states are the nodes of the network. Evader e_i induces a subgraph $G_i \subseteq G$ in which she roams, according to the probabilities specified by her Markov chain. An unreactive or *oblivious* evader [15] behaves the same regardless of the choice of sensor locations (or *interdiction* sites), and so her set of possible routes can be construed as objects we wish to pierce. In contrast, the reactive evader observes the locations of the sensors and can change her motion.

We try to make economical use of the sensors—i.e., to balance the benefits of security (the interdiction of many or all evaders) with the total cost (widely defined) of doing so. The cost of placing a sensor at a given node can incorporate the cost of the device itself, the effort or danger involved in performing the placement, and the inconvenience it causes to any innocent travelers subjected to it. If traffic flow estimates on the graph's edges are known for both evaders and innocent travelers, then it is natural to try to place sensors where they will intercept many evaders but inconvenience few innocents. If a sensor acts as a checkpoint, capturing the evaders but examining and then letting pass the innocents, then the inconvenience cost can be incorporated directly into the node's sensor placement cost since placing two sensors on an innocent's path inconveniences her twice. In this model we study two natural objectives: (1) maximizing the (expected, weighted) number of evaders captured while respecting a budget on sensing cost, and (2) capturing *all* evaders (with probability 1) as cheaply as possible. In the latter case evaders may be *reactive*, i.e., able to observe the sensor locations and choose a different path in G_i. Regardless, e_i is guaranteed to be captured only if her target node is separated from all her source nodes *within subgraph* G_i. By "separated" we mean that all paths of e_i of positive probability pass at least one sensor (see Preliminaries below). We solve these problems optimally in several special graph settings and give hardness and approximation results in general settings.

In contrast, allowing the innocents to react to sensor locations changes the character of the problem significantly. In this setting we study a special case of the problem (suggested by Glazer and Rubinstein [12]) where there are a collection of bridges crossing a river, with each traveler p restricted to using some set $\sigma(p)$ of bridges (because of p's preferences or geography, say), and the task is to decide which bridges to open and close. This can be viewed as a special case of our network setting in which every travel path is of length 2 but with the restriction that sensors cannot be placed on a traveler's start node (see Fig. 1). Note that in this special case, sensors can also be viewed as roadblocks, in the sense that placing a sensor on a node effectively means deleting the node from the network for evader and innocent alike.

An instance of the bridges problem is specified by a set system with realvalued elements that may be either positive or negative, corresponding to the value or cost (respectively) of capturing evaders or blocking innocents. Several possible objective functions could be considered, such as capturing all evaders while blocking as few innocents as possible or capturing as many evaders as possible given a budget allowing a certain number of blocked innocents. We obtain approximation results for several versions of this problem.

Preliminaries

Given is a directed graph $G(V, E)$ with $|V| = n$ unless otherwise noted, used by travelers of two types: evaders (or *bads*) and innocents (or *goods*). (These terms are used interchangeably.) Each person p can travel within some subgraph $G_p \subseteq G$. Depending on the setting, sensors can be placed on nodes or edges to capture the *flow* passing through. All the sensors are placed before the moves of the reactive persons. A person p's Markov chain determines the probability weight $f_{p,v}$ of p's traffic through each node v. If oblivious, p is unable to shift her flow $f_{p,v}$ from the path going through v to some other path, so placing a sensor at v captures all of $f_{p,v}$ (or at least whatever portion of it was not captured upstream). In some settings we assume all innocents, all evaders, or both are *oblivious*, as discussed below.

We emphasize that *reactive* indicates a two-stage setting in which all the sensors are placed and then p can choose an unblocked path in G_p if one exists. We also emphasize that person p is restricted to subgraph G_p regardless of whether p is oblivious or reactive, an evader or an innocent.

Edge and Node Interdiction—In *edge interdiction*, sensors are installed on edges and are represented by a matrix of decision variables $\mathbf{r}$: $r_{uv} = 1$ if (u, v) has a sensor placed at it (with cost c_{uv}) and $r_{uv} = 0$ otherwise. If an evader crosses an edge with a sensor she is detected with probability 1. In *node interdiction*, placing a sensor on node u (with cost c_u) means setting $r_{uv} = 1$ for every edge (u, v), that is, interdicting all evaders *leaving* u. A sensor on a target node does *not* protect that node itself but will stop evaders as they pass through it.

The node and edge settings are equivalent in general, directed graphs with location-varying costs, in the sense that a problem in one setting can be transformed into the other [15].

Oblivious Evaders—An evader is specified in terms of the probabilities of her taking various routes, where a route is a walk (possibly containing cycles) ending at a target node. A Markovian evader is represented by a Markov chain given by an initial *source* distribution $\mathbf{a}$ over nodes and a transition probability matrix $\mathbf{M}$. The matrix $\mathbf{M}$ has the property that a specified target node t is a killing state: upon reaching t the evader is removed from the network. Under mild restrictions on the Markov chain (such as, t is an absorbing state), the probability of capturing the evader as a function of r can be expressed in closed form [15]:

$$J(\mathbf{a}, \mathbf{M}, \mathbf{r}) = 1 - \left(\mathbf{a}[\mathbf{I} - (\mathbf{M} - \mathbf{M} \odot \mathbf{r})]^{-1}\right)_t \quad (1)$$

where the symbol $\odot$ indicates element-wise (Hadamard) multiplication. This formulation generalizes to a setting of multiple simultaneous evaders, each realized with probability w_e, or equivalently having weight $w_i > 0$ representing the importance of capturing her. The probability of capturing e_i is denoted by $J_i(\mathbf{r})$.

Definition 1: *An evader e_i is specified by a $(\mathbf{M_i}, \mathbf{a_i})$ pair. Evader e_i is* deterministic *if from each of her possible starting nodes, $\mathbf{M_i}$ specifies a single next node with probability 1, and is* nondeterministic *(or stochastic) otherwise. In both cases*, $\mathbf{a_i}$ may specify multiple starting points with positive probability.

Budgeted Interdiction (BI)—The BI objective is to capture as many evaders as possible, given a budget on sensors. More precisely, suppose we have a bound on the number of nodes we can monitor (or on their total cost, with costs always scaled to be integral). Any choice of some subset of nodes to observe determines a probability that a given evader will be captured (i.e., that she will pass through at least one observed node) prior to reaching her target *t*. The task in Budgeted Interdiction is to maximize the expected (weighted) number of evaders interdicted, subject to a budget *B* on sensor costs:

$$\text{maximize}\sum_{i} w_i J_i(\mathbf{r}) \ \text{such that} \sum_{u} r_u c_u \leq B$$

The special case of BI where evaders follow unreactive Markov chains is termed the Unreactive Markovian Evader (UME) interdiction problem [15].

Full Interdiction (FI)—This problem seeks a minimum-cost set of nodes to observe in order to capture all of the evaders with probability one.

$$\text{minimize}\sum_{u} r_u c_u \ \text{such that} \ \forall i, J_i(\mathbf{r}) = 1$$

Interdiction with reactive evaders and innocents—In this setting the graph is traversed by both evaders (“bads”) and innocent travelers (“goods”). Both types of users are reactive, which means that a traveler *p* is captured only if *all* her paths *within* G_p from source nodes to target node have received a sensor placed on some node prior to the target; otherwise, she succeeds. The interdiction policy here aims to find the optimal balance between allowing the goods to pass and blocking passage to the bads.

We focus on a special variant termed the Bridges Problem, where each path from the source to target passes through exactly one other node, termed *bridge*. We use *n* to denote the number of bridges and $\sigma(p)$ the set of bridges accessible to user *p*. A weight $\hat{w}_p$ assigned to each person (positive for bads, negative for goods).

One possible formulation is to minimize the total error: the weighted number of bads crossing and goods unable to cross. To to be precise, let *N* and *D* indicate the sets of goods the bads, respectively, and let binary variables $x_p = 1$ indicate *p*’s success and $y_s = 1$ indicate that bridge *s* is open. To minimize the total error we solve:

$$\text{minimize}\sum_{p\in D}\hat{w}_p x_p + \sum_{p\in N}\hat{w}_p(1-x_p) \ \text{such that} \ x_p \leq \sum_{s\in\sigma(p)} y_s \ \forall p \ \text{and} \ x_p \geq y_s \ \forall p, \forall s \in \sigma(p)$$

In this formulation the constraints implement the requirement that a traveler *p* will cross if and only if at least one of her bridges is open.

More generally, we use *TP*, *FP*, *TN*, *FN* to indicate the weighted numbers of true positives (bads failing to cross), false positives (goods failing), true negatives (goods succeeding), and false negatives (bads succeeding), respectively. The above minimal error formulation is expressed as minimization of $FP + FN$, and we will explore other combinations like maximization of net flow: $TN - FN$. We denote the total weight of goods and bads as $W_N = FP + TN$ and $W_D = TP + FN$, respectively. In general capturing the bads might have, for example, disproportionally large weight.

Contributions

With oblivious evaders and innocents, we solve the budgeted problem optimally in path and cycle graphs (Sec. 2). With oblivious innocents (and evaders either oblivious or not), we solve the full interdiction problem optimally in paths, cycles, and trees.

In general graphs, we give hardness results including showing that the budgeted problem is NP-hard with even one Markovian evader (Thm. 7). In contrast, we show that full interdiction with a single evader is in P. (Sec. 3) The problem remains hard with two or more evaders even when the evaders follow acyclic paths (Thm. 6). This stems from an unexpected connection to 4-colorability. With m possible evader paths, the problem is H_m-approximable (Cor. 1, where $H_m = \sum_{i=1}^{m} \frac{1}{i}$), which is essentially optimal, given certain complexity-theoretic assumptions.

When both evaders and innocents are reactive, we optimally solve a special case in which the graph and travelers' sets of paths can be represented by *bridges* and *convex bridge sets*, respectively (Sec. 4). We also show that $FP + FN$ is approximable with factor one plus the maximum size of any innocent's bridge set (Thm. 9).

The following table summarizes the key results. For brevity, we state only the key hypotheses and use the abbreviations *S* for stochastic, *D* for deterministic, *FI* for Full Interdiction and *BI* for budgeted interdiction. Node and edge interdiction often, but not always, have similar complexity [15].

Setting	Result	Existing work
Path graph	S FI on nodes is $O(n \log n)$ D BI on nodes is $O(Bnm)$ S BI on nodes is $O(Bn^2m)$	$O(n \log OPT)$ [20, 27] - -
Tree graph	S FI on nodes is $O(n^3)$	Related [13, 25]
Arbitrary graphs	FI on edges of Markov evader is in *P* BI is NP-hard with 1 evader BI is NP-hard with 2 acyclic evaders	- q evaders [15] ibid.
Bridges	Convex case is $O(n^3)$ Minimal error case is approximable Net flow case is hard to approx. beyond $n^{1-\epsilon}$	new formulation - -

Related Work—The problems analyzed here belong to a large class of discrete optimization problems, collectively termed Network Interdiction [4, 14, 24, 28]. They are motivated by applications such as supply chains, electronic sensing, and counter-terrorism and relate to classical optimization problems like Set Cover and Max Coverage. Our setting of budgeted interdiction with deterministic evaders on the path graph can be solved by a complicated algorithm given by [25], but we present a much simpler algorithm. Recent work on Set Cover with submodular costs [18,22] applies to some of our settings. Interdiction on paths is closely-related to the literature on box stabbing (see e.g. [27] and approximation algorithms [10]). Previous work on the Unreactive Markovian Evader (UME) interdiction problem (maximizing the expected number of Markov chain-based evaders captured with B sensors) showed that it is $\frac{e}{e-1}$-approximable by the natural greedy algorithm [15], which is the optimal approximation factor (we prove this for completeness in Proposition 2).

Other evader models have been studied such as the Most Vital Nodes Problem, in which the task is to delete a set of nodes in order to maximize the weight of the shortest path from source to destination [2, 4] or to decrease the maximum flow [16, 29], both of which could be construed as frustrating an evader's progress. Such evaders are *reactive* in the sense that the routes they take are modified based on the set of available edges or nodes. In [14], an intermediate model was studied in which the evader follows a parametrized generalization of shortest path and random walk.

Reactive evaders are closely related to the Multicut problem [3], in which the objective is to find a minimum cut that separates each of k source-sink pairs (s_i, t_i). Unreactive Full Interdiction is related to the recent work in [17]. They consider the Checkpoint Problem, in which the objective is to cut all specified paths. Unlike in Full Interdiction, the objective in the Checkpoint Problem is based on the cardinality of the intersections between the cut set and the paths.

The Bridges Problem was introduced by Glazer and Rubinstein [12] in an economics context, primarily motivated in terms of strategies for a listener to accept good arguments and reject bad arguments. In this setting, states correspond to travelers and allowing oneself to be persuaded by a statement corresponds to opening a bridge.

2 Interdiction with Oblivious Innocents

In this section we consider the Budgeted (BI) and Full Interdiction (FI) problems where the graph G (on n nodes) is restricted to several special topologies.

Definition 2

In a path graph P with nodes numbered 1 *through n, an* interval $[x, y]$ indicates the sequence of nodes numbered x through y (with x ≤ y) the interval's startpoint *and* endpoint, *respectively. Half-open intervals* $[x,y) = [x, y - 1]$ *and* $(x, y] = [x + 1, y]$ are defined similarly. For nodes x, y we write x < y to indicate that x precedes y in P. Similarly, in a tree T, an interval $[x, y]$ is the sequence of nodes lying on the path in T from x to y. A node v pierces *interval* $[x, y]$ *if* $v \in [x, y]$. *An* interval sequence is a set of intervals that can be ordered so that each interval is strictly contained by the previous one. All the intervals in a suffix sequence *share the same endpoint; all the intervals in a* prefix sequence *share the same start point.*

Theorem 1

When $c_v = 1$ for every vertex v, Full Interdiction is optimally solvable in O(n log OPT) time on paths, where OPT is the size of the optimal solution.

Proof. Consider an evader e_i with start nodes S_i and a target node t_i. We must capture evader e_i in the case of each starting point $s \in S_i$ before she reaches node t_i. Node s lies either to the left or right of t_i, assume to the left, i.e., $s < t_i$ (e.g., node 3 for the first evader e_1 in Fig. 2). Evader e_i may (probabilistically) move to the left before returning right, and so a sensor placed to the left of s may capture the evader with positive probability. For capturing e_i with probability 1, however, it is necessary and sufficient to place a sensor somewhere in the interval $[s, t_i)$ ([3, 6) for e_1 in Fig. 2).

Each starting point s of evader e_i will correspond to an interval $[s, t_i)$ or interval $(t_i, s]$, depending on the relative location of s to t_i. Each such interval must be pierced. Intervals of the former kind (with the evader approaching the target from the left) will form a sequences of suffix intervals; intervals of the latter kind (with the evader is approaching from the right) will form a sequence of prefix intervals. It suffices to consider each evader's smallest left interval and smallest right interval ([3, 6) and (6, 8] for e_1 in Fig. 2), since each such interval is contained within all others in the sequence. Finding these smallest intervals could be done in linear time by using a data structure where the intervals are indexed using both of their end points. We build an *interval graph H* by associating a node with each smallest interval (each of which can be found in time $O(\log n)$ by binary search) and placing an undirected edge for any two smallest intervals that intersect. Because the cost of piercing any interval is $c_v = 1$, and because each intersection of intervals corresponds to a clique of H, Full Interdiction is equivalent to Minimum Clique Cover on H. The latter is solvable in linear time on the interval graph (plus time for sorting) [5].

This gives a solution in $O(n \log n)$ time, however. To obtain the faster algorithm, we note that full interdiction of the smallest left and right intervals is equivalent to the problem of efficient stabbing of boxes in 1 dimension. It could be solved using the algorithm of Nielsen [27] in output-sensitive time of $O(n \log OPT)$, where of course $OPT \le n$.

A generalization is also possible, as follows.

Theorem 2

When $c_v = 1$ for every vertex v, Full Interdiction is optimally solvable in O(n^3) *time on trees.*

Proof. The $O(n^2)$ intervals are now paths in the tree, whose intersection graph (constructable in $O(n^3)$) is a *chordal graph*, on which Minimum Clique Cover can also be solved in linear time [11].

We now turn to interdiction with sensors on edges, specifically, directed edges. In this setting Full Interdiction is closely related to the Minimum Directed Multicut (MDM) problem, in which the task is to find a minimum cut that separates each of k source-sink pairs (s_i, t_i).

Proposition 1

In the edge interdiction setting, Full Interdiction is 2-approximable on trees.

Proof. When restricted to an underlying tree graph, the Full Interdiction problem is identical to Directed Multicut, which has a well-known 2-approximation [9].

Note that the 2-approximation is the best possible in general trees assuming the Unique Games Conjecture [21]. Other algorithms are also known [13, 23] including for the case of partial multi-cuts [23].

Definition 3

For a possible route r traveled by some evader, let V_r indicate the nodes visited along route r before *reaching its target*, *or the* route set of r. Let m be the number of distinct route sets among all evaders.

Note that multiple distinct routes can give rise to the same route set, and that a route set in a path graph is always an interval with an end point at the target node. We now turn to Budgeted Interdiction.

Theorem 3

Let m be the total number of different evader route sets. Budgeted Interdiction with deterministic evaders and unit costs is optimally solvable on the path graph in time O(Bnm) = $O(Bn^3)$, *where B is the budget*.

Proof. We give a dynamic programming solution in Algorithm 1. We compute an optimal solution using a table $opt\,[l, \hat{v}, b]$ that stores the optimal solution restricted to the l left-most intervals, nodes $1 \ldots \hat{v}$ and budget b. We first compute the value of node v restricted to the first l intervals, i.e., $val[l, v]$ is the sum of the weights of those intervals when the only sensor is node v. Each subproblem solution is computed in constant time: given inputs l, v, b, if v is not chosen, then the optimal solution value is the same as inputs l, $v - 1$, b; if v is chosen, then the optimal solution value is the value of choosing v in this situation, plus optimal solution on the intervals lying to the left of v, using the first $v - 1$ nodes and a budget of $b - c_v$ (or 0 if $b - c_v < 0$).

Proof of correctness is by induction: if node v is chosen, then due to the linear ordering, nodes prior to v only contribute to piercing intervals 1 through $pr(v)$. Note that correctness holds also when interval weights may be negative.

Note that in our formulation interdiction costs are integers and hence, the case of B non-integer has the same solution value as the integer case with budget [B].

Theorem 4

Budgeted Interdiction with nondeterministic evaders and unit costs is optimally solvable on the path graph, in time $O(Bn^2m) = O(Bn^4)$.

Proof. Our proof generalizes from our approach deterministic evaders. We observe that the case of nondeterministic evaders gives rise to sequences of suffix intervals and sequences of prefix intervals. For each such sequence corresponding to a single nondeterministic evader, the computation of $val[l, v]$ will be based on all the intervals in the sequence that v pierces. More precisely, let $\{[1, t), [2, t), \ldots, [s, t)\}$ be a suffix sequence for some nondeterministic evader e_i with source s and target t. For each node $v < t$ there is some probability p_v that placing a sensor at node v *suffices* for capturing e_i. Namely, p_v is 1 for any $v \in [s, t)$, while for each node $v < s$ the probability p_v can be computed based on the Markov chain of e_i, that is, just the probability that her Markov chain visits v and is computed as follows. For e_i's Markov chain $(\mathbf{a}, \mathbf{M})$, let $\mathbf{M}_{-\mathbf{v}}$ denote a transition matrix where row v has been replaced by zeros, i.e. the chain with v as a killing state. Then $p_v = \left(\mathbf{a}[\mathbf{I} - \mathbf{M}_{-\mathbf{v}}]^{-1}\right)_v$.

Algorithm 1 Budgeted Interdiction DP for Evaders on the Path Graph

sort the $O(n^2)$ intervals by right endpoint
$pr[v]$ = index of the last interval lying before node v, or 0 if none **for** every v
$val[\ell, v]$ = value of node v, restricted to intervals 1 to ℓ, **for** every v, ℓ
$opt[0, v, b] = 0$ **for** every v, b
$opt[\ell, 0, b] = 0$ **for** every ℓ, b
$opt[\ell, v, 0] = 0$ **for** every ℓ, v
for $b = 1$ to B **do**
    for $\ell = 1$ to m **do**
        for $v = 1$ to n **do**
            $opt[\ell, v, b] = \max\{opt[\ell, v-1, b],\ val[\ell, v] + opt[pr[v], v-1, \max(0, b - c_v)]\}$
        end for
    end for
end for
return $opt[m, n, B]$

For each interval in the sequence, we now define a *marginal* probability $\hat{p}_v$ as follows: $\hat{p}_1 = p_1$; $\hat{p}_v = p_v - p_{v-1}$ for $1 < v \le s$, and $\hat{p}_v = 1 - p_{s-1}$ for $s \le v < t$ By construction, the $\hat{p}_v$ values for all intervals containing a given node u will sum to exactly the probability of evader e_i reaching node u, and hence of such a sensor placement sufficing to capture evader e_i. (The values labeling the intervals in Fig. 2 are the marginal probabilities, weighted by the probability of choosing their starting points.) Marginal probabilities are assigned to prefix intervals similarly. Therefore the value of a set of sensor locations for a given instance of the problem with nondeterministic evaders is exactly the value of those locations for the resulting problem instance with interval sequences of deterministic evaders; that is, the nondeterministic problem reduces to the deterministic problem (albeit with up to a factor n more intervals).

These problems can also be solved on the cycle by reduction to path graphs.

Theorem 5

Full Interdiction is optimally solvable in $O(n^2)$ time on the cycle graph. Budgeted Interdiction with deterministic or nondeterministic evaders and budget B (assuming unit costs) is optimally solvable on the cycle graph in time $O(Bn^4)$ *or* $O(Bn^5)$, *respectively*.

Proof. For the minimization problem, we reduce to a collection of n path graph instances, corresponding to n ways to "cut" the cycle graph, as follows. For each node $v \in V$, consider placing a sensor at node v. It will pierce some set of intervals, with the effect that none of the remaining intervals to pierce include node v, yielding a path graph instance with nodes $v + 1, \ldots, n, 1, \ldots, v - 1$. Solve each resulting path graph instance in linear time, and return the cheapest solution (combined with v). The budgeted problems are solved by a similar reduction.

The process can be generalized to Full Interdiction on arbitrary graphs containing c cycles, though at a cost of $O(n^c)$: find all the cycles [19] and then explore all possible cuts. Previously, [20] has shown an $\Theta(n \log n)$ algorithm for what we termed full interdiction on paths (see therein for other algorithms). However, the algorithm in [20] requires complex special purpose data structures. As well, budgeted interdiction on tree graphs could be solved using the algorithm of [25] in $O(Bn^2)$.

3 Interdiction of Oblivious Evaders on General Graphs

Recall that in the Budgeted Interdiction Problem, the interdictor chooses B locations for the sensors, which are assumed to be invisible to the evaders, who do not change their motion. In effect, the interdictor must find a set of B nodes that collectively give the highest probability of intercepting one or more Markov chains (the evaders) - a problem named Unreactive Markovian Evader Interdiction (UME).

How hard is it to find such a set? It was shown in [15] that UME can be formulated as a Mixed-Integer Program and that UME is NP-hard when the number of evaders can be arbitrarily large. Even the simpler Budgeted Interdiction is weakly NP-hard if interdiction costs are not unitary, as can be seen with a reduction from the Knapsack problem. However, the complexity of UME is an open problem when the number of evaders is bounded and the costs are unitary. Such complexity must arise from the network topology and the stochasticity of motion, and this question is addressed in the following two theorems.

Theorem 6

Budgeted Interdiction with 2 *Markovian oblivious acyclic evaders is NP-hard.*

In the proof we use a reduction of Planar Vertex Cover - an NP-complete problem [8]. Planar Vertex Cover asks to determine whether given an undirected planar graph $G'(V', E')$ there exists a set C of $B' \geq 0$ nodes that can cover all the edges of G'. The set $C \subset V'$ is called a "vertex cover" if all the edges are incident to at least one node in C.

Proof: Given an instance of the Planar Vertex Cover problem $G'(V', E')$ with budget B' construct an instance of UME node interdiction on a derived graph $G(V, E)$ as follows in steps 1-3.

Step 1: Graph Coloring. Run the graph coloring algorithm [30] on G' and compute the color assignment: $f : V' \to \{white, red, green, black$ (abbreviated $\{w, r, g, b\}$).

Step 2: Construction of the UME graph. Assemble $G(V, E)$ as follows (Fig. 3): (a) The nodes are copied from V' and a special target node t is added: $V = V' \cup \{t\}$

(b) Include the original edges E' and for all $u \in V'$ add an edge (u, t) to t: $E = E' \cup \{(u, t) |u \in V'\}$

(c) Define $\mathbf{d}$: All nodes $u \in V$ can be completely interdicted: $d_{uv} = 1\ \forall u, v \in V$.

Step 3: Construction of the source distributions and transition matrices for the evaders. The two evaders travel on 3-node paths: from some source node through penultimate nodes to the node t, as follows.

Define two sets of source nodes: S_1 (=colors $\{w, r\}$) and S_2 (=colors $\{w, g\}$):

$$\begin{aligned} S_1 &= \{u|u \in V \setminus \{t\} \text{ and } f(u) \in \{w, r\}\} \text{ and} \\ S_2 &= \{u|u \in V \setminus \{t\} \text{ and } f(u) \in \{w, g\}\}. \end{aligned}$$

Define two sets of penultimate nodes: P_1 (=colors $\{g, b\}$) and P_2 (=colors $\{r, b\}$):

$$\begin{aligned} P_1 &= \{u|u \in V \setminus \{t\} \text{ and } f(u) \in \{g, b\}\} \text{ and} \\ P_2 &= \{u|u \in V \setminus \{t\} \text{ and } f(u) \in \{r, b\}\}. \end{aligned}$$

Finally, introduce evaders 1 and 2. For each evader i, let $\mathbf{a}^{(\mathbf{i})}$ be uniformly distributed over all nodes of class S_i. Define $\mathbf{M}^{(i)}$ so the evader takes the 3-node path discussed earlier:

1. $M_{uv}^{(i)} = \frac{1}{z_u}$ if $u \in S_i$ and $v \in P_i$ where $z_u = \|\{v|v \in P_i \text{ such that } (u, v) \in E\}\|$

2. $M_{uv}^{(i)} = 1$ if $u \in P_i$ and $v = t$

3. $M_{uv}^{(i)} = 0$ otherwise.

An illustration of the evader motion is found in Fig. 3. In the pathological case where all nodes in G' are singletons (degree= 0), arbitrarily choose any node $u \neq t$ and for evader $i \in \{1, 2\}$ let $\mathbf{a}^{(\mathbf{i})} = \delta_{uv}$ for all $v \in V$ and $\mathbf{M}^{(\mathbf{i})} = \mathbf{0}$.

Observation 1: each of the nodes in V' belongs to one of four disjoint sets corresponding to the four colors $\{w, g, r, b\}$: $w \leftrightarrow S_2 \cap S_1$, $g \leftrightarrow S_2 \cap P_1$, $r \leftrightarrow P_2 \cap S_1$ and $b \leftrightarrow P_2 \cap P_1$. These correspond to the four bit strings of length 2: 00, 01, 10 and 11; Bit $i = 1$ if and only if evader i has the node as a penultimate node.

Observation 2: no evader's source node coincides with the same evader's penultimate node: $P_1 \cap S_1 = \varnothing$ and $P_2 \cap S_2 = \varnothing$. Thus a direct jump from S_i to t has probability = 0.

Observation 3: node t could be pruned from any interdiction set without changing the expected interdiction probability because interdiction only affects outgoing evaders and node t has none.

Define the UME decision problem: Is it possible to find an interdiction set Q of size at most B so that expected interdiction probability $\langle J \rangle = 1$?

Claim: The UME decision problem with budget B set to B' is a "YES" instance if and only if a B'-cover exists for the graph G'.

Justification: The pathological case where all nodes are singletons is a UME "YES" instance for any $B \geq 0$ since the evader cannot reach the target and it is also a Planar Vertex Cover "YES" instance ($B' \geq 0$) since no edges exist.

Suppose now that a non-pathological UME instance is a "YES" instance. Since adjacent nodes in G' have different colors, *Observation 1* implies that any two adjacent nodes $u, v \in V \setminus \{t\}$ must be different by at least one bit. Thus $\exists$evader i such that one of $\{u, v\}$ is a source node (i^{th} bit = 0) while the other is a penultimate node (i^{th} bit = 1). The definitions of $\mathbf{a}^{(i)}$ and $\mathbf{M}^{(\mathbf{i})}$ imply that evader i traverses through (u, v) with positive probability. Since this is a "YES" instance with $\langle J(Q) \rangle = 1$, the interdiction set Q must contain at least one of the endpoints $\{u, v\}$ (whether or not node $t \in Q$, by *Observation 3*). Therefore the set Q is a cover for graph G'.

Conversely, if the Planar Vertex Cover decision problem is a "YES" instance then there exists a vertex cover set C. From *Observation 2* and the definition of $\mathbf{a}^{(\mathbf{i})}$ it follows that with probability 1 the evader passes on his way to the target through the edges copied from the original graph (the edges in E'). Therefore make $Q = C$ and get that $\forall i$, evader i will be interdicted with probability = 1. This a UME "YES" instance.

The proof above shows that UME is NP-hard even under fairly restrictive conditions: (1) only 2 evaders are needed, (2) the interdiction efficiencies $\mathbf{d}$ are everywhere = 1, (3) the graph is unweighted and undirected, and (4) the evaders follow 3-node paths without cycles.

We now show that budgeted interdiction is hard with a single evader as long as the evader is allowed to make cycles.

Theorem 7

Budgeted Interdiction on nodes is NP-hard with a single oblivious cyclic Markovian evader and unit interdiction cost.

Proof. We reduce from Vertex Cover (VC) to the decision problem of determining whether the interdiction probability J can be raised to a certain threshold using at most B sensors. Given a VC problem instance, i.e., a graph G on n nodes and an integer B, we construct a budgeted interdiction (BI) instance with a Markovian evader on a graph G'. The graph G'

extends graph G by adding a target node t, which is made adjacent to all other nodes. We define the evader e thus. Each node corresponds to a state of its Markov chain. All non-target nodes are equally likely to be chosen as e's start node. When at a given node v, e moves to the target t with probability 50%; otherwise, e moves to one of v's other neighbors, chosen uniformly at random.

For a particular solution, let the *profit* for a node be the probability of interdiction if the evader starts at that node. We will now show that the VC instance admits a vertex cover of size B if and only if the BI instance admits a size-B solution of profit at least $B + (n - B)/2$.

Note that an overall interdiction probability of $\frac{(n+B)}{2n}$ is the same as a total profit of $(n + B)/2 = B + (n - B)/2$ over all nodes.

First assume there is a size-B vertex cover C of G. Then an BI solution with sensors placed at all the nodes in C will have profit $B+\frac{(n-B)}{2}:1$ for each of the B nodes in C plus 1/2 for each of the remaining $n - B$ nodes, since for any node v not in C, all v's neighbors in G must be in C.

Now assume there is no size-B vertex cover, and consider a set S of B nodes, a set which must fail to cover some edge. Again for each of the B nodes in S we have profit 1. Every other node v will have profit at most 1/2, since without its own sensor, an evader starting at v goes directly to t with probability 1/2. But now consider an edge (u, v) that is left uncovered by S. The evasion probability when starting at u is greater than 1/2—at least $1/2 + 1/(4 \deg(G))$—since if e reaches node v, it now has a second chance to move to t, and so the profit of u is less than 1/2. Therefore the total profit is strictly less than $B + (n - B)/2$.

It follows from Theorem 6 above that Full Interdiction (not just Budgeted Interdiction) is NP-hard with 2 evaders. It does not remain hard when limited to a single evader, however.

Theorem 8

Full Interdiction with one evader is solvable in polynomial time.

Proof. We solve the problem by reducing to a Min Cut problem. Given a set of routes specifying the evader's behavior, we introduce a source node s pointing to all start nodes of its Markov chain. All edges that the evader has zero probability of reaching and crossing are removed from the graph G. Any unreachable nodes are also removed. Now, in order to interdict the evader before they reach t, we must delete vertices in order to separate s from t in G. It is well known that this Min Vertex Cut problem can be solved in polynomial time, by reduction to Directed Min Cut, as follows [6]. First replace any undirected edge with a pair of directed edges. Then replace each node v (other than s or t) with a pair of nodes and directed edge (v_a, v_b), where each edge directed to v is now directed to v_a and each edge directed from v is now directed from v_b. We compute a Min Cut on the resulting graph G'. If any edge is chosen that does not correspond to a node in G, we can substitute one of the edges corresponding to its two vertices (if one of these is the target, then the non-target node is chosen). The resulting modified Min Cut solution to *G'* will correspond to a Min Vertex Cut solution to *G*, and moreover to a Full Interdiction solution.

Because of the diversity of possible evaders much remains to be done in refining our understanding of UME-like problems. For example, it is not known whether the 1-evader acyclic budgeted interdiction problem has a polynomial-time solution.

We now turn to approximation algorithms for the general setting, by relating interdiction to the Set Cover and Maximum Coverage problems. It was shown in [15] that weighted Budgeted Interdiction with any number of Markovian evaders is 1-1/e-approximable. We now tighten this claim.

Proposition 2

The Budgeted Interdiction problem in NP-hard to approximate within factor $1 - 1/e - \varepsilon$ *for any* $\varepsilon > 0$.

Proof. We reduce from Maximum Coverage, which has the stated hardness property [7].

Given is a family of subsets S_i of a ground set $U = \{e_1, \ldots, e_n$. The task is to choose k subsets whose union is of maximum cardinality. For each set S_i we introduce a corresponding node v_i. For each element e_j we introduce a corresponding evader whose Markov chain takes it deterministically (in some arbitrary order) through all the nodes corresponding to sets containing e_j and thence to a special target node. Then a selection of sets covering evader paths is equivalent to a selection of sets covering elements, with exactly the same solution value.

Identifying nodes and route sets (Def. 3) with elements and sets in the Hitting Set problem yields a reversible reduction, and hence the following immediately results:

Corollary 1

Full Interdiction is hard to approximate with factor $(1 - \varepsilon)$ ln m *for any* $\varepsilon > 0$, *assuming NP* $\subseteq$ *DTIME*$(m^{O(\log \log m)})$, where where n is the number of nodes and m is the number of route sets. It can be approximated with factor H_m in time polynomial in n + m.

Proof. We reduce from Set Cover, as in [15], creating a node for each set and a route set (with a corresponding deterministic evader) for each element. The rest follows from Feige [7].

4 Reactive Innocents and the Bridges Problem

Recall that in the bridges problem each of the users p of the graph has a set $\sigma(p)$ of bridges, representing his paths to the target.

Consider now the the min-error $FP + FN$ setting, i.e., the problem of finding a policy that minimizes the weighted sum of successful bad users and blocked good users. We show below that a geometric or "convex" version of the min-error problem is optimally solvable. Since the two objective functions differ only by a constant and a negation ($TN - FN = (W_N$

$- FP) - FN$, where W_N indicates the total value of all goods), the same holds for the net flow problem.

4.1 Convex bridge sets

Definition 4—*An instance of the Bridges Problem is* convex if the bridges can be ordered so that if two bridges x and y are accessible to a person p then any bridge z with x < z < y is accessible to p as well.

The problem example shown in Figure 1 in the introduction is convex. We assume that the indices of travelers are sorted in order of their positions from left to right and the bridge indices are sorted in order of their rightmost accessing person. This setting can be solved by mapping it to Budgeted Interdiction on the path graph and adapting Algorithm 1.

Corollary 2—*The convex Bridges Problem is solvable in time* $O(n|N|+n|D|) = O(n^3)$.

Proof. Given a Bridges Problem instance (say in the min-error formulation), we introduce a Budgeted Interdiction instance (with budget arbitrarily large) as follows. Each of the bridges is identified with a node on the path graph. For each traveler p we define an evader p on the interval I_p, where I_p are all bridges available to p. This produces $|N| + |D| \le n(n - 1)$ distinct intervals. The weight of evader p is set to negation of the traveler's cost: $w_p = -\hat{w}_p$.

The formulations are now equivalent: in the Bridges Problem, a traveler succeeds if and only if one or more of her bridges is open; in BI, an evader is interdicted if and only if one or more of the nodes in her interval is interdicted by a sensor.

Then we pass the instance to an adaptation of Algorithm 1: we remove the budget dimension from the dynamic programming table and also remove the outer loop iterating over budget values, saving a factor of $O(B)$ in running time. The resulting algorithm computes an optimal interdiction solution. (Recall that Algorithm 1 supports intervals with weights both positive and negative.) Given this solution, we then solve the Bridges Problem by opening a bridge if and only if the corresponding node has a sensor placed at it.

4.2 The general min-error *FP* + *FN* Bridges Problem

In this section we develop an approximation algorithm for the min error setting. The min-error problem is precisely the Positive-Negative Partial Set Cover Problem (PNPSCP) [26], which, as a generalization of Red-Blue Set Cover, is strongly inapproximable. In particular, PNPSCP is hard to approximate with factor $\Omega(2^{\log^{1-\varepsilon} m})$ (where m is the number of sets) unless $NP \subseteq DTIME(m^{\mathrm{polylog}(m)})$ though approximable with factor $2\sqrt{(m+\pi)\log \pi}$, where π is the number of goods.

Inspired by Glazer and Rubinstein, we will call an *m-claw* an object c consisting of a good g_c and minimal set of bads B_c such that for each bridge $s \in \sigma(g_c)$, s is also in $\sigma(b_i)$ for some bad $b_i \in B_c$, which means that in any consistent solution, either g_c must fail or at least one b_i must succeed. Glazer and Rubinstein show that this is also a sufficient condition for being a valid solution, and hence obtain a Set Cover problem: for each m-claw, choose a person to err on, with minimum total error cost over all m-claws. Unfortunately, this instance in general has exponentially many constraints (since for each good g with bridge set $\sigma(g)$, each s of whose bridges admit some number $D(s)$ of bads, there will be $|C| = \pi_{s\in\sigma g}D(s)$ many m-claws), and so the standard log $|C|$ approximability of set cover becomes trivially weak. We therefore modify the definition of m-claw slightly, and introduce a claw, as follows.

Definition 5—*A* claw is an object c consisting of a good g_c and, for each bridge $s_i \in \sigma(g_c)$ the set $b \in \sigma^{-1}(s_i)$ *of all the bads who can use bridge* s_i.

Each claw c therefore imposes the following constraint: in any valid solution, either g_c must fail or all the bads in $\sigma^{-1}(s_i)$ for some $s_i \in \sigma(g_c)$ must succeed. Given c, let a *kill move* be the action of killing g_c (i.e., blocking all of her bridges); let an *open bridge move* be the action of opening some bridge s_i. Now we can interpret this problem as an instance of Submodular Cost Set Cover [18, 22] in which the elements are claws and there are two kinds of sets. For each possible kill move m_g, introduce a set $M_g = \{g\}$; for each possible open bridge move m_i, introduce a set M_i consisting of all the claws that opening bridge i would satisfy. There are N elements (claws) and $N + m$ sets (moves).

Theorem 9—*The general FP + FN Bridges Problem is* $(\max_{g\in N} |\sigma(g)| + 1)$*-approximable*.

Proof. First we claim that the cost of a set of moves is submodular. Indeed, the cost of each kill move is simply the additive cost of the specified good failing; the marginal cost of an open bridge move is monotonically decreasing since it is based on the number of *additional* bads that opening the bridge then allows to succeed. Second we claim that the value of the total error of the Bridge solution returned is at most the cost of the moves chosen. Indeed, first, the only time bridges are opened is during bridge moves, and so the total cost of bads succeeding is at most the cost of the open bridge moves; second, when bridges are closed at the end, all constraints have been handled, and so the failures of all goods have already been “paid for”, in the cost of the kill moves. Therefore the algorithms of [18,22] apply, which provide a solution with approximation factor f, which is the maximum number of sets that any element appears in. In the constructed set cover instance, f translates into 1 plus $\max_{g\in N} |\sigma(g)|$.

In computational experiments, we were able to solve fairly large instances of the general *FP + FN* using a Mixed-Integer Program. In those experiments, we generated problem instances with 10000 bridges and 5000 evaders, 5000 innocents, and with a mean of 20 bridges available to each of them assigned at random. Weights were selected uniformly at random from [0, 1]. To our surprise, IBM’s CPLEX 12.2 was able to solve those instances consistently in less than 10 seconds on a 1GHz dual core Intel i5 processor.

5 Hardness results for the Bridges Problem

The following two results are approximation-preserving reductions from the Maximum Independent Set (MIS) problem. MIS is the problem of finding a maximum cardinality set of vertices such that no pair of elements is connected by an undirected edge. MIS is hard to approximate with factor $n^{1-\varepsilon}$ (where $|V| = n$) for any $\varepsilon > 0$ [31]. A MIS instance consists of a graph $G = (V, E)$ and a positive integer k.

Proposition 3

The Bridges Problem variant in which the goal is to maximize TN subject to a bound on FN is NP-hard to approximate with factor $n^{1-\varepsilon}$.

Proof. In our reduction, each vertex v becomes a bridge s_v and a bad b_v who can cross only s_v. Each edge (u, v) becomes $k + 1$ goods who can cross bridges s_u and s_v. The bound on FN is set to k, which prevents any two goods connected by an edge from both failing.

NP-hardness of optimally solving the min-error setting ($FP + FN$) follows from the hardness of the net-flow setting: maximizing $TN - FN$ is the same as minimizing $FN - TN = FN - (W_N - FP) = FP + FN - W_N$. The hardness of approximation properties, however, are not the same. Whereas the min-error setting can be usefully approximated (Thm. 9 above), the net-flow setting can not:

Proposition 4

The net-flow TN − FN setting of the Bridges Problem is NP-hard to approximate with factor $n^{1-\varepsilon}$.

Proof. In our reduction, each vertex becomes a bridge and (usually) some bad people, and each edge becomes a good person. All the people introduced have value 1 or −1. Specifically, for each vertex $v \in V$, we introduce a bridge s_v and $deg(v) - 1$ bads, whose only accessible bridge is s_v itself. (If $deg(v) = 0$, we similarly introduce one good.) For each edge $e = (u, v) \in E$, we introduce a good p_e, whose accessible bridges are u and v.

We now claim that the MIS instance has a solution of value at least k if and only if the Bridges Problem instance does. First, assume there is an independent set S of size k. For each vertex $v \in S$, we open the corresponding bridge. Each bridge s_v with $deg(v) = 0$ has one good and no bads, for a net value of 1. For each bridge s_v with $deg(v) > 0$, there are $deg(v)$ goods who can cross it (and possibly others) and $deg(v) - 1$ bads who can cross only it. Since no two vertices in S are adjacent, though, for each open bridge the goods who can cross it can cross no other open bridges. Therefore for each open bridge s_v, all its $deg(v)$ goods will use it, which means that bridge contributes exactly $deg(v) - (deg(v) - 1) = 1$ to the solution value, for a total of k.

Conversely, assume there is a bridges solution of value at least k. Observe that no open bridge can contribute value greater than 1, since at most $deg(v)$ goods use it but necessarily all its $deg(v)-1$ bads will do so. Therefore a solution of value k will involve opening at least k bridges. If any bridge can be closed without decreasing the solution quality, do so, repeatedly, until there is no longer any such bridge. At that point, the solution will consist of k open bridges, each of value k. But again by the previous argument, in order for two bridges each to contribute value 1, the corresponding vertices must be independent. Thus the k vertices corresponding to the open bridges form an independent set.

Corollary 3

The maximum-probability net-flow TN−FN setting of the Bridges Problem of [12] is as hard to approximate as the (integral) net-flow TN − FN setting.

Proof. Consider the Bridges Problem instance produced in Proposition 4, but now allow fractional bridge openings and take the max-probability objective. An integral solution is in particular a valid fractional solution, and so the forward direction of the if and only if goes through unchanged. Now assume there is a max-prob fractional solution of value at least k. Suppose some bridge s is open with probability p, $0 < p < 1$. If more bads are using s than goods, then closing s will only improve the solution, so assume otherwise. In this case, assume that γ goods are using s and β bads are, with $\gamma \geq \beta$. Then fully opening the bridge will increase the value of *at least* γ goods by amount $(1 - p)$—any other goods that had chosen other bridges that were also open with probability p will now shift to this bridge—and will increase the value of β bads by the same amount $(1 - p)$, for a total change to the bridge's net flow of at least $(1 - p)\gamma-(1 - p)\beta$, which is non-negative. Therefore we can convert the fractional solution into an integral solution of value still at least k. But then by the previous argument we can use the solution to obtain an independent set of size k.

Corollary 4

The maximum-probability min-error FP + FN setting of the Bridges Problem of [12] is as hard to approximate as the (integral) min-error FP + *FN setting.*

Proof. The proof is similar to that of Corollary 3.

6 Discussion

In this paper, we gave positive results for network interdiction problems in a number of settings, as well as a number of negative results. An interesting open problem is to give unconditional approximation results for Full Interdiction on general graphs, which is a generalization of the Minimum Directed Multicut problem. Unlike in Directed Multicut, in Full Interdiction there is for each evader an arbitrary subgraph G_i of G consisting of the edges that evader e_i is permitted to traverse: G_i are all edges visited by e_i with positive probability. Therefore the following hardness result is inherited from Directed Multicut [3]:

Proposition 5

Full Interdiction is hard to approximate (even in DAGs) with factor $2^{\Omega(\log^{1-\varepsilon} n)}$ *for any* $\varepsilon >$ 0 *unless NP* $\subseteq$ *ZPP.*

As with the Directed Multicut problem, a trivial k-approximation can be obtained by combining the results of k separate Min Cut solutions, corresponding to each evader. Obtaining a nontrivial approximation for this problem on general graphs appears challenging, however. The best-known approximation for Directed Multicut is $\tilde{O}\left(n^{11/23}\right)$ [1].

NIH-PA Author Manuscript

NIH-PA Author Manuscript

These results depend on *disjointness arguments* which limit the number of successive Min Cut problems that need to be solved (and hence bound the final solution cost) by arguing that each Min Cut solution will permanently separate a given edge e from a certain number of nodes or edges in the graph, thus bounding the number of cuts that e can be involved with. In our problem, however, s_i does not have to be separated from t_i in the graph G, but only within the subgraph of G corresponding to evader e_i. The effect of this relaxed separation requirement is that once e is separated from a given node v in the graph *for evader* e_i, v may still be reachable from e *for another evader* j. Obtaining nontrivial approximation results for this problem therefore may require different techniques.

A second clear open problem concerns the Bridges Problem. Iwata and Nagano [18] give a hardness of approximation result for factor $o(n/ \ln^2 n)$ for set cover with monotone submodular functions and an approximation algorithm whose factor is the frequency f (which can be $O(n)$ in general). We formulate the problem of minimizing $FP + FN$ as a monotone submodular set cover problem, which allows us to apply the f-approximation algorithm, but the hardness result we obtain for our problem is only for factor $o(\ln n)$. It remains to close this gap.

Acknowledgments

We thank Amotz Bar-Noy and Rohit Parikh for useful discussions and two anonymous reviewers for valuable criticism. This work was funded by the Department of Energy at the Los Alamos National Laboratory through the LDRD program, and by the Defense Threat Reduction Agency. AG would like to thank Robert Kleinberg for fascinating lectures, and Feng Pan and Aric Hagberg for the support.

Some of the above results appeared as an extended abstract in the Proceedings of the 7th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS) 2011. We have added proofs to many of the theorems and Theorem 6, which establishes a connection from Network Interdiction to graph coloring.

References

1. Agarwal A, Alon N, Charikar M. Improved approximation for directed cut problems. STOC. 2007:671–680.
2. Bar-Noy, A.; Khuller, S.; Schieber, B. Tech. Rep. University of Maryland; College Park, MD, USA: 1995. The complexity of finding most vital arcs and nodes.
3. Chuzhoy J, Khanna S. Polynomial flow-cut gaps and hardness of directed cut problems. J. ACM. 2009; 56(no. 2) Article No. 7.
4. Corley HW, Sha DY. Most vital links and nodes in weighted networks. Operations Research Letters. Sep; 1982 1(no. 4):157–160.
5. Even G, Levi R, Rawitz D, Schieber B, Shahar S, Sviridenko M. Algorithms for capacitated rectangle stabbing and lot sizing with joint set-up costs. ACM Transactions on Algorithms. 2008; 4(no. 3) Article No. 34.
6. Even, S.; Even, G. Graph Algorithms. Cambridge University Press; 2011.
7. Feige U. A threshold of ln for approximating set cover. J. ACM. 1998; 45(no. 4):634–652.
8. Garey, MR.; Johnson, DS.; Stockmeyer, L. STOC ’74. ACM; New York, NY, USA: 1974. Some simplified NP-complete problems; p. 47-63.
9. Garg N, Vazirani V, Yannakakis M. Primal-dual approximation algorithms for integral flow and multicut in trees. Algorithmica. 1997; 18(no. 1):3–20.
10. Gaur D, Ibaraki T, Krishnamurti R. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. Journal of Algorithms. 2002; 43(no. 1): 138–152.

11. Gavril F. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM J. Computing. 1972; 1(no. 2):180–187.
12. Glazer K, Rubinstein A. A study in the pragmatics of persuasion: A game theoretical approach. Theoretical Economics. 2006; 1:395–410.
13. Golovin D, Nagarajan V, Singh M. Approximating the k-multicut problem. Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm. ACM. 2006:621–630.
14. Gutfraind A, Hagberg A, Izraelevitz D, Pan F. Wood RK, Dell RF. Interdiction of a Markovian Evader. Proceedings of the 12th INFORMS Computing Society Conference on OR, Computing, and Homeland Defense. Jan.2011 :3–15. IN-FORMS.
15. Gutfraind, A.; Hagberg, A.; Pan, F. Optimal interdiction of unreactive Markovian evaders. In: Hooker, J.; Hoeve, W.-J. van, editors. Proc. CPAIOR, ser. Lecture Notes in Computer Science. Vol. 5547. Springer; May. 2009 p. 102-116.
16. Corley JHW, Chang H. Finding the n most vital nodes in a flow network. Management Science. Nov; 1974 21(no. 3):362–364.
17. Hajiaghayi M, Khandekar R, Kortsarz G, Mestre J. The checkpoint problem. Theoretical Computer Science. 2012; 452:88–99.
18. Iwata S, Nagano K. Submodular function minimization under covering constraints. FOCS. 2009:671–680.
19. Johnson DB. Finding all the elementary circuits of a directed graph. SIAM Journal on Computing. 1975; 4(no. 1):77–84.
20. Katz M, Nielsen F, Segal M. Maintenance of a piercing set for intervals with applications. Algorithmica. 2003; 36(no. 1):59–73.
21. Khot S, Regev O. Vertex cover might be hard to approximate to within $2 - \varepsilon$. J. Comput. Syst. Sci. 2008; 74(no. 3):335–349.
22. Koufogiannakis C, Young NE. Greedy Δ-approximation algorithm for covering with arbitrary constraints and submodular cost. ICALP (1). 2009:634–652.
23. Levin A, Segev D. Partial multicuts in trees. Theoretical computer science. 2006; 369(no. 1):384–395.
24. McMasters AW, Mustin TM. Optimal interdiction of a supply network. Naval Research Logistics Quarterly. 1970; 17(no. 3):261–268.
25. Megiddo N, Zemel E, Hakimi SL. The maximum coverage location problem. SIAM Journal on Algebraic and Discrete Methods. 1983; 4(no. 2):253–261.
26. Miettinen P. On the positive-negative partial set cover problem. Inf. Process. Lett. 2008; 108(no. 4):219–221.
27. Nielsen F. Fast stabbing of boxes in high dimensions. Theoretical Computer Science. 2000; 246(no. 1):53–72.
28. Pan, F.; Charlton, WS.; Morton, DP. Interdicting smuggled nuclear material. In: Woodruff, D., editor. Network Interdiction and Stochastic Integer Programming. Kluwer Academic Publishers; Boston: 2003. p. 1-19.
29. Ratliff HD, Sicilia GT, Lubore SH. Finding the n most vital links in flow networks. Management Science. Jan; 1975 21(no. 5):531–539.
30. Robertson, N.; Sanders, DP.; Seymour, P.; Thomas, R. STOC ’96. ACM; New York, NY, USA: 1996. Efficiently four-coloring planar graphs; p. 571-575.
31. Zuckerman D. Linear degree extractors and the inapproximability of max clique and chromatic number. Theory of Computing. 2007; 3(no. 1):103–128.

NIH-PA Author Manuscript
NIH-PA Author Manuscript

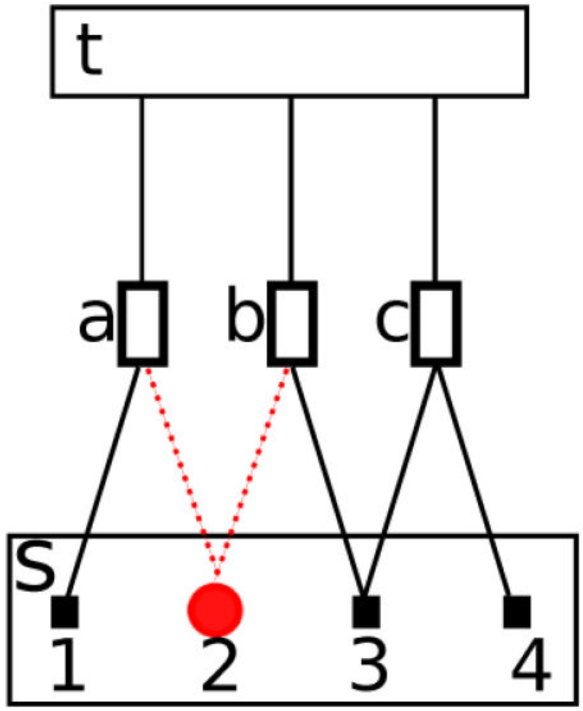

Fig. 1.
A bridges problem instance represented as network interdiction with three intermediate nodes corresponding to bridges. An innocent begins at node 2 and evaders begin at nodes 1,3,4.

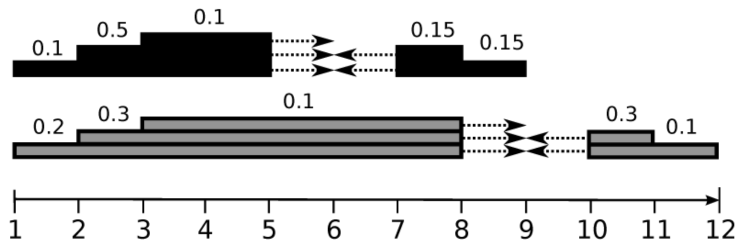

Fig. 2.

An instance of Network Interdiction with two stochastic evaders on the graph P_{12}. One evader is traveling from nodes 3 and 8 to 6, and one is traveling from nodes 3 and 11 to 9. Because of their stochastic motion, the evaders could be partially interdicted at nodes such as 1 or 12 that do not lie on the shortest paths to their targets.

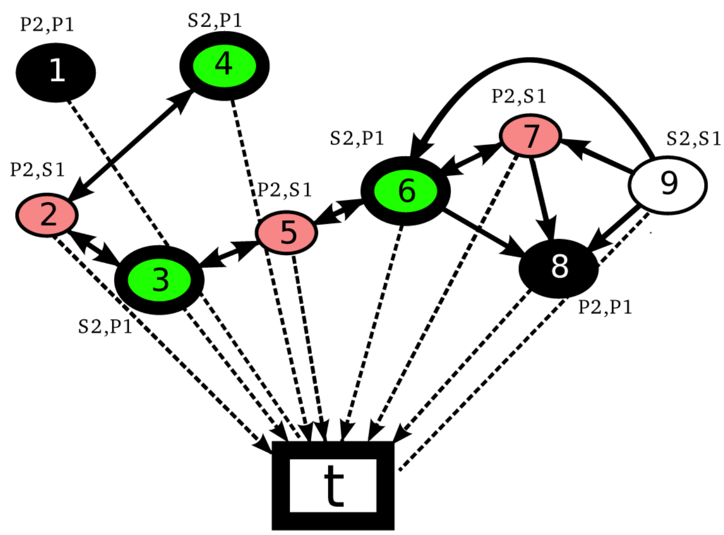

Fig. 3.

The graphs *G'*, *G* showing the evaders and the classes of nodes. The original Vertex Cover instance is on *G'* (drawn with elliptical nodes and solid edges). *G* is created by adding node *t* (rectangle) and the edges to *t* (dashed lines). On *G' White* indicates class $S_2 \cap S_1$, *green* (large ellipses) indicates class $S_2 \cap P_1$, *red* (small ellipses) indicates class $P_2 \cap S_1$ and *black* indicates class $P_2 \cap P_1$. Evader motion is marked with arrows. For example, the bi-directional arrow between nodes 3 and 5 marks that it is passed in both direction by evaders: evader 1 moves along $5 \to 3 \to t$ and evader 2 moves along $3 \to 5 \to t$.

NIH-PA Author Manuscript
NIH-PA Author Manuscript