OPTIMIZING PLACEMENT OF STATIONARY MONITORS *

ANDREW ROMICH †, GUANGHUI LAN ‡, AND J. COLE SMITH §

Abstract. We examine the problem of placing stationary monitors in a continuous space, with the goal of minimizing an adversary's maximum probability of traversing an origin-destination route without being detected. The problem arises, for instance, in defending against the transport of illicit material through some area of interest. In particular, we consider the deployment of monitors whose probability of detecting an intruder is a function of the distance between the monitor and the intruder. Under the assumption that the detection probabilities are mutually independent, we construct a two-stage mixed-integer nonlinear programming formulation for the problem. We first provide an algorithm that optimally locates monitors in a continuous space. Then, we examine this problem for the case in monitor locations are restricted to two different discretized subsets of continuous space. Our analysis provides optimization algorithms for each case, and derives bounds on the worst-case optimality gap between the restrictions and the initial (continuous-space) problem. Empirically, we show that we can obtain discretized solutions whose worst-case and actual optimality gaps are well within practical limits.

Keywords: shortest path network interdiction, network optimization, nonlinear programming

1. Introduction.

Detection of an approaching adversary is crucial to thwarting an attack. As an example of an attack-vulnerable environment, we consider United States (U.S.) seaports. Almost 11 million containers arrive in U.S. ports each year, transporting 95 percent of U.S. non-North American trade by weight and 75 percent by value [1, 2]. Because ports play such a key role, they also serve as a prime target for a terrorist attack. Simulations show that a terrorist attack forcing every U.S. port to shut down could result in manufacturing and production losses of $58 billion [2, 3]. In response to this threat, we examine in this paper the placement of stationary monitors in an area of interest so as to minimize the adversary's maximum probability of evasion. (The terms "monitors" and "sensors" can be used interchangeably, although we employ the former term in our technical description.)

In an effort to optimize placement of these monitors, we develop a two-stage mixed-integer nonlinear programming (MINLP) formulation of the problem. The first stage deploys the monitors, while the second stage models the adversary's choice of an optimal path through the area of interest. The model falls under the category of Stackelberg leader-follower games, in which the follower (adversary) is assumed to correctly identify the monitor locations in the network [18].

Optimizing sensor placement with the goal of minimizing an adversary's maximum evasion probability is related to the problem of interdicting arcs in a network to maximize an adversary's shortest path (see [17] for a survey of network interdiction). In this context, the work of Israeli and Wood [13] is pivotal, as it provides a one-stage mixed-integer programming formulation of the problem that models the basic two-stage shortest path network interdiction model. Brown et al. [10] create a framework for identifying a "near-optimal" defense

*The authors gratefully acknowledge the support of the *Defense Advanced Threat Reduction Agency* under grant # HDTRA-10-01-0050, and the *National Science Foundation* under grants #CMMI-1000347 and #CMMI-1100765.

† Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, (email: aromich222@gmail.com).

‡Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, (email: glan@ise.ufl.edu).

§Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, (email: cole@ise.ufl.edu).

strategy based on a given budget via Stackelberg interdiction models. Cormican et al. [11] study a stochastic network interdiction model, which seeks to minimize the expected maximum flow in which the capacity of an arc after interdiction is uncertain. Bienstock and Verma [9] and Pinar et al. [16] explore the vulnerability of electrical grid networks to the simultaneous failure of a key subset of links. Additionally, studies in [4, 20, 21] analyze the effectiveness of employing defensive "vehicles" to interdict an adversary who attempts to transport a radiological/nuclear weapon into a city.

Variants of the problem we consider have been examined under assumptions that certain characteristics of the network are unknown [12], probabilities of evasion are parameters [15], and information between the adversary and interdictor are asymmetric [5]. Wilhelm and Gokce [22] focus on solving this problem as it relates to the port environment, where the probability of detecting an adversary is known given a discrete set of sensor combinations, sensor locations, surveillance points, and environmental conditions.

The variation of the traditional shortest path network interdiction model that we consider examines a continuous feasible region in which the monitors can be placed. However, the adversary's path is limited to movements on a discretized space. Evasion probability is based on the quantity and location of monitors and the number of discrete adversary moves, with an independent probability of detecting each move. Additionally, we operate under the assumption that the adversary knows the monitor locations so as to account for a "worst-case" scenario. This is in contrast to the work in [19] where both the leader, who chooses monitor locations, and follower, who chooses a path through the network, make decisions simultaneously, resulting in a mixed-strategy solution. Another opposing examination of the problem is from a robust optimization perspective, where the adversary, rather than knowing exactly the monitor locations, must account for any possible set of monitor locations when determining a path [7, 8]. This type of model, however, does not allow the extent of recourse on the adversary's decision that is allowed in our model.

The contributions of this paper are as follows. We present the first algorithm that can converge to an optimal solution for the foregoing interdiction model. Through convergence analysis of this algorithm, we establish a bound on the maximum number of mixed-integer linear programs solved to obtain a particular solution accuracy. Furthermore, we derive bounds on the maximum difference between the optimal objective function values of a model with monitors placed in particular discrete spaces and that of a continuous-space model. We also discuss how our analysis allows one to determine whether it is most appropriate to use the exact algorithm or one of the restriction techniques mentioned above.

The remainder of this paper is organized as follows. In § 2 we present a two-stage MINLP formulation of the problem in which monitors can be deployed anywhere in the continuous space. We present in § 3 an algorithm to solve the continuous space problem, and prove that this algorithm can converge to an optimal solution, and provide an estimate on its rate of convergence. Section 4 provides a restricted formulation that employs a partial discretization, resulting in provably near-optimal solutions. Section 5 discusses a full discretization of the monitors' feasible region, along with an analysis of the impact of full discretization on the (worst-case) optimality gap. Section 6 presents computational results of the proposed methods on randomly generated test instances, and we conclude the paper in § 7.

2. Continuous-Space Formulation. We first provide some preliminary assumptions and model notation in § 2.1, and next formulate the monitor placement problem as a MINLP in § 2.2. We then develop a relaxation of this MINLP that can be solved more efficiently in § 2.3.

2.1. Preliminaries. We begin by considering a rectangular area on which our monitors are deployed. This area is divided into n_c columns, indexed by $I = \{1, \ldots, n_c\}$, and n_r rows, indexed by $J = \{1, \ldots, n_r\}$, with $n_c \geq 2$ and $n_r \geq 2$. The first column is located where the x-axis coordinate equals zero over the area, the last column is located where the x-axis coordinate equals L, and all columns are evenly spaced apart. The n_r rows are likewise evenly spaced from 0 to H. We create $n_c n_r$ nodes, one at each coordinate (i, j) of the grid, $\forall i = 1, \ldots, n_c$ and $j = 1, \ldots, n_r$. It is also useful to define a function $\Upsilon : \mathbb{R}^2 \to \mathbb{R}^2$ that maps a vector (k, l) over the (continuous) grid space, to its corresponding vector in the original space $[(0, 0), (L, H)]$, i.e.,

$$\Upsilon(k, l) = \left(\frac{kL}{n_c - 1}, \frac{lH}{n_r - 1} \right).$$

An adversary can enter the area at any node $(1, j)$ in the first column. The adversary will then move from column i to column $i + 1$, for each $i = 1, \ldots, n_c - 1$. A valid move across an arc, indexed by ijk, is thus from location (i, j), $i \in I \backslash \{n_c\}$, $j \in J$, to location $(i + 1, k)$, where $k \in J$. The set of monitors is indexed by $S = \{1, \ldots, |S|\}$. Figure 2.1 illustrates the foregoing model environment.

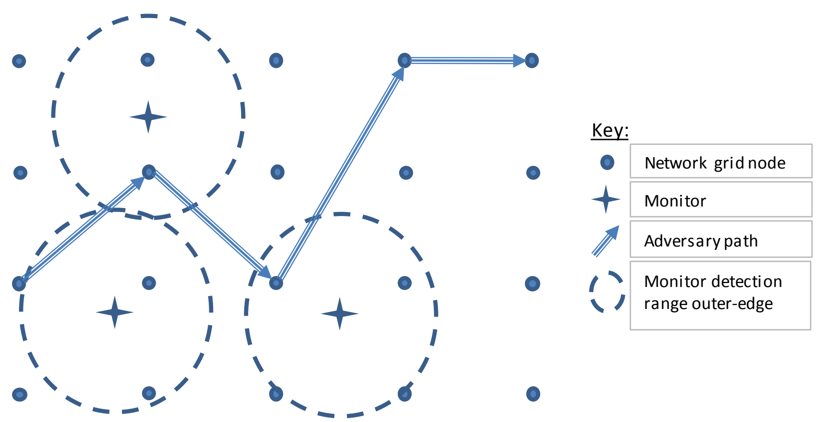

FIG. 2.1. *A feasible adversary path and monitor locations with $n_c = 5$, $n_r = 4$, and $|S| = 3$.*

The adversary is assumed to have full knowledge of the monitor placements and will progress from column i to $i + 1$, $i \in I \backslash \{n_c\}$, in its path in order to take a direct route to the target. (This extension in which an adversary can move along a path that visits multiple nodes in each column can also be handled in our analysis.) Furthermore, we assume that the probability of evading $s \in S$ along a move from column $i \in I \backslash \{n_c\}$ to $i + 1$ is independent of all previous moves.

We now discuss our modeling assumptions that dictate the probability that the adversary is detected on each arc. First, for every arc ijk, let $v_{ijk} \in (0, 1]$ be the probability that the adversary escapes detection on arc ijk in the network if there are no monitors that observe this arc. We refer to these v-parameters as "arc factors," which account for existing detection resources native to the port. Next, define the *distance from a monitor to arc ijk* as the rectilinear distance (ℓ_1-norm) from the monitor to the midpoint location, $\mathbf{c}_{ijk}$, of the arc. We model the probability of evading a monitor during a move across an arc as a function of the distance from the monitor to the arc. In particular, if this distance exceeds some maximum detection radius, R, then the probability of evading that monitor is 1. Furthermore, we define $\delta_p \in (0,1)$ to be a monitor-dampening parameter, which denotes the minimum probability of evading a monitor (even if that monitor were directly on an arc midpoint traversed by the adversary). The adversary therefore evades detection in traversing arc ijk with probability

$$v_{ijk} \prod_{s\in S} \min\left\{ \frac{\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\|}{R}(1-\delta_p)+\delta_p, 1\right\}.$$

2.2. MINLP Formulation. Define variables $f_{ijk} = 1$, $\forall i \in I\backslash\{n_c\}$, $j,\ k \in J$, if the adversary moves on arc ijk, and $f_{ijk} = 0$ otherwise. The following is a two-stage MINLP formulation of the problem, where $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|S|} : (1,1) \le \mathbf{x}_s \le (n_c, n_r), \forall s \in S\}$.

$$\min_{\mathbf{x}\in X} \max \prod_{i\in I\backslash\{n_c\}} \left(\sum_{j\in J}\sum_{k\in J} \left(v_{ijk} f_{ijk} \prod_{s\in S} \min\left\{ \frac{\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\|}{R}(1-\delta_p)+\delta_p, 1\right\}\right)\right) \tag{2.1a}$$

$$\text{s.t.} \quad \sum_{j\in J}\sum_{k\in J} f_{1jk} = 1 \tag{2.1b}$$

$$\sum_{j\in J} (-f_{ijk} + f_{i+1,k,j}) = 0 \quad \forall i \in I\backslash\{n_c\},\ k \in J \tag{2.1c}$$

$$f_{ijk} \in \{0,1\} \quad \forall i \in I\backslash\{n_c\},\ j,\ k \in J. \tag{2.1d}$$

Constraints (2.1b) and (2.1c) are the adversary's flow balance constraints for every grid point except for those in column n_c, which are implied by (2.1b) and (2.1c). Constraints (2.1d) restrict flow on all arcs to be binary. In the remainder of this paper, we express constraints (2.1b) and (2.1c) as $\mathbf{Af} = \mathbf{b}$, and define $\Lambda = \{\mathbf{f} \in \{0,1\}^{n_r^2(n_c-1)} : \mathbf{Af} = \mathbf{b}\}$ as the set of all feasible flows.

The second-stage (adversary's) problem is the following, given fixed values $\mathbf{x}_1, \ldots, \mathbf{x}_{|S|}$:

$$\max_{\mathbf{f}\in\Lambda} \prod_{i\in I\backslash\{n_c\}} \left(\sum_{j\in J}\sum_{k\in J} \left(v_{ijk} f_{ijk} \prod_{s\in S} \min\left\{ \frac{\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\|}{R}(1-\delta_p)+\delta_p, 1\right\}\right)\right). \tag{2.2}$$

The objective function is nonconvex because its Hessian is indefinite. However, recalling that the monitor-dampening parameter δ_p and all arc factors v_{ijk} are positive, and that $\sum_{j\in J}\sum_{k\in J} f_{ijk} = 1$, $\forall i \in I\backslash\{n_c\}$, each term

$$\sum_{j\in J}\sum_{k\in J} \left(v_{ijk} f_{ijk} \prod_{s\in S} \min\left\{ \frac{\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\|}{R}(1-\delta_p)+\delta_p, 1\right\}\right)$$

is positive. Because the natural logarithm function is nondecreasing, we can instead take the natural logarithm of the function in (2.2), yielding the following formulation:

$$\max_{\mathbf{f}\in\Lambda} \sum_{i\in I\backslash\{n_c\}} \ln \left(\sum_{j\in J}\sum_{k\in J} \left(v_{ijk} f_{ijk} \prod_{s\in S} \min\left\{ \frac{\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\|}{R}(1-\delta_p)+\delta_p, 1\right\}\right)\right). \tag{2.3}$$

Because all $\mathbf{f}$-variables are restricted to be binary-valued, (2.3) is equivalent to

$$\max_{\mathbf{f}\in\Lambda} \sum_{i\in I\backslash\{n_c\}} \sum_{j\in J}\sum_{k\in J}\sum_{s\in S} \ln\left(\min\left\{ \frac{||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})||}{R}(1-\delta_p)+\delta_p, 1\right\} v_{ijk}\right) f_{ijk}. \tag{2.4}$$

Using the fact that $\mathbf{A}$ is totally unimodular and $\mathbf{b}$ is a binary vector, we can replace binariness restrictions on $\mathbf{f} \in \Lambda$ with nonnegativity restrictions [6]. Thus, the overall two-stage formulation is

$$\min_{\mathbf{x}\in X} \max_{\mathbf{f}\in\Lambda} \left\{ G(\mathbf{x},\mathbf{f}) = \sum_{i\in I\setminus\{n_c\}} \sum_{j\in J} \sum_{k\in J} \sum_{s\in S} \ln\left(\min\left\{ \frac{||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})||}{R}(1-\delta_p) + \delta_p, 1 \right\} v_{ijk} \right) f_{ijk} \right\}. \tag{2.5}$$

The second-stage problem can be rewritten for simplicity as

$$\max \quad \mathbf{q}(\mathbf{x})^T\mathbf{f} \tag{2.6a}$$

$$\text{s.t.} \quad \mathbf{f} \in \Lambda, \tag{2.6b}$$

where $\mathbf{x} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_{|S|}\right) \in \mathbb{R}^{2\text{x}|S|}$, the coefficient of f_{ijk} is given by

$$\mathbf{q}(\mathbf{x})_{ijk} = \sum_{s\in S} \psi\left(\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\|\right) + |S| \ln\left(v_{ijk}\right), \ \forall i \in I\setminus\{n_c\},\ j,\ k \in J, \tag{2.7}$$

where

$$\psi(d) = \ln\left(\min\left\{ \frac{d}{R}(1-\delta_p) + \delta_p, 1 \right\} \right). \tag{2.8}$$

Associating dual multipliers $\mathbf{y}$ with (2.6b), the dual to (2.6) minimizes y_1, subject to $\mathbf{A}^T\mathbf{y} \geq \mathbf{q}(\mathbf{x})$, where $\mathbf{y} \in \mathbb{R}^{n_r(n_c-2)+1}$, noting that $\mathbf{b}^T\mathbf{y} = y_1$. Since (2.6) is a feasible and bounded linear program, strong duality must hold. Combining this dual with the first-stage problem, we obtain

$$\min \quad y_1 \tag{2.9a}$$

$$\text{s.t.} \quad \mathbf{A}^T\mathbf{y} \geq \mathbf{q}(\mathbf{x}) \tag{2.9b}$$

$$\mathbf{x} \in X, \tag{2.9c}$$

which is equivalent to (2.5).

2.3. Two-Stage Lower-Bound Formulation. Because $\mathbf{q}(\mathbf{x})$ is not a convex function of $\mathbf{x}$, formulation (2.9) is not a convex program and is therefore very difficult to solve. In an effort to obtain a more tractable formulation, we consider every combination of monitors s and arcs ijk, and create a piecewise-linear concave function that underestimates the natural logarithm of the probability of evading monitor s while moving along arc ijk. We first partition the range of all possible values of the distance between $\mathbf{x}_s$ and $\mathbf{c}_{ijk}$ into a set of distance intervals. In particular, we define a set B_{sijk} so that the b-th distance interval is given by $[d_{sijk,b}, d_{sijk,b+1}]$, for $b \in B_{sijk} = \{1, \ldots, |B_{sijk}|\}$. More specifically, we choose the values of $d_{sijk,b}$, $\forall b = 1, \ldots, |B_{sijk}| + 1$, so that the first $|B_{sijk}| - 1$ distance intervals pertaining to distances less than or equal to R are of equal length, i.e., $d_{sijk,b} = \left(\frac{b-1}{B_{sijk}-1}\right) R$, $\forall b \in B_{sijk}$, with one more distance interval for all distances greater than R, i.e., $d_{sijk,|B_{sijk}|+1} = L + H$.

To create this piecewise-linear concave function, we introduce parameters $p_{sijk,b}$, $\forall b = 1, \ldots, |B_{sijk}| + 1$, $s \in S$, $i \in I\setminus\{n_c\}$, j, $k \in J$, so that $p_{sijk,b} = \psi(d_{sijk,b})$, $\forall b = 1, \ldots, |B_{sijk}| + 1$, where the b-th probability interval is given by $[e^{p_{sijk,b}}, e^{p_{sijk,b+1}}]$. Figure 2.2 depicts this function for an instance where $|B_{sijk}| = 5$, $H = 1000$, $L = 1000$, $R = 100$, and $\delta_p = 0.01$.

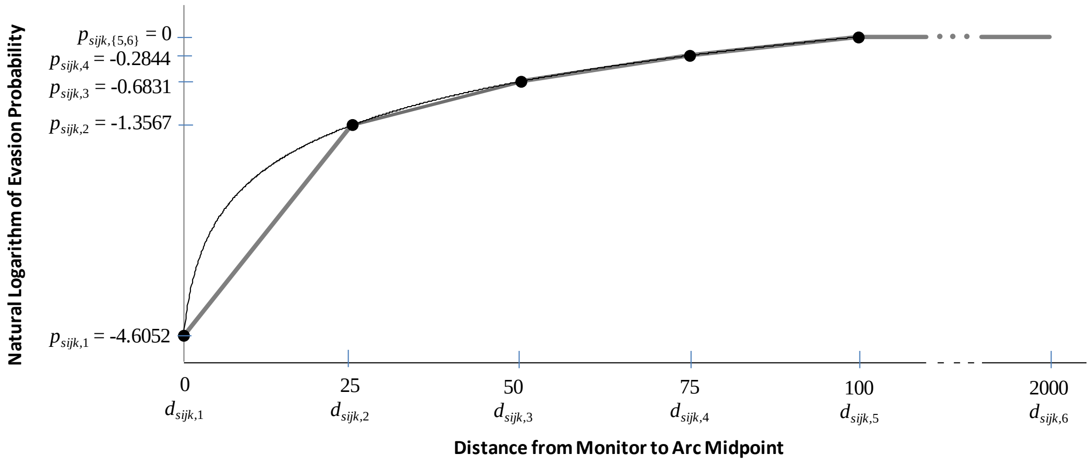

FIG. 2.2. *Curve corresponds to formulation* (2.5)*, while piecewise-linear concave function corresponds to formulation* (2.11)*.*

Subsequently, we introduce a vector of binary variables $w_{sijk,b},\ \forall s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J,\ b \in B_{sijk}$. Variable $w_{sijk,\hat{b}} = 1$ if monitor-arc combination $sijk$ occupies distance interval $\hat{b} \in B_{sijk}$ and $w_{sijk,b} = 0,\ \forall b \neq \hat{b}$. Accordingly, define

$$W = \left\{ \mathbf{w} : \sum_{b \in B_{sijk}} w_{sijk,b} = 1,\ w_{sijk,b} \in \{0,1\},\ \forall s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J,\ b \in B_{sijk} \right\} \tag{2.10}$$

to require that a monitor-arc combination $sijk$ occupies exactly one of its possible $|B_{sijk}|$ distance intervals. Using the foregoing elements, we produce the following two-stage lower-bound formulation:

$$\min_{(\mathbf{x},\mathbf{w}) \in X_{(2.11)}} \max \quad \sum_{i \in I \backslash \{n_c\}} \sum_{j \in J} \sum_{k \in J} \mathbf{t}(\mathbf{x},\mathbf{w})_{ijk} f_{ijk} \tag{2.11a}$$

$$\text{s.t.} \quad \mathbf{f} \in \Lambda, \tag{2.11b}$$

where

$$X_{(2.11)} = \{\mathbf{x} \in X,\ \mathbf{w} \in W :$$

$$\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\| \leq \sum_{b \in B_{sijk}} d_{sijk,b+1} w_{sijk,b} \quad \forall s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J \tag{2.11c}$$

$$x_{s,1} \leq x_{s+1,1} \quad \forall s \in S \backslash \{|S|\}\} \tag{2.11d}$$

and $\mathbf{t}(\mathbf{x},\mathbf{w})_{ijk}$ incorporates the lower-bounding piecewise-linear function and arc factors, given by

$$\mathbf{t}(\mathbf{x},\mathbf{w})_{ijk} = \sum_{s \in S} \sum_{b \in B_{sijk}} \left(\frac{p_{sijk,b+1} - p_{sijk,b}}{d_{sijk,b+1} - d_{sijk,b}} \left(||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})|| - d_{sijk,b+1} \right) + p_{sijk,b+1} + \ln(v_{ijk}) \right) w_{sijk,b}. \tag{2.12}$$

Constraints (2.11c) ensure that the monitor location is within an upper-bound distance, $d_{sijk,b+1}$, of distance interval b, if this particular monitor-arc combination $sijk$ has $w_{sijk,b} = 1$. Note that constraints ensuring that the monitor location is greater than or equal to its corresponding lower-bound distance $d_{sijk,b}$ are not needed because the optimal objective function value to the inner maximization problem of (2.11) is a nondecreasing function of $||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})||$. Therefore, there always exists an optimal solution in which the outer minimization problem sets $w_{sijk,b} = 1$ for the smallest index such that $\|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\| \leq d_{sijk,b+1}$. Constraints (2.11d) eliminate some alternative optimal solutions by ensuring that, for any monitor $\hat{s}$, every other monitor $s \geq \hat{s}$ is not located at a smaller column-axis coordinate than monitor $\hat{s}$.

THEOREM 2.1. *The optimal objective function value of* (2.11) *is a lower bound on* $G(\mathbf{x}^*, \mathbf{f}^*)$*, where* $(\mathbf{x}^*, \mathbf{f}^*)$ *is optimal to* (2.5).

Proof. Define $F_{(2.5)}$ and $F_{(2.11)}$ as the feasible region of (2.5) and (2.11), respectively. We show that every feasible solution $(\mathbf{x}, \mathbf{f})$ to $F_{(2.5)}$ corresponds to a feasible solution $(\mathbf{x}, \mathbf{w})$ to $F_{(2.11)}$, where the objective function value for $(\mathbf{x}, \mathbf{w})$ in (2.11) does not exceed $G(\mathbf{x}, \mathbf{f})$ in (2.5). Consider any feasible solution $(\bar{\mathbf{x}}, \bar{\mathbf{f}}) \in F_{(2.5)}$, and note that $\bar{\mathbf{f}} \in \Lambda$. For each $s \in S$ and arc ijk, let $\hat{b} \in B_{sijk}$ be such that $d_{sijk,\hat{b}} \leq ||\Upsilon(\bar{\mathbf{x}}_s - \mathbf{c}_{ijk})|| \leq d_{sijk,\hat{b}+1}$ (assumed to exist by construction), and set $\bar{w}_{sijk,\hat{b}} = 1$. Thus, we have established a $\bar{\mathbf{w}} : (\bar{\mathbf{x}}, \bar{\mathbf{w}}) \in F_{(2.11)}$. We now show that $G(\bar{\mathbf{x}}, \bar{\mathbf{f}})$ is at least as large as (2.11a) evaluated at $(\bar{\mathbf{x}}, \bar{\mathbf{w}}, \bar{\mathbf{f}})$. For any $sijk$ such that $s \in S$, $i \in I \backslash \{n_c\}$, $j,\ k \in J$, so that $||\Upsilon(\bar{\mathbf{x}}_s - \mathbf{c}_{ijk})||$ lies in the b-th distance interval for s and ijk, we have

$$\begin{aligned}&\ln\left(\min\left\{\frac{||\Upsilon(\bar{\mathbf{x}}_s - \mathbf{c}_{ijk})||}{R}(1-\delta_p)+\delta_p, 1\right\}\right) + \ln(v_{ijk}) \\ &= \ln\left(\min\left\{\lambda\left(\frac{d_{sijk,\hat{b}+1}}{R}(1-\delta_p)+\delta_p\right) + (1-\lambda)\left(\frac{d_{sijk,\hat{b}}}{R}(1-\delta_p)+\delta_p\right), 1\right\}\right) + \ln(v_{ijk}), \qquad (2.13)\end{aligned}$$

for

$$\lambda = \frac{||\Upsilon(\bar{\mathbf{x}}_s - \mathbf{c}_{ijk})|| - d_{sijk,\hat{b}}}{d_{sijk,\hat{b}+1} - d_{sijk,\hat{b}}}. \qquad (2.14)$$

Continuing this analysis, the expression in (2.13) is at least

$$\begin{aligned}&\lambda \ln\left(\min\left\{\frac{d_{sijk,\hat{b}+1}}{R}(1-\delta_p)+\delta_p, 1\right\}\right) \\ &\qquad + (1-\lambda)\ln\left(\min\left\{\frac{d_{sijk,\hat{b}}}{R}(1-\delta_p)+\delta_p, 1\right\}\right) + \ln(v_{ijk}) \qquad (2.15)\\ &= \lambda p_{sijk,\hat{b}+1} + (1-\lambda)p_{sijk,\hat{b}} + \ln(v_{ijk}) \qquad (2.16)\\ &= \frac{||\Upsilon(\bar{\mathbf{x}}_s - \mathbf{c}_{ijk})|| - d_{sijk,\hat{b}}}{d_{sijk,\hat{b}+1} - d_{sijk,\hat{b}}} p_{sijk,\hat{b}+1} \\ &\qquad + \frac{d_{sijk,\hat{b}+1} - ||\Upsilon(\bar{\mathbf{x}}_s - \mathbf{c}_{ijk})||}{d_{sijk,\hat{b}+1} - d_{sijk,\hat{b}}} p_{sijk,\hat{b}} + \ln(v_{ijk}) \qquad (2.17)\\ &= \left(\frac{p_{sijk,\hat{b}+1} - p_{sijk,\hat{b}}}{d_{sijk,\hat{b}+1} - d_{sijk,\hat{b}}}\left(||\Upsilon(\bar{\mathbf{x}}_s - \mathbf{c}_{ijk})|| - d_{sijk,\hat{b}+1}\right)\right. \\ &\qquad \left. + p_{sijk,\hat{b}+1} + \ln(v_{ijk})\right)\bar{w}_{sijk,\hat{b}} \qquad (2.18)\\ &= \mathbf{t}(\bar{\mathbf{x}}, \bar{\mathbf{w}})_{sijk}, \qquad (2.19)\end{aligned}$$

where (2.15) follows by concavity of the natural logarithm function, (2.17) follows from (2.14), and (2.19) follows from defining $\mathbf{t}(\mathbf{x}, \mathbf{w})_{sijk}$ as the s-th term in the outer summation of (2.12) and the fact that $\bar{w}_{sijk,b} = 0$

for all $b \neq \hat{b}$. Since (2.13)–(2.19) holds true for any set of monitor locations $\bar{\mathbf{x}}$ corresponding to any feasible solution $(\bar{\mathbf{x}}, \bar{\mathbf{f}}) \in F_{(2.5)}$ (and its corresponding solution $(\bar{\mathbf{x}}, \bar{\mathbf{w}}) \in F_{(2.11)}$), $G(\bar{\mathbf{x}}, \bar{\mathbf{f}})$ is greater than or equal to the objective function value of (2.11) at $(\bar{\mathbf{x}}, \bar{\mathbf{w}})$. ■

To obtain a single-stage lower-bound formulation, we perform a similar transformation on (2.11) as performed on (2.5). The dual to the second-stage problem of (2.11) minimizes u_1, subject to $\mathbf{A}^T\mathbf{u} \geq \mathbf{t}(\mathbf{x}, \mathbf{w})$; combining this dual with the first stage of (2.11), we obtain the one-stage problem

$$\min \quad u_1 \tag{2.20a}$$

$$\text{s.t.} \quad \mathbf{A}^T\mathbf{u} \geq \mathbf{t}(\mathbf{x}, \mathbf{w}) \tag{2.20b}$$

$$(\mathbf{x}, \boldsymbol{\Omega}, \mathbf{w}) \in X_{(2.20)}, \tag{2.20c}$$

where

$$X_{(2.20)} = \{\mathbf{x} \in X, \boldsymbol{\Omega} \geq \mathbf{0}, \mathbf{w} \in W : \tag{2.20d}$$

$$\sum_{o=1}^{2} \Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk} + \boldsymbol{\Omega}_{sijk})_o \leq \sum_{b \in B_{sijk}} d_{sijk,b+1} w_{sijk,b} \quad \forall s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J \tag{2.20e}$$

$$x_{s,o} - c_{ijk,o} + \Omega_{sijk,o} \geq c_{ijk,o} - x_{s,o} \quad \forall o \in \{1, 2\},\ s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J \tag{2.20f}$$

$$x_{s,2} \leq x_{s+1,2} \quad \forall s \in S \backslash \{|S|\}\}. \tag{2.20g}$$

Constraints (2.20f) ensure that the following relationship holds:

$$\sum_{o=1}^{2} \Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk} + \boldsymbol{\Omega}_{sijk})_o = \|\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})\|, \tag{2.21}$$

where $\boldsymbol{\Omega}$ elements attain their minimum possible values as dictated by (2.20b) and (2.20e), which guarantees that $\mathbf{x}_s - \mathbf{c}_{ijk} + \boldsymbol{\Omega}_{sijk} = ||\mathbf{x}_s - \mathbf{c}_{ijk}||$. Observe that $\mathbf{t}(\mathbf{x}, \mathbf{w})_{ijk}$ is nonlinear in (2.20) due to the multiplication of $\mathbf{x}$ and $\mathbf{w}$. To linearize these terms, we can add additional variables and constraints as prescribed in [14], but at the expense of incurring a far more complex model. Instead, define $\tau(\mathbf{x}, \boldsymbol{\Omega})_{sijk,b}$ to be the linear segment of $\mathbf{t}(\mathbf{x}, \mathbf{w})_{ijk}$ corresponding to monitor s and segment b, i.e.,

$$\tau(\mathbf{x}, \boldsymbol{\Omega})_{sijk,b} = \frac{p_{sijk,b+1} - p_{sijk,b}}{d_{sijk,b+1} - d_{sijk,b}} \left(\sum_{o=1}^{2} \Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk} + \boldsymbol{\Omega}_{sijk})_o - d_{sijk,b+1} \right) + p_{sijk,b+1} + \ln(v_{ijk}). \tag{2.22}$$

We then employ the following formulation, in which (2.23b) and (2.23c) below replace constraints (2.20b).

$$\min \quad u_1 \tag{2.23a}$$

$$\text{s.t.} \quad \mathbf{A}^T\mathbf{u} \geq \mathbf{g} \tag{2.23b}$$

$$\rho_{sijk} \geq \tau(\mathbf{x}, \boldsymbol{\Omega})_{sijk,b} - M_{sijk,b}(1 - w_{sijk,b}) \quad \forall s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J,\ b \in B_{sijk} \tag{2.23c}$$

$$(\mathbf{x}, \boldsymbol{\Omega}, \mathbf{w}) \in X_{(2.20)}, \tag{2.23d}$$

where

$$\mathbf{g}_{ijk} = \sum_{s \in S} \rho_{sijk}, \quad \forall i \in I \backslash \{n_c\},\ j,\ k \in J, \tag{2.24}$$

and

$$M_{sijk,b} = \max\Bigg\{ -\frac{p_{sijk,b+1}-p_{sijk,b}}{d_{sijk,b+1}-d_{sijk,b}} d_{sijk,b+1} + p_{sijk,b+1} - \ln(\delta_p), \\ \frac{p_{sijk,b+1}-p_{sijk,b}}{d_{sijk,b+1}-d_{sijk,b}}(L+H-d_{sijk,b+1}) + p_{sijk,b+1}\Bigg\}, \quad (2.25)$$

$\forall s \in S,\ i \in I\backslash\{n_c\},\ j,\ k \in J,\ b \in B_{sijk}$. Constraints (2.23c) define ρ_{sijk} to be the adversary's evasion probability with respect to monitor s and arc ijk, as measured by the piecewise-linear underestimation function. Observe that (2.23c) is binding for the segment $b \in B_{sijk}$ for which $w_{sijk,b} = 1$, provided that the M-constants in (2.25) are sufficiently large to guarantee that (2.23c) is satisfied when $w_{sijk,b} = 0$. To achieve this, we compute the maximum amount by which a given line segment overestimates the piecewise-linear function, which occurs at either boundary of the range of possible distances from s to ijk. Accordingly, the first (second) term of the maximizing expression of (2.25) is the difference between the segment function and the piecewise-linear function computed at a distance of zero ($L+H$). (Note that the term $L+H$ used in (2.25) for the maximum distance between monitor s and arc ijk is conservative; in our computations, we instead use the tighter bound

$$\max\{n_c - c_{ijk,1}, c_{ijk,1} - 1\}\frac{L}{n_c - 1} + \max\{H - c_{ijk,2}, c_{ijk,2} - 1\}\frac{H}{n_r - 1}.$$

Preliminary results reveal that this substitution, which results in a tightened value for the M-parameters, leads to improved running times.)

3. Optimal Continuous-Space Algorithm. Observe that aside from the binary restrictions on $\mathbf{w}$, (2.23) is a convex problem, and is far easier to solve than the (exact) nonconvex problem (2.5). We thus prescribe an algorithm that solves a series of instances (2.23), which yield lower and upper bounds on the optimal objective function value of (2.5) in § 3.1. Section 3.2 presents an upper-bounding improvement scheme for accelerating the convergence of our proposed algorithm.

Before presenting our approach, we first present the following result.

Lemma 3.1. *Let $n \in \mathbb{Z}^+$, $0 \le L_i \le U_i \le 1$, $\forall i = 1, \ldots, n$, and $0 < \alpha \le \prod_{i=1}^n U_i - \prod_{i=1}^n L_i$. Then, $L_i \le U_i - \alpha/n$, for some $i = 1, \ldots, n$.*

Proof. By contradiction, suppose $L_i > U_i - \alpha/n$, $\forall i = 1, \ldots, n$. Using the foregoing assumptions, we have

$$\prod_{i=1}^n U_i - \prod_{i=1}^n L_i < \prod_{i=1}^n U_i - \left(U_1 - \frac{\alpha}{n}\right)\prod_{i=2}^n L_i \quad (3.1)$$

$$\le \prod_{i=1}^n U_i - U_1\prod_{i=2}^n L_i + \frac{\alpha}{n}, \quad (3.2)$$

where (3.1) follows by $L_1 > U_1 - \alpha/n$, and (3.2) is due to the assumption that $\alpha/n > 0$ and $0 \le L_i \le 1$, $\forall i = 1, \ldots, n$. Repeating the analysis in (3.1)–(3.2) for $j = 2$ yields

$$\prod_{i=1}^n U_i - \prod_{i=1}^n L_i < \prod_{i=1}^n U_i - U_1\left(U_2\prod_{i=3}^n L_i - \frac{\alpha}{n}\right) + \frac{\alpha}{n}, \quad (3.3)$$

$$\le \prod_{i=1}^n U_i - U_1U_2\prod_{i=3}^n L_i + \frac{2\alpha}{n} \quad (3.4)$$

where the last inequality is due to $U_1 \leq 1$. Repeating this procedure for n substitutions, we obtain

$$\prod_{i=1}^{n} U_i - \prod_{i=1}^{n} L_i < \prod_{i=1}^{n} U_i - \prod_{i=1}^{n} U_i + \frac{n\alpha}{n} \tag{3.5}$$

$$= \alpha, \tag{3.6}$$

which achieves the desired contradiction. ■

3.1. Overall Approach. This approach first solves (2.23) to obtain a lower bound on (2.5), and then extracts from this solution a feasible set of monitor locations to use in (2.6), thus obtaining an upper bound. Then, we iteratively refine (2.23) by modifying the piecewise-linear underestimation function until we reach a desired termination gap between the upper and lower bound. The steps of the algorithm, which we call ExactAlg, are as follows.

ExactAlg

0. Choose $\epsilon_{gap} > 0$, and initialize UB $= 0$, LB $= -\infty$, and $\bar{\mathbf{x}}$ and $\bar{\mathbf{f}}$ to be empty.
1. Solve (2.23), obtaining $(\hat{\mathbf{u}}, \hat{\boldsymbol{\rho}}, \hat{\mathbf{x}}, \hat{\boldsymbol{\Omega}}, \hat{\mathbf{w}})$. Set LB $= \hat{u}_1$.
2. Fixing $\mathbf{x} = \hat{\mathbf{x}}$, solve (2.6), and obtain an optimal evasion path $\dot{\mathbf{f}}$.
3. Compute $G(\hat{\mathbf{x}},\ \dot{\mathbf{f}})$. If $G(\hat{\mathbf{x}},\ \dot{\mathbf{f}}) <$ UB, then set $(\text{UB}, \bar{\mathbf{x}}, \bar{\mathbf{f}}) = (G(\hat{\mathbf{x}},\ \dot{\mathbf{f}}), \hat{\mathbf{x}}, \dot{\mathbf{f}})$.
4. Compute GAP $= e^{\text{UB}} - e^{\text{LB}}$. If GAP $< \epsilon_{\text{gap}}$, then terminate with monitor locations $\bar{\mathbf{x}}$, adversary path $\bar{\mathbf{f}}$, and optimality gap $[e^{\text{LB}}, e^{\text{UB}}]$. Otherwise, continue to Step 5.
5. For each $s \in S,\ i \in I\backslash\{n_c\},\ j,\ k \in J,\ b \in B_{sijk}$, such that $\dot{f}_{ijk} = 1$ and $\hat{w}_{sijk,b} = 1$, compute $\text{GAP}_{sijk,b}$ as the difference between the actual probability that the evader avoids detection by monitor s on arc ijk and the probability estimated by segment b. That is,

$$\text{GAP}_{sijk,b} = e^{\psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||) + \ln(v_{ijk})} - e^{\left(\tau(\hat{\mathbf{x}}, \hat{\boldsymbol{\Omega}})_{sijk,b}\right)}. \tag{3.7}$$

   If $\text{GAP}_{sijk,b} \geq \frac{\epsilon_{\text{gap}}}{|S|(n_c - 1)}$, then do the following:
   (a) Increment $|B_{sijk}|$ by 1.
   (b) For all $q = |B_{sijk}|+1, \ldots, b+1$ (in descending order), set $d_{sijk,q+1} = d_{sijk,q}$ and $p_{sijk,q+1} = p_{sijk,q}$.
   (c) Set $d_{sijk,b+1} = ||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||$ and $\ln(p_{sijk,b+1}) = \psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||)$.

   Return to Step 1.

In Step 0, UB and LB are set to be the initial upper and lower bounds on the optimal objective function value to (2.5), respectively, while $\bar{\mathbf{x}}$ and $\bar{\mathbf{f}}$ are the incumbent monitor locations and adversary path, respectively. In Step 1, we update the current lower bound to the objective function value obtained from solving the lower-bound problem (2.23). In Step 2, we find the adversary's optimal path $\dot{\mathbf{f}}$ given monitor locations at $\hat{\mathbf{x}}$. Step 3 computes $G(\hat{\mathbf{x}},\ \dot{\mathbf{f}})$, decides whether this value is better than the best upper bound found so far, and updates UB, $\bar{\mathbf{x}}$, and $\bar{\mathbf{f}}$ accordingly.

In Step 4, we terminate with a near-optimal solution if the current optimality gap is sufficiently small. Otherwise, we compute GAP-values as in (3.7) for each monitor-arc combination $sijk$, and distance interval $b \in B_{sijk}$ such that $\dot{f}_{ijk} = \hat{w}_{sijk,b} = 1$. There exists at least one such term that is at least $\epsilon_{\text{gap}} \left(|S|(n_c - 1)\right)^{-1}$; we prove this claim as follows. We observe that at most $|S|(n_c - 1)$ nonzero GAP-values exist because $\dot{f}_{ijk} = 1$

for $n_c - 1$ arcs, there are $|S|$ monitors, and for each $s \in S$, exactly one value $\hat{w}_{sijk,b} = 1$. From these $|S|(n_c-1)$ terms, create a set of values $L_l = e^{\left(\tau(\hat{\mathbf{x}},\hat{\mathbf{\Omega}})_{sijk,b}\right)}$ and $U_l = e^{\psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||)+\ln(v_{ijk})}$, $\forall l = 1,\ldots,|S|(n_c-1)$, for some $sijk$ and b satisfying $\dot{f}_{ijk} = 1$ and $\hat{w}_{sijk,b} = 1$. By Lemma 3.1 with $\alpha = \epsilon_{\text{gap}}$, $n = |S|(n_c-1)$, and U_l and L_l defined as above, $\forall l = 1,\ldots,n$, we know that $\text{GAP}_{sijk,b} \geq \epsilon_{\text{gap}}\left(|S|(n_c-1)\right)^{-1}$ for some $s \in S,\ i \in I\backslash\{n_c\},\ j,\ k \in J,\ b \in B_{sijk}$. In Step 5, for each such term $\text{GAP}_{sijk,b} \geq \epsilon_{\text{gap}}\left(|S|(n_c-1)\right)^{-1}$, let $d' \in [d_{sijk,b}, d_{sijk,b+1}]$ be the distance from $\hat{\mathbf{x}}_s$ to $\mathbf{c}_{ijk}$. Define the term "breakpoint" as the endpoint of a distance or probability interval $b \in B_{sijk}$. We create a new breakpoint in the piecewise-linear function $\mathbf{t}(\mathbf{x},\mathbf{w})_{sijk}$ at d', and replace this interval with $[d_{sijk,b}, d']$ and $[d', d_{sijk,b+1}]$. As a result, if monitor s is placed at a distance $||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||$ from arc ijk, then there no longer exists a gap at this point between the updated piecewise-linear concave lower-bound function and the function it approximates.

Before proving convergence of our algorithm in Theorem 3.4, we state two lemmas that will be used in this proof.

Lemma 3.2. *Let $\beta:\ \mathbb{R} \to \mathbb{R}$ be a concave, monotonically nondecreasing function. For scalars $d_1 \geq 0$ and $d_2 > 0$, we have*

$$\min\left\{\beta(\zeta_1+\zeta_2) - \beta(\zeta_1):\ \zeta_1+\zeta_2 \leq d_1+d_2,\ \zeta_2 \geq d_2,\ \zeta_1 \geq 0\right\} = \beta(d_1+d_2) - \beta(d_1). \tag{3.8}$$

Proof. For ζ_1 and ζ_2 such that $\zeta_1+\zeta_2 \leq d_1+d_2$, and $\zeta_2 \geq d_2$, it follows that $\zeta_1 \leq d_1$. We have

$$\frac{\beta(\zeta_1+\zeta_2)-\beta(\zeta_1)}{\zeta_2} \geq \frac{\beta(d_1+d_2)-\beta(\zeta_1)}{d_1+d_2-\zeta_1} \tag{3.9}$$

$$\geq \frac{\beta(d_1+d_2)-\beta(d_1)}{d_2}, \tag{3.10}$$

where (3.9) and (3.10) follow because β is concave. Thus, we have

$$\left(\frac{\beta(\zeta_1+\zeta_2)-\beta(\zeta_1)}{\zeta_2}\right)\zeta_2 \geq \left(\frac{\beta(\zeta_1+\zeta_2)-\beta(\zeta_1)}{\zeta_2}\right)d_2 \tag{3.11}$$

$$\geq \left(\frac{\beta(d_1+d_2)-\beta(d_1)}{d_2}\right)d_2, \tag{3.12}$$

where (3.11) follows from the fact that $\zeta_2 \geq d_2$, and (3.12) follows from the result proven by (3.9) and (3.10). Equality in (3.8) holds by setting $\zeta_1 = d_1$ and $\zeta_2 = d_2$. ∎

Lemma 3.3. *Let $\beta:\mathbb{R}\to\mathbb{R}$ be a concave, monotonically nondecreasing function. For scalars $d_1 \geq 0$, and $d_2 > 0$, we have*

$$\max\left\{\beta(\zeta_1+\zeta_2)-\beta(\zeta_1):\ \zeta_1 \geq d_1,\ 0 < \zeta_2 \leq d_2\right\} = \beta(d_1+d_2)-\beta(d_1). \tag{3.13}$$

Proof. First, note that because β is nondecreasing, $\beta(\zeta_1+d_2) \geq \beta(\zeta_1+\zeta_2)$ for any ζ_1, and for any $\zeta_2 > 0$. Hence, the optimization problem in (3.13) is maximized by setting $\zeta_2 = d_2$. The fact that β is concave and nondecreasing then implies that $\max\{\beta(\zeta_1+d_2)-\beta(\zeta_1):\ \zeta_1 \geq d_1\}$ is optimized at $\zeta_1 = d_1$. This completes the proof. ∎

Theorem 3.4. *In ExactAlg, GAP $< \epsilon_{gap}$ after at most*

$$\frac{|S|^2 n_r^2 (n_c-1)^2(1-\delta_p)}{\epsilon_{gap}\delta_p} \tag{3.14}$$

iterations.

Proof. Suppose that during the a-th iteration of ExactAlg, GAP $\geq \epsilon_{\text{gap}}$, $\mathbf{x} = \hat{\mathbf{x}}$, $\mathbf{\Omega} = \hat{\mathbf{\Omega}}$, $\mathbf{w} = \hat{\mathbf{w}}$, and $\mathbf{f} = \dot{\mathbf{f}}$. We know that $\text{GAP}_{sijk,b,a} \geq \frac{\epsilon_{\text{gap}}}{|S|(n_c-1)}$ for some monitor-arc combination $sijk$, and some b such that $\hat{w}_{sijk,b,a} = 1$ and $d_{sijk,b+1,a} \leq R$, where $\text{GAP}_{sijk,b,a}$ is the value of $\text{GAP}_{sijk,b}$ in iteration a (and similar definitions apply for $d_{sijk,b,a}$, $\hat{w}_{sijk,b,a}$, $B_{sijk,a}$, $\tau(\hat{\mathbf{x}}, \hat{\mathbf{\Omega}})_{sijk,b,a}$, and $p_{sijk,b,a}$). Hence, at least one distance interval for $sijk$ is refined in iteration a. Thus, defining $\alpha = e^{\psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||)}$, we first note that

$$\begin{aligned}
\text{GAP}_{sijk,b,a} &= e^{\psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||) + \ln(v_{ijk})} - e^{\tau(\hat{\mathbf{x}}, \hat{\mathbf{\Omega}})_{sijk,b,a}} \\
&= v_{ijk} \left(e^{\psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||)} - e^{\tau(\hat{\mathbf{x}}, \hat{\mathbf{\Omega}})_{sijk,b,a} - \ln(v_{ijk})} \right) \\
\Rightarrow \quad \alpha - \frac{\text{GAP}_{sijk,b,a}}{v_{ijk}} &= e^{\tau(\hat{\mathbf{x}}, \hat{\mathbf{\Omega}})_{sijk,b,a} - \ln(v_{ijk})}. \qquad (3.15)
\end{aligned}$$

Using (3.15), we obtain

$$\begin{aligned}
p_{sijk,b+1,a+1} - p_{sijk,b,a+1} &> \psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||) - \left(\tau(\hat{\mathbf{x}}, \hat{\mathbf{\Omega}})_{sijk,b,a} - \ln(v_{ijk}) \right) && (3.16a) \\
&= \ln(\alpha) - \ln\left(\alpha - \frac{\text{GAP}_{sijk,b,a}}{v_{ijk}} \right) && (3.16b) \\
&\geq \ln(\alpha) - \ln(\alpha - \text{GAP}_{sijk,b,a}) && (3.16c) \\
&\geq -\ln\left(1 - \frac{\epsilon_{\text{gap}}}{|S|(n_c - 1)} \right), && (3.16d)
\end{aligned}$$

where (3.16a) follows because $p_{sijk,b+1,a+1} = \psi(||\Upsilon(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk})||)$ and $p_{sijk,b,a+1} < \left(\tau(\hat{\mathbf{x}}, \hat{\mathbf{\Omega}})_{sijk,b,a} - \ln(v_{ijk}) \right)$ because $p_{sijk,b,a+1} = p_{sijk,b,a}$, noting that $\ln(\cdot)$ is strictly increasing, (3.16b) follows by definition of α and $\text{GAP}_{sijk,b,a}$, (3.16c) follows because $0 < v_{ijk} \leq 1$, and (3.16d) follows by Lemma 3.2, with $\beta(\cdot) = \ln(\cdot)$, $d_1 = 1 - \epsilon_{\text{gap}} \left(|S|(n_c - 1) \right)^{-1}$, $d_2 = \epsilon_{\text{gap}} \left(|S|(n_c - 1) \right)^{-1}$, $\alpha \leq d_1 + d_2$, and $\text{GAP}_{sijk,b,a} \geq d_2$.

At any iteration a of the algorithm, define $p' = e^{p_{sijk,b',a}} \geq \delta_p$ for some $b' \in B_{sijk,a}$ such that $\hat{w}_{sijk,b',a} = 1$, where p' is the endpoint of an interval $[e^{p_{sijk,b',a+1}},\ p']$ or $[p',\ e^{p_{sijk,b'+1,a+1}}]$ created for iteration $a+1$, and where $e^{p_{sijk,b',a+1}}$ and $e^{p_{sijk,b'+1,a+1}}$ are the left and right endpoints, respectively, for the former and latter probability intervals, respectively. For ease of notation, we denote $p_L = e^{p_{sijk,b',a+1}}$ in the former case and $p_R = e^{p_{sijk,b'+1,a+1}}$ in the latter case. Note that $p_L \neq p'$ and $p_R \neq p'$ because GAP-values are zero at endpoints.

First, consider the interval $[p', p_R]$ and define $l = p_R - p'$. We seek an $l^* < l$ such that $p_R \notin [p',\ p' + l^*]$. To bound l, note that

$$\frac{p' + l}{p'} > e^{-\ln\left(1 - \frac{\epsilon_{\text{gap}}}{|S|(n_c - 1)}\right)} = \left(1 - \frac{\epsilon_{\text{gap}}}{|S|(n_c - 1)} \right)^{-1}$$

by (3.16) and the definitions of p' and l. Rearranging terms, we obtain

$$\begin{aligned}
l &> p' \left(1 - \frac{\epsilon_{\text{gap}}}{|S|(n_c - 1)} \right)^{-1} - p' && (3.17) \\
&\geq \frac{\epsilon_{\text{gap}} \delta_p}{|S|(n_c - 1) - \epsilon_{\text{gap}}}, && (3.18)
\end{aligned}$$

since $p' \geq \delta_p$. Letting l^* be the right-hand side of (3.18) and using the fact that $p_R = p' + l$, we obtain

$$l^* < p_R - p', \qquad (3.19)$$

i.e., $p_R \notin [p',\ p'+l^*]$.

Now, consider $[p_L, p']$. Define $p'' \in [\delta_p, p']$ to be the closest breakpoint smaller than p' that exists prior to iteration a, i.e., $p'' = \max_{b \in B_{sijk,a}}\{e^{p_{sijk,b,a}} :\ e^{p_{sijk,b,a}} < p'\}$. We now show that $p_L \notin [p'-l^*, p']$. Suppose, by contradiction, that $p_L \in [p'-l^*, p']$. Define $\bar{x}$ (and, accordingly, $\bar{\Omega}$) as the monitor locations that result in the new breakpoint p_L. Noting that $l^* > 0$, we have

$$\begin{aligned}
-\ln\left(1 - \frac{\epsilon_{\text{gap}}}{|S|(n_c-1)}\right) &= \ln(\delta_p + l^*) - \ln(\delta_p) && (3.20)\\
&> \frac{l^*\left(\ln(p') - \ln(\delta_p)\right)}{p' - \delta_p} && (3.21)\\
&\geq \frac{l^*\left(\ln(p') - \ln(p'')\right)}{p' - p''} && (3.22)\\
&> \ln(p') + \frac{(p'-p_L)\left(\ln(p') - \ln(p'')\right)}{p' - p''} - \ln(p') && (3.23)\\
&\geq \ln(p_L) + \frac{(p'-p_L)\left(\ln(p') - \ln(p'')\right)}{p' - p''} - \ln(p') && (3.24)\\
&= \ln(p_L) - \left(\tau(\bar{\mathbf{x}}, \bar{\boldsymbol{\Omega}})_{sijk,b',a+1} - \ln(v_{ijk})\right) && (3.25)
\end{aligned}$$

where (3.20) follows by substituting l^* by (3.18), (3.21) follows because $\delta_p + l^* < p'$ (by (3.19) with $p_R = p'$ and $p' = \delta_p$) and concavity of $\ln(\cdot)$, (3.22) follows because $\delta_p < p''$ and concavity of $\ln(\cdot)$, (3.23) follows because $p' - p_L < l^*$, (3.24) follows because $p_L < p'$ and by monotonicity of $\ln(\cdot)$, and (3.25) follows by (2.22) and the definition of $\bar{x}$ and $\bar{\Omega}$. Combining (3.20)–(3.25), we obtain the desired contradiction because we should have $-\ln\left(1 - \frac{\epsilon_{\text{gap}}}{|S|(n_c-1)}\right) \leq \ln(p_L) - \left(\tau(\bar{\mathbf{x}}, \bar{\boldsymbol{\Omega}})_{sijk,b',a+1} - \ln(v_{ijk})\right)$ by (3.16) with $p_{sijk,b,a+1} = p_L$ and $p_{sijk,b+1,a+1} = p'$. Thus, $p_L \notin [p'-l^*,\ p']$.

Because each interval length is at least l^*, and $p_{sijk,|B_{sijk}|+1,1} - p_{sijk,1,1} = 1 - \delta_p$, then after $(1-\delta_p)(l^*)^{-1}$ refinements of any one set of distance intervals for some monitor-arc combination $sijk$, the expression (3.7) for $sijk$ must be less than $\epsilon_{\text{gap}}\left(|S|(n_c-1)\right)^{-1}$. Because there are $|S|n_r^2(n_c-1)$ monitor-arc combinations, the termination criterion is reached after at most

$$\frac{|S|n_r^2(n_c-1)(1-\delta_p)}{l^*} = |S|n_r^2(n_c-1)(1-\delta_p)\frac{|S|(n_c-1)-\epsilon_{\text{gap}}}{\epsilon_{\text{gap}}\delta_p} \leq \frac{|S|^2 n_r^2(n_c-1)^2(1-\delta_p)}{\epsilon_{\text{gap}}\delta_p} \tag{3.26}$$

iterations. ∎

3.2. Improvement Step. To reduce the total number of integer programs solved by our algorithm, before updating the upper bound in Step 3, we attempt to move the monitors to more effectively interdict path $\dot{\mathbf{f}}$. For each $s \in S$, we adjust the location $\hat{\mathbf{x}}_s$ of monitor s in a direction that more efficiently interdicts $\dot{\mathbf{f}}$, while ensuring that $\dot{\mathbf{f}}$ remains optimal to (2.6). Accordingly, our approach will seek to move these monitors as far as possible in the direction opposite a subgradient of the objective function of (2.6) with respect to $\mathbf{x}_s$.

Because the objective function of (2.6) is not continuously differentiable with respect to $\mathbf{x}$, we will instead consider the following equivalent objective function.

$$\nu_{\mathbf{f},\mathbf{w}}(\mathbf{x}) = \sum_{i \in I \setminus \{n_c\}} \sum_{j \in J} \sum_{k \in J} \sum_{s \in S} \ln\left(\frac{||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})||}{R}(1-\delta_p) + \delta_p\right)\left(1 - w_{sijk,|B_{sijk}|}\right) f_{ijk}. \tag{3.27}$$

THEOREM 3.5. *Given* $\mathbf{w}$ *obtained from solving* (2.23)*, the function in* (2.6a) *is equivalent to* $\nu_{\mathbf{f},\mathbf{w}}(\mathbf{x})$.

Proof. For any monitor-arc combination $sijk$, two cases exist.

Case 1: $w_{sijk,|B_{sijk}|} = 1$: In this case we have $||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})|| \geq R$, implying that $\psi(||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})||) = 0$; thus, the function in (2.6a) and $\nu_{\mathbf{f},\mathbf{w}}(\mathbf{x})$ are equivalent.

Case 2: $w_{sijk,|B_{sijk}|} = 0$: In this case, we have

$$\psi(||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})||) = \ln\left(\frac{||\Upsilon(\mathbf{x}_s - \mathbf{c}_{jik})||}{R}(1-\delta_p) + \delta_p\right)$$

and the function in (2.6a) and $\nu_{\mathbf{f},\mathbf{w}}(\mathbf{x})$ are equivalent. ■

If we perturb each $||\Upsilon(\mathbf{x}_s - \mathbf{c}_{ijk})||$ in (3.27) by some small $\epsilon > 0$, $\nu_{\mathbf{f},\mathbf{w}}(\mathbf{x})$ is continuously differentiable with respect to $\mathbf{x}_s$. Using this perturbation, observe that given an optimal adversary path $\dot{\mathbf{f}}$ based on a set of monitor locations $\hat{\mathbf{x}}$, we compute the negative gradient direction, $-\nabla_{\mathbf{x}_q}\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\hat{\mathbf{x}})$, with respect to some monitor $q \in S$. We then seek an optimal placement of monitor q along this direction such that $\dot{\mathbf{f}}$ remains optimal, which is equivalent to solving

$$\min_{\mathbf{y},\mu} \quad \sum_{i\in I\setminus\{n_c\}}\sum_{j\in J}\sum_{k\in J}\varphi(\mu)_{ijk}\dot{f}_{ijk} \tag{3.28a}$$

$$\text{s.t.} \quad (0,0) \leq \Upsilon\left(\hat{\mathbf{x}}_q - \frac{\nabla_{\mathbf{x}_q}\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\hat{\mathbf{x}})}{||\nabla_{\mathbf{x}_q}\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\hat{\mathbf{x}})||}\mu\right) \leq (L,H) \tag{3.28b}$$

$$\mathbf{A}^T\mathbf{y} \geq \varphi(\mu) \tag{3.28c}$$

$$y_1 \leq \sum_{i\in I\setminus\{n_c\}}\sum_{j\in J}\sum_{k\in J}\varphi(\mu)_{ijk}\dot{f}_{ijk} \tag{3.28d}$$

$$\mu \geq 0, \tag{3.28e}$$

where

$$\varphi(\mu)_{ijk} = \sum_{s\in S\setminus\{q\}}\left(\psi\left(||\Upsilon\left(\hat{\mathbf{x}}_s - \mathbf{c}_{ijk}\right)||\right) + \ln(v_{ijk}) + \psi\left(\left|\left|\Upsilon\left(\hat{\mathbf{x}}_q - \frac{\nabla_{\mathbf{x}_q}\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\hat{\mathbf{x}})}{||\nabla_{\mathbf{x}_q}\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\hat{\mathbf{x}})||}\mu - \mathbf{c}_{ijk}\right)\right|\right|\right)\right).$$

Constraints (3.28b) state that the updated location of monitor q must be feasible to the original problem (2.5). Constraint (3.28c) ensures feasibility of the dual to (2.6). Constraint (3.28d) requires that strong duality holds between (2.6) and its dual, while (3.28e) guarantees that the stepsize μ is non-negative. (Note that anti-symmetry constraints, (2.11d), have been omitted from (3.28) to allow for feasbility of the most-improved location for monitor q.) As long as constraints (3.28b)–(3.28e) are satisfied for the updated location of monitor q, then we have a solution in which the adversary's optimal path $\dot{\mathbf{f}}$ remains the same, but the evasion probability for $\dot{\mathbf{f}}$ is less than or equal to its previous value.

Unfortunately, (3.28) is not a convex problem, noting that $\varphi(\mu)$ is not a convex function of μ. Rather than attempting to solve (3.28) directly, we instead seek a $\mu \in \text{Proj}_\mu(F_{(3.28)}) = \{\mu\colon \exists\mathbf{y} \text{ such that } (\mu,\mathbf{y}) \in F_{(3.28)}\}$ such that we obtain an overall decrease in the objective function value of (2.6) based on the adjusted monitor locations, where $F_{(3.28)}$ is the feasible region of (3.28). To compute a stepsize μ, we execute a bisection search procedure, which we call Stepsize. For this procedure, we denote the input monitor placement vector by $\dot{\mathbf{x}}$ and the adjusted monitor solution by $\mathbf{x}'$. Note that given μ, $\mathbf{x}'_s = \dot{\mathbf{x}}_s$, $\forall s \in S \setminus \{q\}$, and $\mathbf{x}'_q = \dot{\mathbf{x}} - \mu\nabla_{\mathbf{x}_q}\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\dot{\mathbf{x}})/||\nabla_{\mathbf{x}_q}\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\dot{\mathbf{x}})||$.

We say that μ is *valid* if and only if $\nabla_\mu \nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\mu) \leq 0$, $\mu \in \mathrm{Proj}_\mu(F_{(3.28)})$, and $\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\mathbf{x}') \leq \nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\dot{\mathbf{x}})$, i.e., the objective function is not ascending in our search direction at a stepsize of μ, and μ is part of a feasible solution to (3.28) that has a smaller objective function value than the one corresponding to $\dot{\mathbf{x}}$. The algorithm is then given as follows.

Stepsize

0. Select algorithmic parameters $\epsilon > 0$, $\epsilon_{\text{step}} > 0$ as small termination parameters, and $\bar{M}$ as a maximum stepsize. Initialize $\mu = 1$, $\text{step}^{\text{L}} = 0$ and $\text{step}^{\text{U}} = \bar{M}$, respectively. Continue to Step 1.
1. If $\text{step}^{\text{U}} - \text{step}^{\text{L}} \leq \epsilon_{\text{step}}$, then terminate with $\mu = \text{step}^{\text{L}}$. Otherwise, go to Step 2 if μ is valid, and Step 4 if μ is not valid.
2. If $\nabla_\mu \nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\mu) = 0$ and $\nabla_\mu \nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\mu)(\mu - \epsilon) < 0$, then there exists a local minimum of (2.6a) at μ. Terminate with $\mu = \text{step}^{\text{L}}$ in this case. Otherwise, go to Step 3.
3. Set $\text{step}^{\text{L}} = \mu$. If $\text{step}^{\text{U}} = \bar{M}$, then set $\mu = \min\left\{2\mu, \bar{M}\right\}$, and otherwise set $\mu = \frac{(\text{step}^{\text{U}} + \text{step}^{\text{L}})}{2}$. Go to Step 1.
4. Set $\text{step}^{\text{U}} = \mu$ and $\mu = \frac{(\text{step}^{\text{U}} + \text{step}^{\text{L}})}{2}$, and go to Step 1.

We are now ready to develop the following Monitor Location Improvement Method (MLIM), given $\hat{\mathbf{x}}$, $\hat{\mathbf{w}}$, and $\dot{\mathbf{f}}$, to iteratively adjust monitor locations.

MLIM

0. We initialize by choosing i_{min} and i_{limit} as positive integer parameters, where $i_{\text{min}} < i_{\text{limit}}$, and $\epsilon_{i_{\text{min}}} > 0$ as a termination parameter. The algorithm will attempt to improve each monitor's location until either a maximium limit of i_{limit} monitor improvement iterations is performed, or until the objective improvement over i_{min} iterations becomes less than $\epsilon_{i_{\text{min}}}$. Initialize the updated locations $\dot{\mathbf{x}} = \hat{\mathbf{x}}$. Set $q = 1$, which will represent the index of the monitor whose location is being updated.
1. Initialize the iteration count to $i = 0$, and set the current objective function value $z^{(0)}$ to $\nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\dot{\mathbf{x}})$. Go to Step 2.
2. If $i = i_{\text{limit}}$, or if both $i \geq i_{\text{min}}$ and $z^{(i-i_{\text{min}})} - z^{(0)} < \epsilon_{i_{\text{min}}}$, then go to Step 4. Otherwise, proceed to Step 3.
3. Set $i = i+1$ and determine $\mu = $ Stepsize, given $(\dot{\mathbf{x}}, \hat{\mathbf{w}}, \dot{\mathbf{f}})$ and monitor q. Set $\dot{\mathbf{x}}_q = \dot{\mathbf{x}}_q - \frac{\nabla_{\mathbf{x}_q} \nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\dot{\mathbf{x}})}{\left|\left|\nabla_{\mathbf{x}_q} \nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\dot{\mathbf{x}})\right|\right|}\mu$ and $z^{(i)} = \nu_{\dot{\mathbf{f}},\hat{\mathbf{w}}}(\dot{\mathbf{x}})$, and return to Step 2.
4. If $q = |S|$, then terminate with monitor locations $\dot{\mathbf{x}}$. Otherwise, set $q = q + 1$, and return to Step 1.

We now discuss the convergence of MLIM. Note that Step 3 of MLIM, which is the bottleneck of the algorithm, iterates $|S| i_{\text{limit}}$ times. Each iteration of this step executes Stepsize, which is itself finitely convergent, proven as follows. Suppose that we do not terminate in Step 2 of Stepsize (or else finite termination is certainly achieved). The smallest decrease in the difference between step^{U} and step^{L} during an update to either value occurs if $\text{step}^{\text{U}} \leq \bar{M}$, in which case $\text{step}^{\text{U}} - \text{step}^{\text{L}}$ is reduced by at least half of its previous difference. Thus, solving $(0.5)^\gamma \bar{M} = \epsilon_{\text{step}}$ for γ, where $\bar{M}$ is the initial value of step^{U}, 0 is the initial value of step^{L}, and γ is the number of iterations of Step 1 of Stepsize, we obtain $(\text{step}^{\text{U}} - \text{step}^{\text{L}}) < \epsilon_{\text{step}}$ in at most $\gamma = \left(\ln\left(\epsilon_{\text{step}} \bar{M}^{-1}\right) / \ln(0.5)\right)$ iterations.

4. Midcolumn-Restricted Formulation. Due to the fact that evasion probabilities are computed as a nondecreasing function of distance from a monitor to an arc midpoint, our initial experiments indicate that numerous instances have optimal or near-optimal solutions in which the monitors are located at column coordinates corresponding to these arc midpoints. In an effort to obtain a formulation that exploits this tendency, we investigate a restriction that only allows monitors to be placed such that all monitors' column-axis coordinates belong to the set $\{1.5, 2.5, \dots, n_c - 0.5\}$. Visually, the monitors will belong to the "midcolumn" of the space, depicted by dotted lines in Figure 4.1. Accordingly, we will refer to the region midway between columns l and $l+1$ as midcolumn l, $\forall l = 1, \dots, n_c - 1$. To account for this restriction, we amend the previous

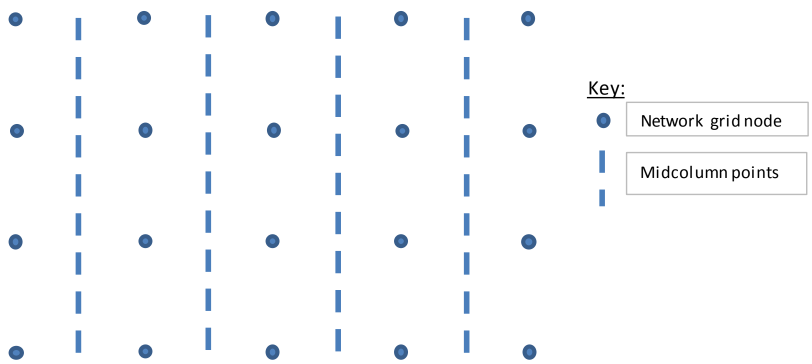

FIG. 4.1. *Illustration of midcolumn points.*

monitor placement variables to include binary variables $\mathbf{x}^{\mathrm{c}}$, where $x^{\mathrm{c}}_{s,l} = 1$ if monitor s is located at midcolumn l and $x^{\mathrm{c}}_{s,l} = 0$, otherwise. As a result, we will restrict variables in X to adhere to the following additional constraints:

$$\sum_{l=1}^{n_c-1} (l + 0.5) x^{\mathrm{c}}_{s,l} = x_{s,1} \quad \forall s \in S \tag{4.1a}$$

$$\sum_{l=1}^{n_c-1} x^{\mathrm{c}}_{s,l} = 1 \quad \forall s \in S \tag{4.1b}$$

$$x^{\mathrm{c}}_{s,l} \in \{0, 1\} \quad \forall s \in S,\ l = 1, \dots, n_c - 1. \tag{4.1c}$$

The algorithm for the midcolumn-restricted version is the same one described in § 3, except for the incorporation of $\mathbf{x}^{\mathrm{c}}$ within the updated set X used in all formulations solved in the algorithm. Also, the gradients used in the improvement step from § 3.2 are only computed with respect to $x_{s,2}$-variables, with all $x_{s,1}$-variables remaining fixed.

The following theorem establishes a bound on the absolute optimality gap between the optimal objective function value to the midcolumn-restricted version of (2.5) and that for problem (2.5) itself.

THEOREM 4.1. *Let* $(\mathbf{x}^*, \mathbf{f}^*)$ *and* $(\mathbf{x}^{*,m}, \mathbf{f}^{*,m})$ *be optimal to problems* (2.5) *and the midcolumn-restricted version of* (2.5), *respectively. Then, defining* $Q = \lfloor (n_c - 1)\left(R + 0.5L(n_c - 1)^{-1}\right) L^{-1} \rfloor$ *we have*

$$G(\mathbf{x}^*, \mathbf{f}^*) + \Delta_m \geq G(\mathbf{x}^{*,m}, \mathbf{f}^{*,m}) \geq G(\mathbf{x}^*, \mathbf{f}^*), \tag{4.2}$$

where

$$\Delta_m = \sum_{s\in S}\left[\psi(H) - \psi\left(H + \frac{L}{2(n_c-1)}\right) + \sum_{q=1}^{Q}\left(\psi\left(\frac{qL}{n_c-1}\right) - \psi\left(\frac{(q-0.5)L}{n_c-1}\right)\right)\right]. \quad (4.3)$$

Proof. The last inequality of (4.2) follows from the fact that the midcolumn-restricted version of (2.5) is a restriction of (2.5). To show the first inequality, consider an optimal solution $(\mathbf{x}^*, \mathbf{f}^*)$ to (2.5), and suppose that we shift each monitor $s \in S$ from location $\mathbf{x}^*_s$ to its nearest midcolumn location $\mathbf{x}^{\mathrm{m}}_s$, breaking ties arbitrarily. Suppose $\mathbf{f}^{\mathrm{m}}$ is an optimal adversary path corresponding to $\mathbf{x}^{\mathrm{m}}$. Our proof demonstrates that the first inequality of (4.2) holds even if $G(\mathbf{x}^{*,\mathrm{m}}, \mathbf{f}^{*,\mathrm{m}})$ is replaced with $G(\mathbf{x}^{\mathrm{m}}, \mathbf{f}^{\mathrm{m}})$, which establishes the claim. We have

$$\begin{aligned} G(\mathbf{x}^{*,\mathrm{m}}, \mathbf{f}^{*,\mathrm{m}}) - G(\mathbf{x}^*, \mathbf{f}^*) &\le G(\mathbf{x}^{\mathrm{m}}, \mathbf{f}^{\mathrm{m}}) - G(\mathbf{x}^*, \mathbf{f}^*) && (4.4)\\ &\le G(\mathbf{x}^{\mathrm{m}}, \mathbf{f}^{\mathrm{m}}) - G(\mathbf{x}^*, \mathbf{f}^{\mathrm{m}}) && (4.5)\\ &= \sum_{s\in S}\sum_{i\in I\setminus\{n_c\}}\sum_{j\in J}\sum_{k\in J}\left[\psi\left(||\Upsilon(\mathbf{x}^{\mathrm{m}}_s - \mathbf{c}_{ijk})||\right) - \psi\left(||\Upsilon(\mathbf{x}^*_s - \mathbf{c}_{ijk})||\right)\right] f^{\mathrm{m}}_{ijk}, && (4.6) \end{aligned}$$

where (4.5) follows because $G(\mathbf{x}^*, \mathbf{f}^*) \ge G(\mathbf{x}^*, \mathbf{f}^{\mathrm{m}})$ by definition of $\mathbf{x}^*$ and $\mathbf{f}^*$.

Because we employ the ℓ_1-norm in this paper, each monitor shift affects the detection probabilities of at most Q arcs on either side of the monitor affected by its detection radius; thus, at most $2Q+1$ arcs, indexed by $q \in \{-Q, \ldots, 0, \ldots, Q\}$, are affected in total. Define $(ijk)_{sq}$, $\forall s \in S$, $q \in \{-Q, \ldots, 0, \ldots, Q\}$, as the arc midpoint located q midcolumns in the opposite direction of the shift of monitor s, where $f^{\mathrm{m}}_{(ijk)_{sq}} = 1$. For instance, if for monitor $s \in S$ we have $x^*_{s,1} = 3.8$, the shift is to the left. Hence, $q = 0$ refers to an arc with column midpoint 3.5, $q = 1$ refers to an arc with column midpoint 4.5, and $q = -1$ refers to an arc with column midpoint 2.5, i.e., in the same direction of the shift.

We now bound the right-hand side of (4.6) as follows.

$$\begin{aligned} &\sum_{s\in S}\sum_{i\in I\setminus\{n_c\}}\sum_{j\in J}\sum_{k\in J}\left[\psi\left(||\Upsilon(\mathbf{x}^{\mathrm{m}}_s - \mathbf{c}_{ijk})||\right) - \psi\left(||\Upsilon(\mathbf{x}^*_s - \mathbf{c}_{ijk})||\right)\right] f^{\mathrm{m}}_{ijk}\\ &\quad = \sum_{s\in S}\sum_{q=-Q}^{Q}\left[\psi\left(\left|\left|\Upsilon(\mathbf{x}^{\mathrm{m}}_s - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right) - \psi\left(\left|\left|\Upsilon(\mathbf{x}^*_s - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right)\right] && (4.7)\\ &\quad \le \sum_{s\in S}\sum_{q=0}^{Q}\left[\psi\left(\left|\left|\Upsilon(\mathbf{x}^{\mathrm{m}}_s - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right) - \psi\left(\left|\left|\Upsilon(\mathbf{x}^*_s - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right)\right], && (4.8) \end{aligned}$$

where we take $\psi\left(\left|\left|\Upsilon(\mathbf{x}_s - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right) = 0$ for any q coresponding to a column midpoint outside the range $[1.5, n_c - 0.5]$. Note that (4.8) follows due to monotonicity of $\psi(\cdot)$ and the fact that

$$\psi\left(\left|\left|\Upsilon(\mathbf{x}^{\mathrm{m}}_s - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right) - \psi\left(\left|\left|\Upsilon(\mathbf{x}^*_s - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right) \le 0,\ \forall q = -Q, \ldots, -1,\ s \in S.$$

Using Lemmas 3.2 and 3.3, we can bound the right-hand side of (4.8) as follows.

$$
\begin{aligned}
&\sum_{s\in S}\sum_{q=0}^{Q}\left[\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{m}}-\mathbf{c}_{(ijk)_{sq}})\right|\right|\right)-\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{sq}})\right|\right|\right)\right]\\
&\quad\le\sum_{s\in S}\Bigg[\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{m}}-\mathbf{c}_{(ijk)_{s0}})\right|\right|\right)-\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{s0}})\right|\right|\right)\\
&\qquad\qquad+\sum_{q=1}^{Q}\left(\psi\left(\frac{qL}{n_c-1}\right)-\psi\left(\frac{(q-0.5)L}{n_c-1}\right)\right)\Bigg] && (4.9)\\
&\quad\le\sum_{s\in S}\left[\psi(H)-\psi\left(H+\frac{L}{2(n_c-1)}\right)+\sum_{q=1}^{Q}\left(\psi\left(\frac{qL}{n_c-1}\right)-\psi\left(\frac{(q-0.5)L}{n_c-1}\right)\right)\right], && (4.10)
\end{aligned}
$$

where (4.9) follows from Lemma 3.3 with $\beta(\cdot)=\psi(\cdot)$, $d_1=(q-0.5)L(n_c-1)^{-1}$, and $d_2=0.5L(n_c-1)^{-1}$, and the fact that $\left|\left|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{sq}})\right|\right|\ge d_1$ and $\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{m}}-\mathbf{c}_{(ijk)_{sq}})\right|\right|-\left|\left|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{sq}})\right|\right|=|\Upsilon(\mathbf{x}_s^{\mathrm{m}}-\mathbf{c}_{(ijk)_{sq}})_1|-|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{sq}})_1|\le d_2$, $\forall q=1,\ldots,Q$. Similarly, (4.10) follows from Lemma 3.2 with $\beta(\cdot)=\psi(\cdot)$, $d_1=H$, and $d_2=0.5L(n_c-1)^{-1}$, and the fact that $\left|\left|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{s0}})\right|\right|\le d_1+d_2$ and $\left|\left|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{s0}})\right|\right|-\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{m}}-\mathbf{c}_{(ijk)_{s0}})\right|\right|=|\Upsilon(\mathbf{x}_s^{*}-\mathbf{c}_{(ijk)_{s0}})_1|-|\Upsilon(\mathbf{x}_s^{\mathrm{m}}-\mathbf{c}_{(ijk)_{s0}})_1|\ge d_2$. Combining (4.4)–(4.10), we obtain the first inequality of (4.2). ■

By Theorem 4.1, the maximum difference in probability of evasion between the optimal location found by solving (2.5) and the midcolumn-restricted version of (2.5) is

$$e^{G_{\mathrm{m}}^{*}}-e^{G_{\mathrm{m}}^{*}-\Delta_{\mathrm{m}}}, \tag{4.11}$$

where $G_{\mathrm{m}}^{*}=G(\mathbf{x}^{*,\mathrm{m}},\mathbf{f}^{*,\mathrm{m}})$.

5. Full-Discretization Formulation. While the exact algorithm presented in § 3 for solving (2.5) and the midcolumn-restricted version of this algorithm both converge to their respective optimal solutions in a finite number of iterations, the computational time needed to solve larger problem instances is often prohibitively large. Thus, in an effort to obtain near-optimal solutions using a less computationally-intensive formulation, we restrict the feasible monitor locations to a subset of the midcolumn-restricted set of locations X from § 4. Define a binary variable vector $\boldsymbol{\chi}$ such that $\chi_{s,lr}=1$ if monitor $s\in S$ is located at discrete location $r\in\{1,\ldots,N\}$ of midcolumn $l\in\{1,\ldots,n_c-1\}$; otherwise, $\chi_{s,lr}=0$. The actual coordinate locations $\mathbf{x}^{\mathrm{FDF}}$ will be such that

$$\mathbf{x}_s^{\mathrm{FDF}}=\sum_{l=1}^{n_c-1}\sum_{r=1}^{N}\chi_{s,lr}\boldsymbol{\omega}_{lr}\quad\forall s\in S, \tag{5.1}$$

where $\boldsymbol{\omega}_{lr} = \big(l + 0.5, (r-1)(n_r - 1)(N-1)^{-1} + 1\big)$. We formulate the fully-discretized formulation, denoted as problem FDF, as follows.

$$\min \quad y_1 \tag{5.2a}$$

$$\text{s.t.} \quad \mathbf{A}_{ijk}^T \mathbf{y} \geq \sum_{s \in S} \rho_{sijk} \quad \forall i \in I \backslash \{n_c\},\ j,\ k \in J \tag{5.2b}$$

$$\rho_{sijk} \geq \sum_{l=1}^{n_c - 1} \sum_{r=1}^{N} \tau_{s,lr,ijk} \chi_{s,lr} \quad \forall s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J \tag{5.2c}$$

$$\sum_{l=1}^{n_c - 1} \sum_{r=1}^{N} \chi_{s,lr} = 1 \quad \forall s \in S \tag{5.2d}$$

$$\chi_{s,lr} \in \{0, 1\} \quad \forall s \in S,\ r = 1, \ldots, N, \tag{5.2e}$$

where $\mathbf{A}_{ijk}^T$ is the row vector of the transpose of contraint matrix $\mathbf{A}$ corresponding to arc ijk and

$$\tau_{s,lr,ijk} = \psi\left(||\Upsilon(\boldsymbol{\omega}_{lr} - \mathbf{c}_{ijk})||\right) + \ln(v_{ijk}), \tag{5.3}$$

$\forall s \in S,\ i \in I \backslash \{n_c\},\ j,\ k \in J,\ r = 1, \ldots, N$. Constraints (5.2b) and (5.2c) are equivalent to (2.9b) by (2.7), (2.8), (5.1), and (5.3). Constraints (5.2d) and (5.2e) restrict each monitor to be located at exactly one midcolumn location.

We establish in Theorem 5.1 a bound on the maximum difference between an optimal objective function value of the midcolumn-restricted version of (2.5) and that of FDF.

Theorem 5.1. *Let* $(\mathbf{x}^{*,m}, \mathbf{x}^{*,c}, \mathbf{f}^{*,m})$ *and* $y_1^{*,FDF}$ *be optimal to the midcolumn-restricted version of* (2.5) *and FDF, respectively. Then, defining* $\bar{Q} = \{-Q_{md}, \ldots, 0, \ldots, Q_{md}\}$ *and* $Q_{md} = \left\lfloor \frac{(n_c - 1)R}{L} \right\rfloor$*, we have*

$$G(\mathbf{x}^{*,m}, \mathbf{f}^{*,m}) + \Delta_{FDF} \geq y_1^{*,FDF} \geq G(\mathbf{x}^{*,m}, \mathbf{f}^{*,m}), \tag{5.4}$$

where

$$\Delta_{FDF} = \psi\left(\frac{H}{2(N-1)}\right) - \psi(0) + \sum_{q \in \bar{Q} \backslash \{0\}} \left(\psi\left(\frac{H}{2(N-1)} + \frac{qL}{n_c - 1}\right) - \psi\left(\frac{qL}{n_c - 1}\right)\right). \tag{5.5}$$

.

Proof. The last inequality of (5.4) follows from the fact that FDF is a restriction of the midcolumn-restricted version of (2.5). To show the first inequality, consider an optimal solution $(\mathbf{x}^{*,\mathrm{m}}, \mathbf{x}^{*,\mathrm{c}}, \mathbf{f}^{*,\mathrm{m}})$ to the midcolumn-restricted version of (2.5) and suppose that we shift each monitor $s \in S$ from location $\mathbf{x}_s^{*,\mathrm{m}}$ to its nearest location $\mathbf{x}_s^{\mathrm{FDF}}$ pertaining to a feasible solution $\boldsymbol{\chi}$ to FDF, breaking ties arbitrarily, with objective function value y_1^{FDF}. Suppose $\mathbf{f}^{\mathrm{FDF}}$ is the optimal adversary path corresponding to $\mathbf{x}^{\mathrm{FDF}}$. Our proof validates that the first inequality of (5.4) holds even if $y_1^{*,\mathrm{FDF}}$ is replaced with y_1^{FDF}, which establishes the claim. We have

$$\begin{aligned} y_1^{*,\mathrm{FDF}} - G(\mathbf{x}^{*,\mathrm{m}}, \mathbf{f}^{*,\mathrm{m}}) &\leq y_1^{\mathrm{FDF}} - G(\mathbf{x}^{*,\mathrm{m}}, \mathbf{f}^{*,\mathrm{m}}) && (5.6)\\ &\leq y_1^{\mathrm{FDF}} - G(\mathbf{x}^{*,\mathrm{m}}, \mathbf{f}^{\mathrm{FDF}}) && (5.7)\\ &= \sum_{s \in S} \sum_{i \in I \backslash \{n_c\}} \sum_{j \in J} \sum_{k \in J} \left(\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{ijk})\right|\right|\right) - \psi\left(||\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{ijk})||\right)\right) f_{ijk}^{\mathrm{FDF}}, && (5.8) \end{aligned}$$

where (5.7) follows because $G(\mathbf{x}^{*,\mathrm{m}}, \mathbf{f}^{\mathrm{FDF}}) \leq G(\mathbf{x}^{*,\mathrm{m}}, \mathbf{f}^{*,\mathrm{m}})$ by definition of $\mathbf{x}^{*,\mathrm{m}}$ and $\mathbf{f}^{\mathrm{FDF}}$.

For this proof, since each monitor shift is within the same column, we define $(ijk)_{sq}$ the same as in Theorem 4.1 with the exception that positive (negative) q refers to an arc with column midpoint q columns to the right (left) of the monitor. Because we employ the ℓ_1-norm in this paper, each monitor shift affects the detection probabilities of at most $2Q_{\mathrm{md}} + 1$ arcs. Using Lemma 3.3, we now bound the right-hand side of (5.8) as follows.

$$\sum_{s\in S}\sum_{i\in I\setminus\{n_c\}}\sum_{j\in J}\sum_{k\in J}\left(\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{ijk})\right|\right|\right) - \psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{ijk})\right|\right|\right)\right) f_{ijk}^{\mathrm{FDF}}$$

$$= \sum_{s\in S}\sum_{q\in\bar{Q}}\left(\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right) - \psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{sq}})\right|\right|\right)\right) \tag{5.9}$$

$$\leq \sum_{s\in S}\Bigg(\psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{(ijk)_{s0}})\right|\right|\right) - \psi\left(\left|\left|\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{s0}})\right|\right|\right) + \sum_{q\in\bar{Q}\setminus\{0\}}\left(\psi\left(\frac{H}{2(N-1)} + \frac{qL}{n_c-1}\right) - \psi\left(\frac{qL}{n_c-1}\right)\right)\Bigg) \tag{5.10}$$

$$\leq \sum_{s\in S}\left(\psi\left(\frac{H}{2(N-1)}\right) - \psi(0) + \sum_{q\in\bar{Q}\setminus\{0\}}\left(\psi\left(\frac{H}{2(N-1)} + \frac{qL}{n_c-1}\right) - \psi\left(\frac{qL}{n_c-1}\right)\right)\right), \tag{5.11}$$

where (5.10) follows from Lemma 3.3 with $\beta(\cdot) = \psi(\cdot)$, $d_1 = qL(n_c-1)^{-1}$, and $d_2 = 0.5H(N-1)^{-1}$, and the fact that $\left|\left|\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{sq}})\right|\right| \geq d_1$ and $\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{(ijk)_{sq}})\right|\right| - \left|\left|\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{sq}})\right|\right| = |\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{(ijk)_{sq}})_2|$ $- |\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{sq}})_2| \leq d_2$, $\forall q \in \bar{Q}\setminus\{0\}$. Similarly, inequality (5.11) follows from Lemma 3.3 with $\beta(\cdot) = \psi(\cdot)$, $d_1 = 0$, and $d_2 = 0.5H(N-1)^{-1}$, and the fact that $\left|\left|\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{s0}})\right|\right| \geq d_1$ and $\left|\left|\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{(ijk)_{s0}})\right|\right| - \left|\left|\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{s0}})\right|\right| = |\Upsilon(\mathbf{x}_s^{\mathrm{FDF}} - \mathbf{c}_{(ijk)_{s0}})_2| - |\Upsilon(\mathbf{x}_s^{*,\mathrm{m}} - \mathbf{c}_{(ijk)_{s0}})_2| \leq d_2$. Combining (5.6)–(5.11), we obtain the first inequality of (5.4).

■

By Theorem 5.1, the maximum difference in evasion probability between the optimal monitor locations found by solving the midcolumn-restricted version of (2.5) and monitor locations found by solving FDF is

$$e^{\mathrm{FDF}^*} - e^{\mathrm{FDF}^* - \Delta_{\mathrm{FDF}}}, \tag{5.12}$$

where FDF^* is the optimal objective function value of (5.2).

6. Computational Results. In this section, we compare the efficacy of ExactAlg, MidcolAlg, and FDF, along with the quality of solutions obtained by these algorithms. All optimization problems are solved with CPLEX 12.3 using Concert 2.9 Technology on an IBM System x3650 with two Intel E5640 Xeon processors and 24 GB memory, and are implemented using C++. Each ExactAlg and MidcolAlg instance is initialized with $|B_{sijk}| = 2$, and the algorithms are allowed to run until either reaching a termination gap of 0.01 or a computational time of 3600 seconds, whichever comes first.

To evaluate the effectiveness of our proposed solution methods, we generate a set of test instances by varying parameters n_c, n_r, $|S|$, and R. We set $L = 1000$ and $H = 1000$ for all instances. We randomly generate a set of arc factor (AF) vectors, denoted by $\mathbf{v}^a$, $\mathbf{v}^b$, and $\mathbf{v}^c$, and label them as a, b, and c, respectively. Each $\mathbf{v}$-vector is a list of 22,275 elements, representing the largest number of v_{ijk}-parameters needed over all instances tested. The first $n_c n_r^2$ elements from each **v**-vector serves as the arc factors for that instance, where only three arc factor vectors were used to better isolate the impact of the varied parameters on the solution methods. Tables that omit a column for AF include the average output statistics over a, b, and c for each instance (only including in the averages those instances terminating in under 3600 seconds), where exponents on output statistics denote the number of these AF instances unsolved after 3600 seconds, and a "{3}" denotes the instance is unsolved for all three AF vectors.

TABLE 6.1
MLIM *Effectiveness in ExactAlg;* $n_c = \{4, 7, 10\}$, $n_r = \{5, 7\}$, $|S| = 2$, $R = \{100, 1.25L/(n_c - 1)\}$, *and* $\delta_p = \{0.01, 0.25\}$.

				No MLIM		MLIM	
n_c	n_r	R	δ_p	Iterations	Time	Iterations	Time
4	5	100	0.25	6	6	6	12
4	5	100	0.01	27	139	27	116
4	5	417	0.25	12	110	12	129
4	5	417	0.01	$43^{\{1\}}$	$679^{\{1\}}$	$43^{\{1\}}$	$556^{\{1\}}$
4	7	100	0.25	3	19	3	20
4	7	100	0.01	8	104	4	48
4	7	417	0.25	15	212	14	229
4	7	417	0.01	$36^{\{1\}}$	$2252^{\{1\}}$	$35^{\{1\}}$	$1712^{\{1\}}$
7	5	100	0.25	3	33	3	30
7	5	100	0.01	11	281	11	269
7	5	208	0.25	3	34	3	34
7	5	208	0.01	20	781	18	699
7	7	100	0.25	8	346	3	129
7	7	100	0.01	$6^{\{1\}}$	$1199^{\{1\}}$	4	916
7	7	208	0.25	8	416	6	365
7	7	208	0.01	{3}	{3}	{3}	{3}
10	5	100	0.25	3	96	2	111
10	5	100	0.01	$3^{\{1\}}$	$204^{\{1\}}$	$2^{\{1\}}$	$138^{\{1\}}$
10	5	139	0.25	3	211	3	212
10	5	139	0.01	8	1176	7	1045
10	7	100	0.25	4	613	3	507
10	7	100	0.01	{3}	{3}	{3}	{3}
10	7	139	0.25	6	1310	6	1100
10	7	139	0.01	{3}	{3}	{3}	{3}

In Table 6.1, we show the effectiveness of incorporating the improvement step, MLIM, within ExactAlg. The column labeled **Time** refers to the CPU time (in seconds) required to solve the instances, and **Iterations** refers to the number of ExactAlg iterations reached before the algorithm terminated. For each (n_c, n_r, δ_p)-instance of this table, we tested detection radii of 100 units and $1.25L/(n_c - 1)$ units (i.e., a length resulting in a detection radius spanning at least two column midpoints), and fixed $|S| = 2$. Incorporating MLIM in these instances never substantially increases computational time, and reduces computational time by at least 10% in 12 of the 24 instances, and by at least 50% in two instances. (We found similar results for MidcolAlg.) Thus, we utilize MLIM within ExactAlg and MidcolAlg in the remaining computational experiments.

Table 6.2 compares the solution quality obtained by our three approaches, where Column **BndsGap** gives

TABLE 6.2

Comparison of Solution Methods with $n_c = \{10, 15, 20\}$, $n_r = 10$, $|S| = \{2, 4\}$, $R = 100$, $\delta_p = 0.75$, and $N = 10$.

			ExactAlg	MidcolAlg			FDF		
n_c	$\|S\|$	AF	Time	Time	BndsGap	TheoGap	Time	BndsGap	TheoGap
10	2	a	80	109	0.004	0.066	1	0.027	0.126
10	2	b	93	98	0.001	0.071	1	0.001	0.102
10	2	c	99	117	0	0.068	2	0.005	0.107
10	4	a	1740	834	0.002	0.035	3	0.012	0.064
10	4	b	1083	388	0.001	0.042	3	0.015	0.074
10	4	c	3337	829	0.002	0.037	2	0.007	0.063
15	2	a	205	204	0.003	0.030	6	0.005	0.085
15	2	b	432	281	0.002	0.029	4	0.010	0.086
15	2	c	253	284	0	0.031	6	0.006	0.089
15	4	a	3600	3600	0.027	0.010	9	0.029	0.024
15	4	b	3600	3600	0.023	0.007	6	0.025	0.020
15	4	c	3600	3600	0.022	0.008	7	0.025	0.021
20	2	a	1354	1129	0.001	0.022	12	0.005	0.066
20	2	b	1899	863	0.004	0.020	7	0.006	0.060
20	2	c	405	696	0.003	0.019	10	0.012	0.059
20	4	a	3600	3600	0.009	0.006	19	0.009	0.008
20	4	b	3600	3600	0.008	0.003	13	0.008	0.007
20	4	c	3600	3600	0.006	0.004	19	0.007	0.006

the difference between the final upper-bound value of MidcolAlg (or FDF) and the final lower-bound value of ExactAlg. Column **TheoGap** reports the maximum gap that can exist between the final upper bound for MidcolAlg (or optimal objective function value of FDF) and that of (2.1), which is given by the worst-case gap values given in (4.11) (or the combination of (4.11) and (5.12) for FDF). MidcolAlg requires an average of 286 seconds less computational time than ExactAlg. Importantly, for the instances where both ExactAlg and MidcolAlg terminate in under 3600 seconds, the **BndsGap** is no more than 0.002 over these instances, which shows that all of our instances have optimal or near-optimal solutions at midcolumn locations. Among instances converging within the allotted time, MidcolAlg requires an average of 486 seconds to converge, while FDF takes only 5 seconds. The maximum probability difference between the optimal objective function value for FDF and that of (2.1) is no more than 0.029 over all 18 instances of this table, and less than 0.01 — the termination gap for ExactAlg — in 11 instances. Also, note that the computational time of ExactAlg tends to decrease as the monitor dampening parameter (δ_p) increases from 0.01 to 0.25 (with similar results for MidcolAlg). This result is logical because for any monitor-arc combination $sijk$, the piecewise-linear concave function that underestimates the natural logarithm curve on the probability interval $[\delta_p, 1]$ becomes tighter as δ_p increases.

The computational times of the smaller instances in Table 6.2 show that ExactAlg is efficient in solving these instances. Hence, ExactAlg is the most appropriate algorithm of the three for these instances. However, as the instance sizes grow larger, FDF becomes the most appropriate method: Its computational times are very small compared to that of ExactAlg, and its **TheoGap** values decrease as the size of the grid and number of monitors placed increases.

TABLE 6.3

Effect of N on FDF Solution with $n_c = \{20, 40\}$, $n_r = \{10, 15\}$, $|S| = \{2, 4\}$, $R = 200$, and $\delta_p = 0.75$.

n_c	n_r	$\|S\|$	N	Time	TheoGap	n_c	n_r	$\|S\|$	N	Time	TheoGap
20	10	2	6	5	0.06325	40	10	2	6	24	0.00504
20	10	2	10	7	0.05044	40	10	2	10	45	0.00435
20	10	4	6	11	0.00492	40	10	4	6	106	0.00003
20	10	4	10	19	0.00385	40	10	4	10	143	0.00002
20	15	2	6	15	0.15519	40	15	2	6	97	0.03242
20	15	2	10	25	0.12877	40	15	2	10	210	0.02949
20	15	4	6	50	0.02741	40	15	4	6	336	0.00083
20	15	4	10	93	0.02378	40	15	4	10	1002	0.00070

TABLE 6.4

FDF Solution Quality on Large Instances with $n_c = \{80, 100\}$, $n_r = 15$, $|S| = \{2, 4\}$, $R = \{100, 200\}$, $\delta_p = \{0.75, 0.95\}$, and N= 6.

n_c	$\|S\|$	R	δ_p	Time	TheoGap	n_c	$\|S\|$	R	δ_p	Time	TheoGap
80	2	100	0.95	499	0.0025264	100	2	100	0.95	870	0.0007059
80	2	100	0.75	851	0.0029472	100	2	100	0.75	1331{1}	0.0006585{1}
80	2	200	0.95	539	0.0023656	100	2	200	0.95	863	0.0006130
80	2	200	0.75	1189	0.0012269	100	2	200	0.75	1703{1}	0.0002057{1}
80	4	100	0.95	1208	0.0000163	100	4	100	0.95	2519	0.0000011
80	4	100	0.75	1756{1}	0.0000068{1}	100	4	100	0.75	{3}	{3}
80	4	200	0.95	1315	0.0000104	100	4	200	0.95	2706	0.0000006
80	4	200	0.75	2362{2}	0.0000003{2}	100	4	200	0.75	{3}	{3}

Tables 6.3 and 6.4 report the solution quality obtained by FDF on larger instances, which are too difficult to solve within 3600 seconds using ExactAlg or MidcolAlg. Table 6.3 illustrates that the solution time increases not only with an increase in grid size and number of monitors placed, but also with an increase in the number of feasible locations per column (N). For Table 6.3 instances, the average solution time grows from 81 to 193 when N jumps from 6 to 10. Furthermore, this table also shows that the average **TheoGap** decreases from 0.036 to 0.030 when N increases from 6 to 10. These results are intuitive because an increase in N leads to an increase in the number of binary variables in FDF. This increase in problem size tends to increase the time required to solve each lower-bound problem, and also leads to a smaller maximum possible gap between the optimal objective function value of (2.1) and that of FDF.

Table 6.4 includes the largest instances solved within 3600 seconds using the given memory limits. Among the instances solving within 3600 seconds, the average **TheoGap** is 0.001, and never exceeds 0.003. Note that in Tables 6.3 and 6.4, our algorithms do not utilize MLIM because the very small **TheoGap** values show that MLIM is not necessary to improve the monitor positions.

We also observe that as the detection radius increases, **TheoGap** decreases for FDF, but at a much greater rate than the solution time increases. For the results in Table 6.4, as R increases from 100 to 200, we observe a 64% decrease in **TheoGap** with only an 18% increase in computational time. To further express the effectiveness of FDF on achieving high-quality solutions, note that FDF **TheoGap** values computed for all Table 6.2 instances is as large as 0.126, even though FDF **BndsGap** is no more than 0.029. This result implies that although our theoretical bound on the gap between the optimal objective function value of FDF and that of (2.1) is small for instances of the largest grid size, it is conservative and may still be significantly greater than the true gap.

7. Conclusion. In this paper, we prescribed alternative solution methods for the problem of efficiently placing stationary monitors in an area of interest to minimize an adversary's maximum evasion probability. We modeled the problem as a two-stage MINLP in a network setting. One solution method, ExactAlg, is a finitely convergent algorithm that determines optimal monitor locations over the continuous space. Computational experiments demonstrate that ExactAlg converges to optimality on instances involving grids of relatively small size within reasonable computational limits. Another solution method, MidcolAlg, utilizes ExactAlg but restricts monitors to midcolumn locations, exploiting the result that many instances have optimal solutions existing at midcolumn locations. This midcolumn-restricted version of the algorithm reduces solution time, while still obtaining provably near-optimal solutions. In an effort to further decrease solution time, while still obtaining a near-optimal solution, we implement a third approach (FDF), which solves a mixed-integer linear program that fully discretizes the feasible monitor locations. For instances taking place on larger grids, we obtain provably near-optimal solutions with FDF in well under one hour.

Because adaptability of a model to different scenarios is crucial to its success in real-world implementation, one important note is that while we model the evasion probability as a concave function of the distance from a monitor to an arc, any nondecreasing function can easily substituted without altering our solution schemes. One noteworthy extension involves developing a method that iteratively determines the set of feasible monitor locations based on a prior solution rather than initially fixing this set. This change may further improve the effectiveness of our approach by allocating binary variables more efficiently. Additionally, a major reason that the exact algorithm was not able to solve larger problem instances involves the incorporation of big-M constants in the constraints, which weaken the mixed-integer programming formulations. Finding a relaxation more suitable to integer programming solution approaches is another promising future research direction.

REFERENCES

[1] *Containing risk*. `http://americancityandcounty.com`. Accessed September 2012.

[2] *Homeland security: Summary of challenges faced in targeting oceangoing cargo containers for inspection*, United States Government Accountability Office, GAO-04-557T (2004).

[3] *Maritime security: New structures have improved information sharing, but security clearance processing requires further attention.*, United States Government Accountability Office, GAO-05-394 (2005).

[4] M. P. Atkinson, Z. Cao, and L. M. Wein, *Optimal stopping analysis of a radiation detection system to protect cities from a nuclear terrorist attack*, Risk Analysis, 28 (2008), pp. 353–371.

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

[6] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, Wiley, Hoboken, NJ, fourth ed., 2009.

[7] A. Ben-Tal and A. Nemirovski, *Robust optimization – methodology and applications*, Mathematical Programming, 92 (2002), pp. 453–480.

[8] D. Bertsimas, D. Brown, and C. Caramanis, *Theory and applications of robust optimization*, SIAM Review, 53 (2011), pp. 464–501.

[9] D. Bienstock and A. Verma, *The $n-k$ problem in power grids: New models, formulations, and numerical experiments*, SIAM Journal on Optimization, 20 (2010), pp. 2352–2380.

[10] G. G. Brown, W. M. Carlyle, J. Salmerón, and R. K. Wood, *Defending critical infrastructure*, Interfaces, 36 (2006), pp. 530–544.

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

[12] R. Hemmecke, R. Schultz, and D. L. Woodruff, *Interdicting stochastic networks*, in Network Interdiction and Stochastic Integer Programming, D. L. Woodruff, ed., Kluwer Academic Publishers, Boston, MA, 2003, pp. 71–80.

[13] E. Israeli and R. K. Wood, *Shortest-path network interdiction*, Networks, 40 (2002), pp. 97–111.

[14] G. P. McCormick, *Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems*, Mathematical Programming, 10 (1976), pp. 147–175.

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

[16] A. Pinar, J. Meza, V. Donde, and B. Lesieutre, *Optimization strategies for the vulnerability analysis of the electric power grid*, SIAM Journal on Optimization, 20 (2010), pp. 1786–1810.

[17] J. C. Smith, M. Prince, and J. Geunes, *Modern network interdiction problems and algorithms*, in Handbook of Combinatorial Optimization, P. M. Pardalos, D.-Z. Du, and R. Graham, eds., Springer, New York, 2nd ed., 2012 (Forthcoming).

[18] H. von Stackelberg, *The Theory of the Market Economy*, William Hodge and Co., London, U.K., 1952.

[19] A. Washburn and R. K. Wood, *Two-person zero-sum games for network interdiction*, Operations Research, 43 (1995), pp. 243–251.

[20] L. M. Wein and M. P. Atkinson, *The last line of defense: Designing radiation detection-interdiction systems to protect cities from a nuclear terrorist attack*, IEEE Transactions on Nuclear Science, 54 (2007), pp. 654–669.

[21] ———, *Spatial queueing analysis of an interdiction system to protect cities from a nuclear terrorist attack*, Operations Research, 56 (2008), pp. 247–254.

[22] W. E. Wilhelm and E. I. Gokce, *Branch-and-price decomposition to design a surveillance system for port and waterway security*, IEEE Transactions on Automation Science and Engineering, 7 (2010), pp. 316–325.