Game-theoretic methods for locating camera towers and scheduling surveillance

Javier Salmeron* and R. Kevin Wood

Operations Research Department,
Naval Postgraduate School,
Monterey, CA, 93943, USA
Email: jsalmero@nps.edu
Email: kwood@nps.edu
*Corresponding author

Abstract: We develop techniques to optimise the locations and surveillance scheduling of tower-mounted camera systems used by a military force in an urban setting. Using a game-theoretic foundation, we seek to minimise expected damage from attacks or other adversarial events (e.g., emplacements of improvised explosive devices). Assuming that at most one camera may surveil a single point of interest (POI) at any time, a mixed-integer program uses an additive-probability model to optimise the placement of towers, while allocating 'aggregate, normalised surveillance time' between cameras and POIs. Linear-programming-based column generation then creates a probability distribution for camera-to-POI assignments to define implementable schedules. We prove that such schedules must exist, making the additive probability model exact. Computational examples on realistically sized problems produce high-quality solutions quickly, with quality suffering only when the number of cameras available nears the number of POIs to be surveilled. We show that an alternative game-theoretic model may produce better solutions when such a situation arises.

Keywords: camera tower; surveillance; column generation; integer programming; randomised algorithm.

Reference to this paper should be made as follows: Salmeron, J. and Wood, R.K. (2017) 'Game-theoretic methods for locating camera towers and scheduling surveillance', *Int. J. Simulation and Process Modelling*, Vol. 12, Nos. 3/4, pp.264–273.

Biographical notes: Javier Salmeron is an Associate Professor in the Operations Research Department at the Naval Postgraduate School. His research focuses in the area of mathematical modelling and optimisation, and its applications in civilian and military problems.

R. Kevin Wood is a Distinguished Professor Emeritus of Operations Research at the Naval Postgraduate School. At NPS from 1982 to 2015, he taught courses in optimisation and studied problems in network reliability; graph algorithms; linear, integer and stochastic programming; game theory; and system interdiction and defence.

This paper is a revised and expanded version of a paper entitled 'Optimally locating surveillance assets in urban areas' presented at The 5th International Defense and Homeland Security Simulation Workshop (I3M-DHSS), Bergeggi, Italy, 21–23 September 2015.

1 Introduction

This paper develops and tests models for optimally locating a set of camera towers used for surveillance of an urban area during a military conflict. We build on Salmeron and Wood (2015) who develop models that minimise the damage caused by undetected random or adversarial attacks at surveilled 'point of interests' (POIs). We overcome the key limitation of that work, the assumption of a fixed assignment of cameras to POIs, by also scheduling 'surveillance sessions' with various camera-to-POI assignments.

Using Salmeron and Wood's (2015) adversarial paradigm, we first develop a model to optimise tower locations using an objective that evaluates aggregate, normalised camera-to-POI assignment times. We go on to show that:

a. a post-processing algorithm can produce an implementable session schedule whose long-term coverage matches those aggregate values

b. the complete procedure can be carried out efficiently using standard software and hardware.

In essence, the camera schedule is a probability distribution over surveillance sessions, which is to be implemented as a Monte Carlo simulation. Thus, our paper concerns itself with efficient simulation, but in a less standard fashion than do papers covering, say, efficient 'mechanics' of a simulation (e.g., Gillespie, 2001; Chen and Yucesan, 2005), efficient statistical tests to optimise the design of a simulated system (e.g., Nelson et al., 2001), or efficient optimisation of a simulated system (e.g., Fu et al., 2005; Jafferali et al., 2005).

Examples of tower-based surveillance systems include 'ground-based operational surveillance system' (GBOSS) and base expeditionary targeting and surveillance systems-combined (BETSS-C). These systems have been used in Iraq and Afghanistan by coalition forces to help thwart the emplacement of improvised explosive devices (Defense Industry Daily, 2015). They can also help identify other events to which forces should respond, e.g., armed attacks, riots, sightings of suspicious vehicles; for simplicity, we refer to all such events as 'attacks'. In addition to camera towers, low-altitude aerostats (tethered blimps) could also be included in the set of modelled surveillance assets.

Murray et al. (2007) develop a model, similar to facility-location model, to identify the best subset of potential locations to place a limited number of surveillance cameras. Their objective function ignores probabilities of detection and simply tries to maximise the weighted number of POIs that can be surveilled. Using a similar physical setting, Salmeron and Wood (2015) develop several models that do account for detection probabilities, which may vary by distance between tower and POI and by other factors. However, neither of these deals with the actual scheduling of surveillance sessions. Scheduling could be critical given that the total number of POIs that should be surveilled may exceed the number of available cameras.

The current paper takes camera scheduling into account, while still beginning with the solution to a static model for optimising 'facility locations'. This model, a mixed-integer program (MIP), locates towers and determines the fraction of time that a camera at a particular location should surveil each POI. The MIP assumes a game-theoretic environment of an 'attacker-defender model' (Brown et al., 2006). In particular, a single *attacker*, who will carry out attacks over time, understands:

a the value that the *defender* places on the various POIs

b the defender's nominal probability of detecting an attack at a POI from any potential tower location.

While these worst-case assumptions may seem strong, they are appropriate for a military setting in which a central authority directs attacks (and another central authority directs defences). A model that minimises 'average damage' would require even stronger assumptions: the defender would also need to know the frequency with which attacks will occur at each POI. We do briefly develop in the body of the paper, however, an average-damage variant of our techniques, which might apply in a civilian setting.

Our tower-location model does have some similarities with certain camera-placement models which, in turn, resemble certain facility-location models. Bodor et al. (2007) address the problem of camera placement for maximum observability of moving subjects in a given area, and introduce a joint measure of observability with quality of the view. Their optimisation in terms of the 'motion statistics of a scene' resembles the optimisation of facility locations over an empirical probability distribution of customer demands. [See the survey paper on stochastic facility-location models by Snyder (2006).] Hörster and Lienhart (2006) address a problem involving both coverage and resolution of images, accounting for cost of operations and effectiveness of a set of cameras with orientation-dependent effectiveness. They develop a number of models, but at least one has a strong flavour of a deterministic facility-location model. [See the survey paper on deterministic facility-location models by Owen and Daskin (1998).] In particular, variables determine whether a camera is placed at particular location and with a particular orientation (cf. facility operations), and with rewards that depend on whether a particular POI is covered by (cf. served by) a camera. In summary, our basic model has some similarities with certain camera-location and facility-location models, but our adversarial objective function appears to be unique.

Although our problem involves surveillance and scheduling, it shares few similarities with the topic often referred to in the literature as *surveillance scheduling*. This topic involves the coordination of one or more cameras to surveil one or more moving targets (e.g., Costello and Wang, 2005; Qureshi and Terzopoulos, 2006; Krahnstoever et al., 2008; Natarajan et al., 2012; Jänen et al., 2013). Surveillance-scheduling models assume a fixed set of camera locations, so they cannot help us with optimally locating camera towers. These models may seek to maximise a probabilistic function such as the expected number of targets recognised – superficially, this might resemble our objective, which measures expected damage from attacks – but surveillance scheduling always models a stochastic environment with no adversarial characteristics. In summary, we find no useful connections between 'surveillance scheduling' and the camera-to-POI scheduling that we require.

We design our foundational tower-location model to identify aggregate, normalised camera-to-POI surveillance times – no scheduling is involved yet – because a model that combines location and scheduling would be complicated and might solve too slowly for practical use. In effect, a solution to the tower-location model identifies marginal probabilities for a camera-scheduling distribution, a distribution we prove to exist under reasonable assumptions. Of course, we also describe a method for computing such a distribution. To establish correctness, this effort employs the theory exploited by Liu et al. (2005) to solve a 'maximal lifetime sensor-scheduling problem'. For a survey of related sensor-scheduling problems, see Section 4.2 in Wang (2011), but note that the techniques described there lack the game-theoretic foundations of our techniques.

The current paper is based on, but diverges substantially from, a conference paper, Salmeron and Wood (2015). In particular, the current paper:

a omits details of a supporting database

b emphasises an adversarial, game-theoretic environment

c adds a new model and mathematical derivation

d develops a completely new scheduling paradigm and algorithm, along with validating theoretical results

e demonstrates the computational behaviour of our models and algorithms using new software implementations and test problems.

2 A MIP for locating camera towers

This section develops a MIP to identify the best subset of potential locations to place a limited number of camera towers. The MIP assigns cameras to POIs for surveillance, and determines the fraction of time, called 'aggregate, normalised time', that each camera will be tasked with surveilling each POI to which it is assigned. Using an additive model of detection probability, the objective function seeks to minimise the maximum expected cost of damage from any attack that goes undetected. Here, 'limited number of camera towers' simply implies that a collection of towers is an expensive, limited resource that places demands on other limited resources such as operating personnel. 'Potential locations' reflect a limited number of sites that have good viewsheds and that have or could have acceptable security, for example, existing or planned command posts, respectively.

The MIP and subsequent scheduling algorithm ignore the fact that some surveillance time will be lost in switching from one camera-to-POI assignment to another. We believe this to be reasonable, because it should require only few seconds to change a camera's angle, declination and focus to shift surveillance from one POI to another, and because we expect a given assignment to be maintained for several minutes.

2.1 Minimising worst-case expected damage

Using an additive model of detection probability and assuming that all camera towers are identical, the MIP below, **MX1**, minimises the maximum (i.e., worst-case) expected damage from an undetected attack. Section 2.2 provides the game-theoretic interpretation of this model.

Indices and index sets

$\ell \in \mathcal{L}$ potential camera-tower locations

$i \in \mathcal{I}$ POIs that should be kept under surveillance

$c \in \mathcal{C}_\ell$ cameras on a tower located at $\ell \in \mathcal{L}$

$\mathcal{C}$ set of all cameras, i.e., $\mathcal{C} = \bigcup_{\ell \in \mathcal{L}} \mathcal{C}_\ell$.

Note: To simplify later exposition, we define $n_{C/T} = |C_\ell|$ for any $\ell \in \mathcal{L}$. This definition is valid because all towers are assumed to be identical.

Data

n_T number of camera towers to be installed

$n_{I/C}$ maximum number of POIs assignable to a single camera

d_i expected cost of damage that an attack at POI i will cause if the attack is not detected

$p_{\ell i}$ probability of detecting an attack at POI i from a camera on a tower at location ℓ when the camera is surveilling i.

Variables

y_ℓ 1 if a tower is located at ℓ, 0 otherwise

$\bar{y}_{ci}$ 1 if camera c has some of its time assigned to surveil POI i ('allowed assignments')

f_{ci} normalised time that camera c is assigned to surveil POI i.

Formulation

$$\mathbf{MX1}: z^*_{\mathbf{MX1}} = \min_{\mathbf{y},\bar{\mathbf{y}},\mathbf{f},z} z \tag{1}$$

subject to:

$$z \geq d_i \left(1 - \sum_{\ell \in \mathcal{L}} \sum_{c \in C_\ell} p_{\ell i} f_{ci}\right) \forall i \in I \tag{2}$$

$$\sum_{\ell \in \mathcal{L}} y_\ell = n_T \tag{3}$$

$$\sum_{i \in I} f_{ci} = y_\ell \forall \ell \in \mathcal{L}, c \in C_\ell \tag{4}$$

$$\sum_{\ell \in \mathcal{L}} \sum_{c \in C_i} f_{ci} \leq 1 \ \forall i \in I \tag{5}$$

$$f_{ci} \leq \bar{y}_{ci} \forall c \in C, i \in I \tag{6}$$

$$\sum_{i \in I} \bar{y}_{ci} \leq n_{I/C} y_\ell \ \forall \ \ell \in \mathcal{L}, c \in C_\ell \tag{7}$$

$$\sum_{c \in C_\ell} \bar{y}_{ci} \leq y_\ell \ \forall \ \ell \in \mathcal{L}, i \in I \tag{8}$$

$$y_\ell \in \{0,1\} \ \forall \ell \in \mathcal{L} \tag{9}$$

$$\bar{y}_{ci} \in \{0,1\} \ \forall c \in C, i \in I \tag{10}$$

$$f_{ci} \geq 0 \ \forall c \in C, i \in I \tag{11}$$

The objective function (1), together with constraints (2), minimises the worst case of an approximate expected damage function for undetected attacks across all POIs. Constraint (3) limits the number camera towers being located and installed to the number available n_T. Constraints (4) ensure that the total, normalised surveillance time from each camera equals the time available. Constraints (5) ensure that the total normalised time assigned to surveilling a particular POI does not exceed 1; this approximates the requirement that no more than a single camera surveil a POI at any time. Constraints (6) insist that no surveillance time be allocated from camera c to POI i unless that camera is assigned explicitly to that POI. Taking advantage of (6), constraints (7) limit the number of POIs that can be surveilled from a single tower; these constraints reflect the limited capabilities of human operators and the loss of effectiveness that would arise if a single camera had to constantly switch focus from one POI to another. Constraints (8) require that cameras on a single, installed tower be assigned to different POIs. Constraints (9) to (11) limit the variables' domains.

For notational simplicity, we assume that every POI $i \in \mathcal{I}$ can be usefully surveilled from every potential tower location $\ell \in \mathcal{L}$. In practice, the distance between a location and a POI, or the lack of a line of sight, may make surveillance of little or no value between certain pairs (i, ℓ), and the implemented model can be simplified by taking that into account.

We note that **MX1** does not distinguish one camera on a tower from another. Thus, the model may exhibit a great deal of symmetry in potential solutions, a symmetry that will grow with increasing values of $n_{C/T}$. This could reduce the efficiency of a branch-and-bound solution algorithm and necessitate the application of some symmetry-breaking constructs. With few exceptions, however, reasonably large test instances solve easily, so we do not pursue this issue.

2.2 Game-theoretic interpretation of MX1

This section presents the game-theoretic interpretation of **MX1**. In this interpretation, the tower-location model includes both the decision vector $\mathbf{y}$ for tower locations and the decision vector for camera-to-POI assignments $\bar{\mathbf{y}}$. Given those decisions, the defender and attacker engage in a simultaneous-play game with decision vectors $\mathbf{f}$ and $\boldsymbol{\varphi}$:

a with probability f_{ci}, the defender surveils POI $i \in \mathcal{I}$ with camera $c \in C^* \mid \bar{y}_{ci} = 1$

b with probability $\varphi_i, i \in \mathcal{I}$, the attacker attacks POI i.

The model has this formulation:

MMX1:

$$z^*_{\mathbf{MMX1}} \equiv \min_{\mathbf{y},\bar{\mathbf{y}}} \min_{\mathbf{f}} \max_{\boldsymbol{\varphi}} \sum_{i \in \mathcal{I}} d_i \left(1 - \sum_{\ell \in \mathcal{L}} \sum_{c \in C_\ell} p_{\ell i} f_{ci}\right) \varphi_i \tag{12}$$

subject to: (3) to (11), and

$$\sum_{i \in \mathcal{I}} \varphi_i = 1 \tag{13}$$

$$\varphi_i \geq 0 \ \forall \ i \in \mathcal{I} \tag{14}$$

MX1 formally derives from **MMX1** by fixing $\mathbf{y}$, $\bar{\mathbf{y}}$ and $\mathbf{f}$, taking the dual of the resulting linear program and then releasing the fixed variables. Given this derivation, a standard game-theory result applies to obtain the optimal attack probabilities $\boldsymbol{\varphi}^*$: these are the optimal dual variables for constraints (2) in the linear program that appears after fixing $\mathbf{y} = \mathbf{y}^*$ and $\bar{\mathbf{y}} = \bar{\mathbf{y}}^*$.

2.3 Minimising average damage

We have advocated, in the context of a military conflict, for scheduling urban surveillance to minimise maximum expected damage. The reason is that attacks in this context are likely to be guided by a central authority and, thus, the game-theoretic model of two adversaries with directly competing goals is credible. In a peacetime urban setting, however, attacks could represent independent street crimes. Historical data may then justify the modelling of attacks at each POI as a Poisson process with known rate (Braga and Bond, 2008). Such modelling yields a fixed probability of attack $\hat{\varphi}_i$ for each POI $i \in \mathcal{I}$, and we could create the following optimisation model to minimise the average damage from attacks across all POIs:

$$\mathbf{AVG1}: \min_{\mathbf{y},\bar{\mathbf{y}},\mathbf{f}} \sum_{i \in \mathcal{I}} \hat{\varphi}_i d_i \left(1 - \sum_{\ell \in \mathcal{L}} \sum_{c \in C_\ell} p_{\ell i} f_{ci}\right) \tag{15}$$

subject to: (3) to (11).

Camera towers may be irrelevant in a peacetime urban setting, however. Rather, we expect this setting to involve an existing collection of cameras on fixed mounts, with each camera aimed at a 'fixed scene'. (In some cases, the camera and scene can be adjusted in real-time to better monitor suspicious activity, but 'fixed scene' would be a reasonable modelling assumption.) Human observers monitor the scenes on video screens for 'attacks', with this monitoring limited by human ability and available screens. Because the number of cameras may exceed the number that can be monitored effectively (Neil et al., 2007), efficient scheduling of that monitoring could be important. Thus, we believe that a model related to AVG1, along with a schedule-producing algorithm like the one described next, could be useful in a peacetime urban setting. Because of our interest in military applications, however, we do not pursue this topic.

3 Scheduling

Let $(\mathbf{y}^*, \bar{\mathbf{y}}^*, \mathbf{f}^*)$ denote a solution to **MX1**, and define $C^* = \bigcup_{\ell \in \mathcal{L} | y^*_\ell = 1} C_\ell$, which is the set of usable cameras following selection of tower locations. This section shows how the normalised surveillance times represented by $\mathbf{f}^*$ can be scheduled as surveillance sessions. Such a session is simply a period of time with a valid POI-to-camera assignment; 'valid' implies that every camera $c \in C^*$ is assigned to a POI $i \in \mathcal{I}$, with at most one camera per POI. In turn, this implies that the additive probability approximation that MX1 uses in (2) is exact. To show that such scheduling is possible, we employ the theory outlined in Liu et al. (2005), but in a much condensed fashion. We create a much simpler schedule-generating algorithm, also.

Let $\mathbf{g}_k$ be a 0-1 vector that represents the k^{th} possible assignment of cameras $c \in C^*$ to POIs $i \in I$. That is, $g_{kci} = 1$ if camera $c \in C^*$ views POI $i \in I$ in assignment k, and $g_{kci} = 0$, otherwise; 'at most one camera per POI' implies that $g_{kci} \cdot g_{kc'i} = 0$ for all k, i and $c \neq c'$. (For notational simplicity below, we allow assignments k with $g_{kci} = 1$ for $c \in C^*$, even though $f^*_{ci} = 0$. In practice, we eliminate variables and constraints using this information.) Finally, let $\mathcal{K}$ denote the index set for all $\binom{|I|}{|C^*|}$ assignment vectors. Now, an optimal schedule is represented by a solution ($\boldsymbol{\alpha}^{**}$, $\mathbf{s}^{**}$) to the following linear program, provided that:

a $\mathbf{s}^{**} = \mathbf{0}$ (i.e., $z^{**}_{\text{SKED}} = 0$, below)

b $\sum_{k\in\mathcal{K}} \alpha^{**}_k = 1$, i.e., $\boldsymbol{\alpha}^{**}$ is a probability distribution for the surveillance sessions represented by $\mathbf{g}_k$, $k \in \mathcal{K}$:

$$\mathbf{SKED}(\mathcal{K})\ z^{**}_{\mathbf{SKED}} = \min_{\boldsymbol{\alpha},\mathbf{s}} \sum_{c\in C^*} \sum_{i\in I} s_{ci} \tag{16}$$

subject to:

$$\sum_{k\in\mathcal{K}} g_{kci}\alpha_k + s_{ci} = f^*_{ci}\ \forall c \in C^*, i \in I \tag{17}$$

$$\alpha_k \geq 0\ \forall k \in \mathcal{K};\ s_{ci} \geq 0\ \forall c \in C^*, i \in I_c \tag{18}$$

If condition (a) is true, summing all constraints (17) gives $\sum_{k\in\mathcal{K}} |C|\,\alpha^{**}_k = |C|$, which means that condition (b) will hold automatically. Assuming $z^{**}_{\text{SKED}} = 0$, the installed cameras are scheduled very simply then: every pre-specified interval of time (e.g., every 10 minutes), create a new surveillance session by, according to the probability distribution $\boldsymbol{\alpha}^{**}$, randomly selecting and applying some assignment $\mathbf{g}_k$.

If a valid distribution exists, we can find it by solving $\mathbf{SKED}(\mathcal{K})$ using a standard, linear-programming-based column-generation algorithm (Gilmore and Gomory, 1961), provided that the algorithm does not cycle. The algorithm follows:

GenDist

Input: $I, C, \mathbf{f}^*$ and C^* from **MX1**.

Output: Probability distribution α_k on assignment vectors $\mathbf{g}_k$, $k \in \mathcal{K}'$.

{

$\mathcal{K}' \leftarrow \varnothing$;

Repeat{

Solve $\mathbf{SKED}(\mathcal{K})$ for

(i) $z^{**}_{\mathbf{SKED}}$,

(ii) α^{**}_k for $k \in \mathcal{K}'$, and

(iii) μ^{**}_{ci} for $c \in C^*, i \in I$, which are optimal dual variables for constraints (17);

If $(z^{**}_{\mathbf{SKED}} = 0)$ go to Solved;

Solve this assignment problem for $\hat{\mathbf{g}}$:

$$\min_{\mathbf{g}} \sum_{c\in C^*} \sum_{i\in I} \mu^{**}_{ci} g_{ci} \tag{19}$$

subject to:

$$\sum_{i\in I | f^*_{ci}>0} g_{ci} = 1\ \forall c \in C^* \tag{20}$$

$$\sum_{c\in C^* | f^*_{ci}>0} g_{ci} \leq 1\ \forall i \in I \tag{21}$$

$$g_{ci} \in \{0,1\} \forall c \in C^*, i \in I; \tag{22}$$

$k \leftarrow k + 1$;

$\mathbf{g}_k \leftarrow \hat{\mathbf{g}}$;

$\mathcal{K}' \leftarrow \mathcal{K}' \cup \{k\}$;

}

Solved:

For $(k \in \mathcal{K}')$ {

Print ('Assign', $\mathbf{g}_k$, 'with probability', α^{**}_k);

}

}

Of course, the assignment problem (19) to (22) solves efficiently using combinatorial or linear-programming-based techniques [e.g., Ahuja et al., (1993), pp.470–473].

To show that $\mathbf{SKED}(\mathcal{K})$ has an optimal solution with $\mathbf{s}^{**} = \mathbf{0}$, we use the following standard definitions:

A *permutation matrix* G^P_k is an $n \times n$ 0-1 matrix with exactly one 1 in each row and column.

An *assignment matrix* G^A_k is an $m \times n$ 0-1 matrix, $m \leq n$, with exactly one 1 in each row and at most one 1 in each column.

A (*singly*) *stochastic matrix* F^{SS} is an $m \times n$ non-negative real matrix, $m \leq n$, with each row sum being 1 an each column sum being at most 1.

A *doubly stochastic matrix* F^{DS} is an $n \times n$ non-negative real matrix each of whose row and column sums is 1.

Lemma 1: An $n \times n$ doubly stochastic matrix F^{DS} can be written as a convex combination of the $n \times n$ permutation matrices $G^P_k, k \in \mathcal{K}^P$, where $\mathcal{K}^P$ is the index set for all such permutation matrices.

Proof: For example, see Theorem 5.4 in Berman and Plemmons (1994). ■

Lemma 2: Any stochastic matrix F^{SS} can be expanded to a doubly stochastic matrix F^{DS} by defining:

a $f^{DS}_{ij} = f^{SS}_{ij}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$

b $f_{i'j}^{DS} = \left(1 - \sum_{i=1}^{m} f_{ij}^{SS}\right) \Big/ (n-m)$ for $i' = m+1, ..., n$ and $j = 1, \ldots, n$.

Proof: The first m rows of F^{DS} are identical to the first m rows of F^{SS}, so each such row sums to 1. It remains to show that all columns of F^{DS} sum to 1 as do all rows $i = m + 1, \ldots, n$. We leave the details to the reader. ■

Lemma 3: Any $m \times n$ stochastic matrix F^{SS} with $m \leq n$ can be written as a convex combination of the assignment matrices $G_k^A, k \in \mathcal{K}^P$, that are created by deleting the last $n - m$ rows from the corresponding permutation matrices $G_k^P, k \in \mathcal{K}^P$.

Proof: If $m = n$, F^{SS} must actually be a doubly stochastic matrix and the result follows trivially from Lemma 1 because $G_k^A = G_k^P, k \in \mathcal{K}^A = \mathcal{K}^P$. So, now assume that $m < n$ for F^{SS}. Using Lemma 2, add $n - m$ rows i, $i = m + 1, \ldots, n$ with appropriate entries to extend this matrix to a doubly stochastic matrix F^{DS}. From Lemma 1, we know that a convex combination of the $n \times n$ permutation matrices $G_k^P, k \in \mathcal{K}^P$, must exist. Find such a convex combination, that is, find $\boldsymbol{\alpha}^{**}$ satisfying

$$\sum_{k \in \mathcal{K}^P} G_k^A \alpha_k = F^{DS} \tag{23}$$

$$\sum_{k \in \mathcal{K}^P} \alpha_k = 1 \tag{24}$$

$$\alpha_k \geq 0 \forall k \in \mathcal{K}^P. \tag{25}$$

Deleting the last $n - m$ constraints in (23) shows that the convex combination required by the theorem exists. ■

Lemma 4: Any $m \times m$ stochastic matrix F^{SS} with $m \leq n$ can be written as a convex combination of the set of all $m \times n$ assignment matrices $G_k^A, k \in \mathcal{K}^A$.

Proof: As in the proof of Lemma 3, the case of $m = n$ is trivially true. Thus, assume $m < n$ and let (23)′ denote the subset of constraints (23) that remain after the last (row-deletion) step in that proof. We know that vector $\boldsymbol{\alpha}^{**}$ applied to the system (23)′, (24), (25) defines a convex combination of assignment matrices, although some of these matrices will be duplicates. Collecting like terms in this system yields the desired result. ■

Theorem 1: **SKED**($\mathcal{K}$) has an optimal solution ($\boldsymbol{\alpha}^{**}$, $\mathbf{s}^{**}$) with $\mathbf{s}^{**} = \mathbf{0}$

Proof: This theorem's statement is equivalent to that of Lemma 4, with the stochastic matrix F^{SS} being represented as the vector $\mathbf{f}^*$ and the assignment matrices G_k^A, $k \in K^A$, represented by corresponding assignment vectors $k \in \mathcal{K} \equiv \mathcal{K}^A$. ■

Theorem 2: In the absence of cycling, algorithm **GenDist** produces a valid probability distribution for camera-assignment scheduling.

Proof: This follows from:

a Theorem 1 which proves that a valid distribution exists

b the fact that **GenDist** just implements a special version of the simplex algorithm for solving a linear program

c the simplex algorithm always converges unless it cycles. ■

In practice, we find that **GenDist** converges in a modest number of steps, and thus no anti-cycling techniques are required. [For examples of anti-cycling techniques that might be applied if needed, see Gill et al. (1989) and the references therein.]

4 Computational results

Using simulated scenarios, this section investigates the empirical behaviour of our procedure for locating camera towers and scheduling surveillance sessions for the cameras on those towers. The computation also validates our methods in a self-evident fashion by:

a providing empirical evidence of the correctness of Section 3's theory

b substantiating our claim that the procedure consisting of model **MX1** and the algorithm **GenDist** solves realistic problems efficiently enough for practical applications using standard hardware and software

c identifying extreme parameter ranges for which **MX1** produces poor results or must fail

d exploring an alternative camera-location model for such parameter ranges.

4.1 Computational environment

We generate all mathematical programs using the standard modelling system GAMS (version 23.3.3), and solve them with CPLEX (version 12.6.0.1). A Lenovo W541 laptop computer carries out all computations, running at 2.9 GHz and using 16 GB of RAM. All MIPs are solved using a relative optimality tolerance of 1%, subject to a time limit of 1,000 seconds.

4.2 Scenario description

We model a square study region with each side having a length of 100 in arbitrary units. Two 'geographical scenarios' apply, a medium-sized one '*M*' and a large-sized one '*L*'. The former has 15 candidate locations and 60 POIs and the latter has 30 and 120, respectively. The x- and y-coordinates for all of these are randomly chosen according to a uniform distribution over the study region. Letting $D_{\ell i}$ denote the distance between ℓ and i, and letting $\bar{D} = 30$, both geographical scenarios use

$$p_{\ell i} \equiv \begin{cases} 1 & \text{if } D_{\ell i} \leq \bar{D} \\ \bar{D}^2 / D_{\ell i}^2 & \text{if } D_{\ell i} > \bar{D}. \end{cases}$$

The reduction in $p_{\ell i}$ with $D_{\ell i}^{-2}$ reflects reduced resolution of an image on a rectangular screen with a given number of pixels as camera-to-POI distance increases (Bodor et al., 2007).

We further create 32 complete 'test scenarios' from the geographical scenarios *M* and *L*, by using variants of:

a damage values (either selected according to a discrete uniform distribution over $d_i \in \{1, \ldots, 5\} \forall i$ or set as $d_i = 1 \forall i$)

b number of potential tower locations $n_T \in \{5, 10, 15, 20\}$, depending on the geographical scenario

c number of cameras per tower $n_{C/T} \in \{1, \ldots, 12\}$, depending on other parameters.

Table 1 describes the set of test scenarios. For example, *M*5/10/3 indicates a medium-sized geographical scenario that is completed by specifying $d_i \in \{1, \ldots, 5\} \forall i$, $n_T = 10$ and $n_{C/T} = 3$.

Table 1 Test scenarios analysed

Scenario	$\lvert \mathcal{L} \rvert$	$\lvert I \rvert$	d_i	n_T: $n_{C/T}$
$Md/n_T/n_{C/T}$	15	60	{1}	5: 1, …, 12
			{1, …, 5}	10: 1, …, 6
$Ld/n_T/n_{C/T}$	30	120	{1}	15: 1, …, 8
			{1, …, 5}	20: 1, …, 6

Note: See the text for a detailed explanation.

4.3 Test scenario results

Tables 2 and 3 present the results for the medium- and large-sized test scenarios, respectively. In addition to the optimal objective value $z^*_{\mathbf{MX1}}$, the tables list: $\lvert \mathcal{K}' \rvert$, the total number of assignment vectors that are generated; $\lvert \mathcal{K}'' \rvert$, the number of those vectors with positive weight in the optimal scheduling distribution α^*; Δ_{avg}, the average number of individual camera-to-POI assignments that change in transitioning from one session to another; $\Delta_{\max}$ the maximum number of such changes that can occur; and 'Time' which gives the total elapsed seconds used to solve **MX1** and execute **GenDist**. We note that each row in these tables represents only a single, randomly generated instance. Our computational experience indicates that, while optimal objective values may vary greatly among randomly generated instances, the conclusions reached on computational behaviour do not change across the range of parameters displayed.

We conclude from Tables 2 and 3 that **MX1** and **GenDist** typically produce good schedules quickly, with probabilities of detection that increase sensibly as $n_{C/T}$ increases. We also note that $\lvert \mathcal{K}' \rvert$, the number of assignments generated, tracks well with total solution time. Thus, this value seems to give a good implementation-independent measure of the difficulty of solving a particular model instance.

Table 2 Results for medium-sized scenarios

Scenario $\bar{D} = 30$	$z^*_{\mathbf{MX1}}$	$\lvert \mathcal{K}' \rvert$	$\lvert \mathcal{K}'' \rvert$	Δ_{avg}	$\Delta_{\max}$	*Time (sec.)*
M1/5/1	0.9240	106	60	4.6	5	18.3
M1/5/2	0.8448	103	58	8.4	10	18.7
M1/5/3	0.7642	156	58	11.4	15	28.6
M1/5/4	0.6812	88	60	13.8	19	19.4
M1/5/5	0.6028	96	60	16.0	20	25.2
M1/5/6	0.5193	105	60	16.7	23	42.7
M1/5/7*	0.4449	95	60	18.5	27	1018.9
M1/5/8	0.3900	114	60	20.0	30	39.1
M1/5/9	0.3866	115	58	19.7	25	26.4
M1/5/10	0.3866	86	32	14.1	20	18.6
M1/5/11	0.3866	26	14	5.4	8	7.4
M1/5/12	0.3866	4	4	10.1	20	6.9
M5/5/1	3.7218	65	30	3.9	5	10.8
M5/5/2	3.0110	45	30	6.7	10	8.2
M5/5/3	2.5040	80	38	9.3	13	16.7
M5/5/4	2.0062	56	38	10.2	17	14.8
M5/5/5	1.9329	128	36	9.6	17	23.7
M5/5/6	1.9329	66	26	7.8	13	12.7
M5/5/7	1.9329	82	29	9.3	14	15.8
M5/5/8	1.9329	50	18	5.4	9	9.8
M5/5/9	1.9329	55	16	7.1	13	11.1
M5/5/10	1.9329	52	14	5.9	10	11.1
M5/5/11	1.9329	58	10	9.1	16	16.4
M5/5/12	1.9445	3	3	6.7	14	4.7
M1/10/1	0.8386	95	60	8.4	10	8.4
M1/10/2	0.6740	83	60	13.7	18	13.7
M1/10/3	0.5131	82	60	16.6	24	16.6
M1/10/4	0.3866	90	52	17.9	25	17.9
M1/10/5	0.3866	105	31	16.3	26	16.3
M1/10/6	0.3866	11	11	16.7	25	16.7
M5/10/1	2.9892	112	38	6.9	10	24.0
M5/10/2	1.9620	124	50	10.8	16	29.6
M5/10/3	1.9329	81	32	9.5	13	99.5
M5/10/4	1.9329	128	25	8.9	15	29.8
M5/10/5	1.9329	128	25	8.9	15	29.5
M5/10/6	1.9329	68	18	10.1	20	16.3

Notes: See the text for an explanation of the table's entries. For the case marked '*', **MX1** times out with a 2.2% optimality gap remaining. **GenDist** operates normally on the resulting solution, however.

Tables 2 and 3 also show, however, that the methodology is imperfect: when the number of towers is small or the best set of towers has poor geographical coverage, improvement in expected damage can stall as $n_{C/T}$ increases; for example, see $M1/5/n_{C/T}$ for $n_{C/T} \geq 10$. Because expected damage does not stall as $n_{C/T}$ increases for the large-sized scenarios reported in Table 3, we conclude that a high-quality solution requires a good geographical dispersion of potential tower locations.

GenDist never generates more than 600 camera-to-POI assignments, and the number used in a schedule (i.e., having positive probability) never exceeds 120. The only result that may seem unfortunate is that the average number of changes in POI surveillance in moving from one assignment to another, Δ_{avg}, is a large fraction of the maximum number that might occur, $\Delta_{\max}$. For example, the schedule for test scenario *M*5/5/4 surveils 20 POIs simultaneously but, on average over half of the POIs being surveilled (i.e., 10.2) change from one session to the next.

Table 3 Computational results for large-sized scenarios

Scenario $\bar{D} = 30$	$z^*_{\mathbf{MX1}}$	$\lvert \mathcal{K}' \rvert$	$\lvert \mathcal{K}'' \rvert$	Δ_{avg}	$\Delta_{\max}$	*Time (sec.)*
L1/15/1	0.8752	285	119	13.1	15	54.8
L1/15/2	0.7501	599	120	22.5	26	103.1
L1/15/3	0.6250	131	102	30.2	35	32.4
L1/15/4	0.5000	22	13	30.0	44	7.1
L1/15/5	0.3750	179	106	37.8	44	49.2
L1/15/6	0.2500	83	66	37.6	46	26.2
L1/15/7	0.1251	320	106	35.3	43	100.6
L1/15/8	0.0000	1	1	0	0	2.5
L5/15/1	3.0457	63	45	10.4	14	12.3
L5/15/2	2.2150	132	71	18.0	25	26.9
L5/15/3	1.6076	138	89	24.	32	34.4
L5/15/4	1.0708	169	89	28.9	37	41.7
L5/15/5	0.7758	226	117	32.7	47	61.9
L5/15/6	0.5193	310	119	37.9	48	99.1
L5/15/7	0.2599	336	117	34.8	52	121.7
L5/15/8	0.0000	1	1	0	0	7.2
L1/20/1	0.8753	243	120	13.1	15	47.6
L1/20/2	0.6667	144	80	26.7	32	34.0
L1/20/3	0.5000	18	8	30.0	46	4.5
L1/20/4	0.3333	63	42	36.4	46	15.8
L1/20/5	0.1667	318	96	36.6	49	89.4
L1/20/6	0	1	1	0	0	3.9
L5/20/1	2.7543	177	71	13.6	18	32.8
L5/20/2	1.7847	171	89	22.7	32	34.9
L5/20/3	1.0708	160	90	28.2	33	35.9
L5/20/4	0.6895	237	105	31.1	38	60.0
L5/20/5	0.3447	456	120	35.4	48	119.8
L5/20/6	0	1	1	0	0	4.61

Note: See the text for an explanation of the entries.

4.4 *Many cameras: an alternative model*

As the total number of cameras available $n_{C/T} \cdot n_T$ starts to approach the number of POIs $\lvert \mathcal{L} \rvert$, focusing two or more cameras on distinct towers upon a single POI might improve detection probabilities and reduce expected damage. This could alleviate some of the stalling of the expected-damage values seen going down the rows of Table 2. Assuming independent detections, and requiring that each camera be permanently assigned to some POI, the following model applies **MX1**'s game-theoretic paradigm (Salmeron and Wood, 2015):

$$\mathbf{MX2}: z^*_{\mathbf{MX2}} = \min_{\mathbf{y},\mathbf{y}',z} z \tag{26}$$

subject to:

$$z \ge d_i \prod_{\ell \in L} q_{\ell i}^{y'_{\ell i}} \, \forall \, i \in \mathcal{I} \tag{27}$$

$$\sum_{i \in \mathcal{I}} y'_{\ell i} = n_{C/T} \, y_\ell \;\; \forall \; \ell \in \mathcal{L} \tag{28}$$

$$y'_{\ell i} \le y_\ell \;\; \forall \; \ell \in \mathcal{L}, i \in \mathcal{I} \tag{29}$$

$$y_\ell \in \{0,1\} \; \forall \ell \in \mathcal{L} \tag{30}$$

$$y'_{\ell i} \in \{0,1\} \; \forall \ell \in \mathcal{L}, i \in \mathcal{I} \tag{31}$$

MX2 linearises easily, of course, and we solve that linearised version in practice. Constraints (29) are not required, but they may tighten the model's continuous relaxation and reduce solution times. Note also that the fixed camera-to-POI assignments assumed in **MX2** alleviate the need for a secondary scheduling algorithm and any related theory.

For testing, we fix $n_T = 5$, and vary $n_{C/T}$ from 1 to 8 in 'small-sized scenarios' defined to have $\lvert \mathcal{L} \rvert = 9$ and $\lvert \mathcal{I} \rvert = 30$. Following the nomenclature above, $Sd/n_T/n_{C/T} = Sd/5/n_{C/T}$ denotes instances of these scenarios. We use small-sized scenarios because certain larger instances of **MX2** can be difficult to solve. Also, to avoid scenarios in which expected damage drops abruptly to nearly 0, we reduce $\bar{D}$ to 20.

Table 4 compares the behaviour of **MX1** and **MX2**. Consider first the eight instances with $d_i = 1$ for all POIs i. The 'fixed-assignment system' represented by **MX2** must leave some POIs unsurveilled when $1 \le n_{C/T} \le 5$, and hence, the initial collection of 1s in column 4. By contrast, the 'variable-assignment system' represented by **MX1** can spend some time surveilling every POI in those scenarios and reduce expected damage below 1. But then, because of the requirement for the **MX1**-based solution that no POI be surveilled simultaneously by two or more cameras, the value of adding more cameras to available towers trails off and may stop improving as the total number of cameras becomes large; of course, the model becomes infeasible once that number exceeds $\lvert \mathcal{I} \rvert$.

When $n_{C/T} = 6$, the total number of cameras equals $\lvert \mathcal{I} \rvert$, and thus exactly one camera can be assigned to each POI in either model to reduce expected damage below 1. The optimal objective values for the two models are therefore identical. For $n_{C/T} > 6$, **MX2** can assign two or more cameras from separate towers to some POIs in order to continue reducing expected damage. In these cases, however, the variable-assignment system based on **MX1** simply becomes infeasible, given the requirement that no two cameras surveil the same POI simultaneously.

Table 4 Comparison of solutions to **MX1** and **MX2** on small-sized scenarios as the total number of cameras approaches and exceeds the number of POIs

Scenario $\bar{D} = 20$	*MX1*		*MX2*		
	z^*_{MX1}	*Time (sec.)*	z^*_{MX2}	*POIs w/ extra cover*	*Time (sec.)*
S1/5/1	0.8848	0.30	1	-	-
S1/5/2	0.7727	0.48	1	-	-
S1/5/3	0.7385	0.33	1	-	-
S1/5/4	0.7385	0.28	1	-	-
S1/5/5	0.7385	0.33	1	-	-
S1/5/6	0.7385	0.58	0.7385	0	1.95
S1/5/7	NA	NA	0.5668	3	1.41
S1/5/8	NA	NA	0.5111	4	1.02
S5/5/1	2.8766	0.20	5	0	12.76
S5/5/2	2.3572	0.41	3	0	0.25
S5/5/3	2.0706	0.67	3	0	203.29
S5/5/4	2.0706	0.75	2.4735	0	1.14
S5/5/5	2.0706	0.75	1.7073	2	1.20
S5/5/6	2.0706	0.75	1.4930	4	1.56
S5/5/7	NA	NA	1.4275	7	1.53
S5/5/8	NA	NA	1.4275	7	1.42

Notes: 'NA' indicates that **MX1** is infeasible, while '-' indicates that any solution to **MX2** would give the same objecitve value. 'POIs w/ extra cover' gives the number of POIs with at least two cameras assigned to them in **MX2**'s solution.

For the scenarios with $d_i \in \{1, \ldots, 5\}$ the fixed-assignment system (**MX2**) must leave some POIs unsurveilled, and the worst damage from an unsurveilled POI defines the expected damage of 5, 3 or 3, when $n_{C/T} = 1$, 2 or 3, respectively. However, when $n_{C/T} = 4$, some POIs remain unsurveilled, but the expected damage of 2.4735 is defined by a POI that is, in fact, surveilled.

When $n_{C/T} = 5$, we see that the fixed-assignment system outperforms the variable-assignment system, even though some POIs must be left unsurveilled. This happens because the fixed-assignment system can assign at least one camera to every POI i with $d_i > 2$ and, where that does not suffice to bring the worst-case expected damage down sufficiently, that reduction can be achieved by assigning two or more cameras to certain 'high-value POIs' (i.e., POIs i with $d_i \geq 3$). Thus, when nominal damage values vary, the fixed-assignment system can outperform the variable-assignment system even when some POIs must be left unsurveilled.

We remind the reader of the strong assumption of independent detections made in developing the results above. The apparent superiority of the fixed-assignment system modelled by **MX2**, for certain parameter settings, might not be realisable in practice.

5 Conclusions

We have demonstrated a method for optimally locating a set of surveillance-camera towers and then scheduling surveillance of 'POIs' by installed cameras. The model seeks to minimise worst-case expected damage from attacks or other events in an adversarial setting. A key assumption applies: at most one camera may surveil a POI at any time.

By using a two-step process, we create a solution technique that solves quickly enough on a laptop computer for practical use. First, the solution to a facility-location-like model identifies camera locations that optimise aggregate, normalised, camera-to-POI assignment times. Then, a linear-programming-based column-generation algorithm produces a probability distribution for camera-to-POI assignments – each with at most one camera per POI – whose marginal probabilities match the normalised, aggregate assignment times. We prove that this solution must be optimal and show that it is achievable in practice.

When the total number of cameras becomes large compared to the number of POIs, we also demonstrate the potential benefits a tower-location model that uses a fixed assignment of cameras to POIs. Specifically, each camera surveils a single POI, but two or more cameras may surveil a POI, provided that those cameras are mounted on separate towers. The fixed-assignment model may reduce expected damage as the total number of cameras grows, but that result depends strongly on:

a the variation across POIs of the nominal damage associated with a successful attack

b the assumption of independent detections among cameras surveilling the same POI from different locations.

These issues will require further research.

Our general topic area has numerous opportunities for additional research involving simulation. For instance, surveillance cameras can detect riots, but simulation of human behaviour in this context might lead to methods that can detect and help stop a riot at the earliest possible stage (Bruzzone et al., 2011; Lacko et al., 2013). Also, if we relax some of the assumptions required of our optimisations models – suppose simultaneous attacks at multiple locations can occur – then the models' results could be viewed as approximate solutions that need to be verified or modified through simulation studies; for example, see Schriber and Stecke (1987) and Wiese et al. (2009).

Acknowledgements

We thank the Defense Threat Reduction Agency for supporting this work. We also thank two anonymous referees for their helpful comments to improve the presentation of this paper.

References

Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993) *Network Flows: Theory Algorithms and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey.

Berman, A. and Plemmons, R.J. (1994) *Nonnegative Matrices in the Mathematical Sciences*, Society for Industrial and Applied Mathematics, Philadelphia.

Bodor, R., Drenner, A., Schrater, P. and Papanikolopoulos, N. (2007) 'Optimal camera placement for automated surveillance tasks', *Journal of Intelligent and Robotic Systems*, Vol. 50, No. 3, pp.257–295.

Braga, A.A. and Bond, B.J. (2008) 'Policing crime and disorder hot spots: a randomized controlled trial', *Criminology*, Vol. 46, No. 3, pp.577–607.

Brown, G., Carlyle, M., Salmerón, J. and Wood, K. (2006) 'Defending critical infrastructure', *Interfaces*, Vol. 36, No. 6, pp.530–544.

Bruzzone, A.G., Tremori, A., Tarone, F. and Madeo, F. (2011) 'Intelligent agents driving computer generated forces for simulating human behaviour in urban riots', *International Journal of Simulation and Process Modelling*, Vol. 6, No. 4, pp.308–316.

Chen, C-H. and Yucesan, E. (2005) 'An alternative simulation budget allocation scheme for efficient simulation', *International Journal Simulation and Process Modelling*, Vol. 1, Nos. 1/2, pp.49–57.

Costello, C.J. and Wang, I.J. (2005) 'Surveillance camera coordination through distributed scheduling', in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp.1485–1490.

Defense Industry Daily (2015) 'The USA's RAID program: small systems, big surveillance time', 15 May [online] http://www.defenseindustrydaily.com/the-usas-raid-program-small-aerostats-big-surveillance-time-02779/ (accessed 8 November 2016).

Fu, M.C., Glover, F.W. and April, J. (2005) 'Simulation optimization: a review, new developments, and applications', in *Proceedings of the 37th Winter Simulation Conference*, pp.83–95.

Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1989) 'A practical anti-cycling procedure for linearly constrained optimization', *Mathematical Programming*, Vol. 45, No. 1, pp.437–474.

Gillespie, D.T. (2001) 'Approximate accelerated stochastic simulation of chemically reacting systems', *The Journal of Chemical Physics*, Vol. 115, No. 4, pp.1716–1733.

Gilmore, P.C. and Gomory, R.E. (1961) 'A linear programming approach to the cutting-stock problem', *Operations Research*, Vol. 9, No. 6, pp.849–859.

Hörster, E. and Lienhart, R. (2006) *On the Optimal Placement of Multiple Visual Sensors*, Universität Augsburg, Institut für Informatik, Report 2006-18.

Jafferali, M., Venkateshwaran, J. and Son, Y.J. (2005) 'Performance comparison of search-based simulation optimisation algorithms for operations scheduling', *International Journal of Simulation and Process Modelling*, Vol. 1, Nos. 1–2, pp.58–71.

Jänen, U., Spiegelberg, H., Sommer, L., von Mammen, S., Brehm, J. and Hähner, J. (2013) 'Object tracking as job-scheduling problem', in *Proceedings Seventh International Conference on Distributed Smart Cameras*, pp.1–7.

Krahnstoever, N., Yu, T., Lim, S., Patwardhan, K. and Tu, P. (2008) 'Collaborative real-time control of active cameras in large scale surveillance systems', in *Workshop on Multi-Camera and Multi-modal Sensor Fusion Algorithms and Applications – M2SFA2 2008*, Marseille, France, October <inria-00326743>.

Lacko, P., Ort, M., Kyžňanský, M., Kollár, A., Pakan, F., Ošvát, M. and Branišová, J. (2013) 'Riot simulation in urban areas', in *2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp.489–492.

Liu, H., Wan, P., Yi, C.W., Jia, X., Makki, S.A.M. and Pissinou, N. (2005) 'Maximal lifetime scheduling in sensor surveillance networks', in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 4, pp.2482–2491.

Murray, A.T., Kim, K., Davis, J.W., Machiraju, R. and Parent, R. (2007) 'Coverage optimization to support security monitoring', *Computers, Environment and Urban Systems*, Vol. 31, No. 2, pp.133–147.

Natarajan, P., Hoang, T.N., Low, K.H. and Kankanhalli, M. (2012) 'Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance', in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, Vol. 1, pp.155–162.

Neil, D., Thomas, N. and Baker, B. (2007) 'Threat image projection in CCTV', in *Proceedings SPIE 6741. Optics and Photonics for Counterterrorism and Crime Fighting III*, p.674102.

Nelson, B.L., Swann, J., Goldsman, D. and Wheyming, S. (2001) 'Simple procedures for selecting the best simulated system when the number of alternatives is large', *Operations Research*, Vol. 49, No. 6, pp.950–963.

Owen, S.H. and Daskin, M.S. (1998) 'Strategic facility location: a review', *European Journal of Operational Research*, Vol. 111, No. 3, pp.423–447.

Qureshi, F.Z. and Terzopoulos, D. (2006) 'Surveillance camera scheduling: a virtual vision approach', *Multimedia Systems*, Vol. 12, No. 3, pp.269–283.

Salmeron, J. and Wood, R.K. (2015) 'Optimally locating surveillance assets in urban areas', in *Proceedings of the International Defense and Homeland Security Simulation Workshop 2015*.

Schriber, T.J. and Stecke, K.E. (1987) 'Using mathematical programming and simulation to study FMS machine utilizations', in *Proceedings of the 19th Winter Simulation Conference*, pp.725–730.

Snyder, L.V. (2006) 'Facility location under uncertainty: a review', *IIE Transactions*, Vol. 38, No. 7, pp.547–564.

Wang, B. (2011) 'Coverage problems in sensor networks: a survey', *ACM Computing Surveys*, Vol. 43, No. 4, Paper 32, pp.32:1–32:53.

Wiese, J., Suhl, L. and Kliewer, N. (2009) 'Mathematical programming and simulation based layout planning of container terminals', *International Journal of Simulation and Process Modelling*, Vol. 5, No. 4, pp.313–323.