Game-Theoretic Resource Allocation for Protecting Large Public Events

Yue Yin
The Key Lab of Intelligent Information Processing, ICT, CAS
University of Chinese Academy of Sciences
Beijing 100190, China
melody1235813@gmail.com

Bo An
School of Computer Engineering
Nanyang Technological University
Singapore 639798
boan@ntu.edu.sg

Manish Jain
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
jmanish@cs.vt.edu

Abstract

High profile large scale public events are attractive targets for terrorist attacks. The recent Boston Marathon bombings on April 15, 2013 have further emphasized the importance of protecting public events. The security challenge is exacerbated by the dynamic nature of such events: e.g., the impact of an attack at different locations changes over time as the Boston marathon participants and spectators move along the race track. In addition, the defender can relocate security resources among potential attack targets at any time and the attacker may act at any time during the event.

This paper focuses on developing efficient patrolling algorithms for such dynamic domains with continuous strategy spaces for both the defender and the attacker. We propose SCOUT-A, which makes assumptions on relocation cost, exploits payoff representation and computes optimal solutions efficiently. We also propose SCOUT-C to compute the exact optimal defender strategy for general cases despite the continuous strategy spaces. SCOUT-C computes the optimal defender strategy by constructing an equivalent game with discrete defender strategy space, then solving the constructed game. Experimental results show that both SCOUT-A and SCOUT-C significantly outperform other existing strategies.

Introduction

Public events in major cities are prime terrorism targets since they usually provide easy access to large number of targets for the adversary. There have been some successful terrorist attacks on large public events in the US and Europe in the past few years, e.g., the recent Boston Marathon bombings on April 15, 2013; the 7/7 2005 London bombings. Intelligent deployment of limited security resources to protect such events is therefore extremely important and challenging since the importance of targets changes over time. For example, the value of targets along a marathon track changes over time with the changing number of participants and spectators at any specific area over the course of the race.

In addition, since the attacker may attack at any time and the defender can relocate resources among targets at any time, the strategy space of each agent is continuous and infinite. Furthermore, due to the relative infrequency of such events, the attacker may not be able to conduct surveillance and respond to a distribution of defender strategies (Korzhyk, Conitzer, and Parr 2011; Letchford and Vorobeychik 2013). In this case, a pure defender strategy sampled from the optimal mixed strategy does not necessarily outperform the one-shot optimal pure strategy in terms of ex-post payoff. Thus, we propose algorithms to compute the optimal pure defender strategy despite the infinite strategy space.

There has been lots of related research that applies game theoretic approaches to real world security domains (Tambe 2011; An et al. 2013b). However, the proposed approaches cannot be used to solve problems in our domain, since most existing work (An et al. 2013a; Agmon, Urieli, and Stone 2011; Basilico, Gatti, and Amigoni 2009; Kiekintveld, Islam, and Kreinovich 2013; Varakantham, Lau, and Yuan 2013) assumes that the payoffs of targets are static over time. While some researchers addressed time-critical domains (Fang, Jiang, and Tambe 2013; Yin et al. 2012), there are two key differences in our work. First, previous work arbitrarily discretizes defender strategy space and as such, their solution is not optimal when continuous defender strategy space is considered. Second, they compute the mixed strategies for the defender assuming that the attacker will observe and react to the defender, which is substantially different from the solution concept in our domain, where we compute the optimal pure strategy for the defender.

This paper makes four key contributions. First, we design a game model to minimize the worst-case loss of the defender. In our model, each agent has a continuous and infinite strategy space and the payoff of an attack is time-dependent. Second, we propose SCOUT-A to compute the optimal dynamic allocation of resources when the defender is able to relocate security resources among targets without any delay in time. To avoid traversing the whole strategy space, SCOUT-A exploits a novel approach to represent payoffs, in which utilities of targets are represented by continuous functions. Third, we propose SCOUT-C to deal with general cases in our model. SCOUT-C computes the optimal defender strategy by constructing an equivalent game with discrete defender strategy space, then solving the constructed game. Finally, we present experimental results showing that both SCOUT-A and SCOUT-C outperform some simple defender strategies observed in previous public events.

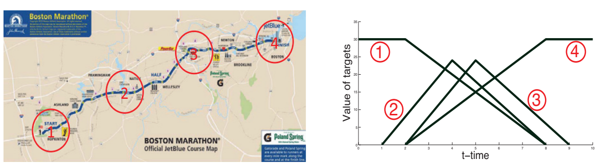
Motivating Domain

While our model is quite general for modeling dynamic patrolling in domains with time-critical payoff, we use the tragedy of bombings near the finish line in the 2013 Boston Marathon as a motivating example (Figure 1).

(a) Moment of bombing (b) Boston Marathon course

Figure 1: Boston Marathon bombings on April 15, 2013

For the purpose of illustration, we divide the course into four segments to represent four potential targets as is marked in Figure 2(a). Example values for the targets are shown in Figure 2(b). As the event proceeds, the number of participants and spectators near the starting line decreases, thus the damage caused by attacking target 1 decreases, leading to the decreasing in the value of target 1, depicted by Line 1 in Figure 2(b). Similar variations in the value of targets happen for targets 2, 3 and 4 as well (shown via lines 2-4).



(a) Marked targets (b) Time-critical values

Figure 2: Example values for targets

The ending point belongs to target 4 and is of great importance when the race is approaching the end. Intuitively, if the total number of resources is fixed, it may be better if the defender transfers some resources from target 1 to target 4 as the event proceeds. Thus a *dynamic* allocation of resources during the event is in need. In addition, Boston Marathon is held only once a year, which is too rare for the attacker to observe and react to a mixed defender strategy. Therefore, a pure dynamic strategy which minimizes the worst case loss of the defender is a robust solution for the defender.

Problem Statement

Assume that there are n targets (e.g., segments in the marathon scenario) represented by $\mathcal{T} = \{1, \ldots, n\}$ and the defender has m identical security resources (each resource can be a police patrol team in the marathon domain). We assume that a public event starts at time 0 and ends at time $t_e > 0$. The value $v_i(t)$ of each target $i \in \mathcal{T}$ at time $t \in [0, t_e]$ is common knowledge and is a continuous function of t. For ease of analysis, we assume $v_i(t)$ to be piecewise linear (Figure 2(b) shows an example) since such functions are widely used to approximate complicated functions, while the accuracy of approximation can be controlled by setting the number of linear segments. Our algorithms can be easily extended to deal with general form value functions. The defender executes an assignment of resources at time 0 when the event begins. As the event proceeds, the defender may move resources from some targets to other targets. Thus a pure defender strategy includes the initial assignment of resources and the subsequent transfers during $[0, t_e]$. The attacker will choose a certain target to attack during $[0, t_e]$.

Formally, let $Q^0 = \langle q_i^0 \rangle$ represent the initial assignment of resources where q_i^0 is the number of resources assigned to target i when the event begins. We denote all resource transfers during $[0, t_e]$ as $C = \langle C_k \rangle$ where C_k represents the k^{th} transfer. $C_k = \langle c_{ij}^k : i, j \in \mathcal{T} \rangle$ where c_{ij}^k represents the number of resources transferred from target i to target j in the k^{th} transfer. Let τ_k denote the time when the k^{th} transfer begins. Thus a pure defender strategy is fully represented by a tuple $S = (Q^0, C)$. Let $\mathcal{S}$ be the defender strategy space.

Let $Q^t(S) = \langle q_i^t(S) : i \in \mathcal{T} \rangle$ denote the resource assignment at t, with $q_i^t(S)$ representing the number of resources assigned to target i at time t in S. We represent the time required to transfer resources from target i to target j as d_{ij}. Given the set of transferring time of all target pairs, $D = \langle d_{ij} : i, j \in \mathcal{T} \rangle$, and a defender strategy $S = (Q^0, C)$, $q_i^t(S)$ can be computed as follows.

$$q_i^t(S) = q_i^0 + \sum_{C_k \in C, \tau_k \le t - d_{ji}, j \in \mathcal{T}} c_{ji}^k - \sum_{C_k \in C, \tau_k \le t, j \in \mathcal{T}} c_{ij}^k \quad (1)$$

The attacker's pure strategy is represented as (i, t) representing that the attacker attacks target i at time $t, t \in [0, t_e]$. Let $p(r)$ be the probability of a successful attack if the target of the attack is protected by r resources. We set $p(r) = \frac{1}{e^{\lambda r}} (\lambda > 0)$, satisfying $p(r) \in [0, 1]$. λ is a parameter measuring the marginal utility of adding one more security resource. Since $\frac{\partial p(r)}{\partial r} \le 0$ and $\frac{\partial^2 p(r)}{\partial r^2} \ge 0$, $p(r)$ satisfies the law of diminishing marginal returns, i.e., when the number of resources assigned to a target becomes larger, the effect of adding one more resource decreases. We assume that the payoff for a failed attack is 0 for both the players.

Since $v_i(t)$ is a continuous function of t, the payoff of attacking target i when r resources are assigned to i is also a continuous function of t, denoted as $W_i^r(t) = p(r) v_i(t)$. If an attacker chooses to attack target i at time t given the defender strategy S, the expected attacker utility is

$$U^a(i, t, S) = W_i^{q_i^t(S)}(t). \quad (2)$$

We assume a zero-sum game to reduce the complexity of the model. Thus the defender's payoff is opposite to the attacker's payoff, i.e., $U^d(i, t, S) = -U^a(i, t, S)$.

We model the problem as a one-shot game due to the rarity of public events and adopt the maximin strategy as the solution concept. Namely, the defender chooses a strategy maximizing the worst case defender utility, which indicates that the attacker maximizes his utility under the zero-sum game assumption. We focus on computing optimal pure defender strategy. Let the attacker's response function be $f(S) = \{f_{tg}(S) : S \to i, f_{tm}(S) : S \to t\}$ where $f_{tg}(S)$ is the target attacked and $f_{tm}(S)$ is the time of attack.

Definition 1. *A pair of strategies $(S, f(S))$ form a maximin equilibrium if they satisfy the following:*

$$U^a(f_{tg}(S), f_{tm}(S), S) \ge U^a(i, t, S), \forall i \in \mathcal{T}, t \in [0, t_e],$$

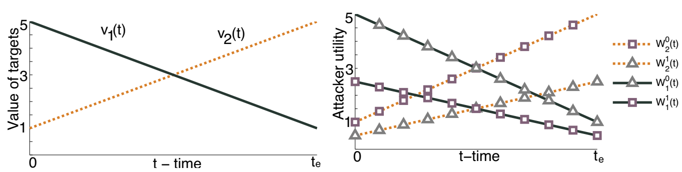
$$U^d(f_{tg}(S), f_{tm}(S), S) \ge U^d(f_{tg}(S'), f_{tm}(S'), S'), \forall S' \in \mathcal{S}.$$

Equilibrium Without Transfer Delay

We first describe an algorithm for computing the maximin equilibrium when the time needed to transfer resources among targets is negligible compared to the event duration (i.e., $d_{ij} = 0$). While we will relax this assumption in the following section, this is a realistic approximation in domains in which the event is held in a relative small area, the event proceeds for a long time, or resources can be transferred quickly (e.g., using helicopters).

When the transfer time is 0, one straightforward approach is computing the optimal assignment at each time point during $[0, t_e]$. We denote a feasible defender assignment by $A = \langle a_i \rangle$ where a_i represents the number of resources assigned to target i. Let $\mathcal{A}$ be the set of possible assignments. For any defender strategy S, the assignment at time t is $Q^t(S) \in \mathcal{A}$. We use the following example to explain concepts used in the algorithm.

Example 1: There are 2 targets whose value functions over time $v_i(t)$ are shown in Figure 3(a). There is 1 resource. $\mathcal{A} = \{A = \langle 1, 0\rangle, A' = \langle 0, 1\rangle\}$. If the defender executes $\langle 1, 0\rangle$ at time t, the attacker's utility of attacking targets 1 and 2 at time t is $W_1^1(t)$ and $W_2^0(t)$ respectively, as is shown by the lines marked by squares in Figure 3(b). Similarly, the lines marked by triangles represent the attacker's utilities when the defender executes $\langle 0, 1\rangle$. Our goal is to determine when to play $\langle 1, 0\rangle$ and when to play $\langle 0, 1\rangle$, so that the maximum attacker utility during $[0, t_e]$ is minimized.



(a) Value functions (b) All possible attacker utility

Figure 3: Example of 2 targets and 1 resource

Formally, an assignment $A = \langle a_i \rangle$ is a minimax assignment at time t if for any $A' = \langle a'_i \rangle \in \mathcal{A}$, $\max_{i \in \mathcal{T}} W_i^{a_i}(t) \leq \max_{i \in \mathcal{T}} W_i^{a'_i}(t)$. Consider Example 1, $\max_{i \in \mathcal{T}} W_i^{a_i}(t)$ ($A = \langle 1, 0\rangle$) is shown by the bold black line in Figure 4 while $\max_{i \in \mathcal{T}} W_i^{a'_i}(t)(A' = \langle 0, 1\rangle)$ is shown by the bold gray line. At time point t_0, $\max_{i \in \mathcal{T}} W_i^{a_i}(t_0) \leq \max_{i \in \mathcal{T}} W_i^{a'_i}(t_0)$, thus A is a minimax assignment at t_0. A pure strategy of the defender that executes the minimax assignments for all time points $t \in [0, t_e]$ will then be the desired minimax pure strategy.

Proposition 1. *If a defender strategy S satisfies that $\forall t \in [0, t_e]$, $Q^t(S)$ is a minimax assignment at t, then S is an optimal defender strategy.*

Proof. Since $Q^t(S)$ is a minimax assignment at any time point t, $\max_{i \in \mathcal{T}} W_i^{q_i^t(S)}(t) \leq \max_{i \in \mathcal{T}} W_i^{q_i^t(S')}(t)$ for $S' \in \mathcal{S}$ and $\forall t \in [0, t_e]$. It follows that $U^a(f_{tg}(S), f_{tm}(S), S) = \max_{i \in \mathcal{T}} W_i^{q_i^t(S)}(t) \leq \max_{i \in \mathcal{T}} W_i^{q_i^t(S')}(t) \leq U^a(f_{tg}(S'), f_{tm}(S'), S')$, where $t = f_{tm}(S)$. Thus S minimizes the maximum attacker utility. □

Proposition 1 suggests that we just need to compute the minimax assignment at each time point to achieve a minimax pure defender strategy. In this paper, we focus on a subclass of minimax assignments — Preferred Assignment (PA), which always exists and is easier to compute.

Definition 2. *Assignment A is a PA at time t if* $\max_{j \in \mathcal{T}} W_j^{a_j}(t) \leq W_i^{a_i - 1}(t) = \frac{v_i(t)}{e^{\lambda(a_i - 1)}}, \forall i \in \mathcal{T}, a_i > 0$.

Intuitively, a PA is better than any assignment which has only one resource different from it.

Proposition 2. *A preferred assignment A at time t is a minimax assignment at time t.*

Proof. Given $A = \langle a_i \rangle$, any assignment $A' = \langle a'_i \rangle$ can be represented as $a'_i = a_i + k_i (\forall i \in \mathcal{T})$, where k_i is an integer. Since the number of resources is fixed, $\sum_{i \in \mathcal{T}} k_i = 0$. If $A' \neq A$, there must be at least one target $j \in \mathcal{T}$ such that $k_j < 0$. Thus we have $W_j^{a'_j}(t) = \frac{v_j(t)}{e^{\lambda(a_j + k_j)}} \geq \frac{v_j(t)}{e^{\lambda(a_j - 1)}} = W_j^{a_j - 1}(t)$. Since A is a preferred assignment at t, we have $W_j^{a_j - 1}(t) \geq \max_{i \in \mathcal{T}} W_i^{a_i}(t)$ based on Definition 2, thus $W_j^{a'_j}(t) \geq \max_{i \in \mathcal{T}} W_i^{a_i}(t)$. Given that $\max_{i \in \mathcal{T}} W_i^{a'_i}(t) \geq W_j^{a'_j}(t)$, we have $\max_{i \in \mathcal{T}} W_i^{a'_i}(t) \geq \max_{i \in \mathcal{T}} W_i^{a_i}(t)$. □

Thus, we can compute an optimal defender strategy by computing a PA for each time point $t \in [0, t_e]$. However, it is computationally infeasible to list all continuous time points to check which assignment is a PA at t. Fortunately, we can show that the PA will not change continuously, thus we do not need to compute PAs for all time points. Assume that A is a PA at t_0. For any pair of targets i, j which satisfies that $\exists t \in [t_0, t_e]$ such that $\frac{\partial W_i^{a_i - 1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$, let $I_{ij}(A, t_0)$ represent the time of the first intersection of line $W_i^{a_i - 1}(t)$ and line $W_j^{a_j}(t)$ when $t \in [t_0, t_e)$. Let $\mathbf{I}(A, t_0) = \{I_{ij}(A, t_0) : i, j \in \mathcal{T}\}$. We have the following proposition about the period during which a PA is valid.

Proposition 3. *If A is a PA at time t_0, A is a PA at any $t \in [t_0, \min(\mathbf{I}(A, t_0))]$.*

Proof. We prove this result by contradiction. Let $t_1 = \min \mathbf{I}(A, t_0)$. If there exists a time point $t_2 \in (t_0, t_1)$ at which A is not a preferred assignment, then there exist i, j such that $W_i^{a_i - 1}(t_2) < W_j^{a_j}(t_2)$ based on Definition 2. Since A is a preferred assignment at t_0, $W_i^{a_i - 1}(t_0) \geq W_j^{a_j}(t_0)$. Therefore, line $W_i^{a_i - 1}(t)$ and line $W_j^{a_j}(t)$ must intersect at some time $t \in [t_0, t_2]$. Then $t_1 > t_2 \geq \min(\mathbf{I}(A, t_0))$, which contradicts our assumption of t_1.

Since A is a preferred assignment at t_0, thus we have $\max_{j \in \mathcal{T}} W_j^{a_j}(t_0) < W_i^{a_i - 1}(t_0)(\forall i \in \mathcal{T}, a_i > 0)$. If no targets i, j satisfy that $\frac{\partial W_i^{a_i - 1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$ at any $t \in [t_0, t_e]$, $\max_{j \in \mathcal{T}} W_j^{a_j}(t) \leq W_i^{a_i - 1}(t)(\forall i \in \mathcal{T}, t \in [t_0, t_e])$. Thus A is a preferred assignment till t_e. Similarly, if there do not exist any targets i, j such that line $W_i^{a_i - 1}(t)$ and line $W_j^{a_j}(t)$ intersects in $[t_0, t_e]$, A is a preferred assignment till t_e. □

Proposition 3 lets us deduce that the optimal defender strategy over continuous time can be computed when only considering finite time points at which the PA changes. Consider Example 1 again. Figure 4 shows that at time 0, $A = \langle 1, 0\rangle$ is a PA since $W_1^0(0) > \max\{W_1^1(0), W_2^0(0)\}$. t_1 in Figure 4 represents the intersection of line $W_1^0(t)$ and $W_2^0(0)$, which is the only element in $\mathbf{I}(A, 0)$, and hence the minimum value. Thus A is a PA in $[0, t_1]$. We can directly see from Figure 4 that A is not a PA any more after t_1. Next, we explore how to convert an 'expiring' PA to a new PA until we compute PAs for all time points.

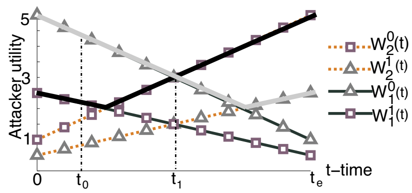


Figure 4: Minimax assignment

Proposition 4. *If assignment A is a PA at time t and there exist targets i, j such that $W_i^{a_i-1}(t) = W_j^{a_j}(t)$, then an assignment A' with $a'_i = a_i - 1, a'_j = a_j + 1, a'_k = a_k (\forall k \in \mathcal{T}, k \neq i, j)$ is also a PA at t.*

Proof. Since only the number of resources assigned to i decreases and the number of resources assigned to j increases, to prove A' is a preferred strategy at t, we just need to prove that $W_j^{a'_j-1}(t) \geq W_k^{a'_k}(t)$ and $W_k^{a'_k-1}(t) \geq W_i^{a'_i}(t) (\forall k \in \mathcal{T}, a'_k > 0)$. First, given $W_j^{a'_j-1}(t) = W_j^{a_j}(t) = W_i^{a_i-1}(t)$ based on our assumptions and $W_i^{a_i-1}(t) \geq W_k^{a'_k}(t)$ based on Definition 2, we have $W_j^{a'_j-1}(t) \geq W_k^{a'_k}(t)$. Similarly, given $W_k^{a'_k-1}(t) W_k^{a_k-1}(t) \geq W_j^{a_j(t)}$ based on Definition 2 and $W_j^{a_j(t)} = W_i^{a_i-1}(t) = W_i^{a'_i}(t)$ based on our assumptions, we have $W_k^{a'_k-1}(t) \geq W_i^{a'_i}(t)$. □

Figure 4 shows that in Example 1, $A = \langle 1, 0\rangle$ at t_1 is a PA. Since $W_1^{a_1-1}(t_1) = W_1^0(t_1) = W_2^{a_2}(t_1) = W_2^0(t_1)$, $A' = \langle 0, 1\rangle$ is also a minimax assignment at t_1. For A', after t_1, there are no targets i, j such that $\frac{\partial W_i^{a_i-1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$. Thus A' is the PA till t_e based on Proposition 3.

We exploit these propositions in the SCOUT-A algorithm (Scheduling seCurity resOurces in pUblic evenTs with no relocating delAy). SCOUT-A is built on the following ideas: 1) Compute a PA at current time. 2) Compute the time point at which this assignment stops being a PA, set this time point as current time. 3) Convert this assignment to a new PA based on Proposition 3, then repeat step 2 and step 3 till for all targets pair i, j, line $W_i^{a_i-1}(t)$ and line $W_j^{a_j}(t)$ do not intersect between current time and t_e. In this way, we can find an optimal defender strategy for the continuous time strategy space in which all assignments are PAs.

We now discuss SCOUT-A described in Algorithm 1 line by line. Lines 1-5 compute the PA at time 0. Firstly, all targets are not protected (Line 1). Then a resource is assigned

Algorithm 1: SCOUT-A

1 **for** $i \in \mathcal{T}$ **do**
2     $V_i \leftarrow v_i(0), a_i \leftarrow 0$
3 $left \leftarrow m$
4 **while** $left > 0$ **do**
5     $i \leftarrow \text{argmax}_{i \in \mathcal{T}} V_i$, a_i++, $left$- -, $V_i \leftarrow W_i^{a_i}(0)$
6 $t_m \leftarrow 0, k \leftarrow 0$
7 **while** $t_m < t_e$ **do**
8     $\mathbf{I} \leftarrow \emptyset$
9   **for** $\forall i, j \in \mathcal{T}$ **do**
10   **if** $\exists t$ *such that* $\frac{\partial W_i^{a_i-1}(t)}{\partial t} < \frac{\partial W_j^{a_j}(t)}{\partial t}$ **then**
11             $I_{ij} \leftarrow I_{ij}(A, t_m)$
12   **if** $I_{ij} < t_e$ **then** $\mathbf{I} \leftarrow \mathbf{I} \cup I_{ij}$;
13   **if** $\mathbf{I} = \emptyset$ **then** break;
14     $I_{ij} \leftarrow \min(\mathbf{I})$
15   **if** $W_i^{a_i-1}(I_{ij} - \Delta t) \geq W_j^{a_j}(I_{ij} - \Delta t)$ **then**
16         $\tau_k \leftarrow I_{ij}, t_m \leftarrow I_{ij}$
17         $a_i - -, a_j + +, c_{ij}^k \leftarrow 1, k + +$

to the target with the highest payoff until all resources are assigned (Lines 3-5). t_m in Line 6 is used to record when the current PA stops being a PA. k is used to record how many transfers have been made. Lines 7-14 compute the next time point to transfer resources based on Proposition 3. If no intersections of $W_i^{a_i-1}(t)$ and $W_j^{a_j}(t)$ exist before t_e, the current assignment is the PA until t_e, then the iteration is terminated (Line 13). Otherwise, I_{ij} records the time at which the PA changes (Line 14). Lines 15-17 change the PA based on Proposition 4. In line 15, Δt is a number smaller than the interval of any adjacent intersections of line $W_i^{a_i}(t)$ and $W_j^{a_j}(t) (\forall i, j \in \mathcal{T}, \forall a_i, a_j \in \{0, \ldots, m\})$. Line 15 checks whether the assignment is a PA immediately before I_{ij}. This is to avoid the algorithm repeatedly exchanging two assignments at one time point. Since SCOUT-A computes a defender strategy in which all assignments are minimax assignments at any time during $[0, t_e]$, this defender strategy is optimal based on Proposition 2. The time complexity of SCOUT-A is polynomial in the number of resources, targets, and the number of linear segments in value functions.

Equilibrium Considering Transfer Time

In this section, we propose algorithms to calculate optimal defender strategies when the time needed to transfer resources is not negligible, i.e., $d_{ij} \neq 0$. When a resource is in transfer, this resource is not assigned to any target.

Discrete Defender Strategy Space

We begin by assuming that the defender can only begin transferring resources at discrete time points, denoted by a set $\Phi = \{t_k\}$. Thus a resource may arrive at a target only at time points in a set $\phi = \{t_\delta : t_\delta = t_k + d_{ij}, \forall t_k \in \Phi, \forall i, j \in \mathcal{T}\}$. Denote $\Psi = \{t_\eta : t_\eta \in \Phi \,\text{or}\, t_\eta \in \phi\}$ with elements in increasing order. Let η be an index specifying a $t_\eta \in \Psi$. Let $H = |\Psi|$. For $\eta \in \{1, \ldots, H\}$, denote a vector $\sigma_\eta = \langle \sigma_{ij}^\eta : \sigma_{ij}^\eta \in \{1, \ldots, H\}\rangle$ with σ_{ij}^η meaning that if a resource is transferred from target i to target

j, and is supposed to arrive at target j at t_η, then the transfer should begin at time $t_{\sigma_{ij}^{\eta}}$. Let $a_i^{t_\eta}$ represent the number of resources assigned to target i at time t_η before any resource is transferred from target i to other targets. Let $b_i^{t_\eta}$ represent the number of resources left at target i at time t_η after the transfers. We propose SCOUT-D (Scheduling seCurity resOurces in pUblic evenTs with Discrete defender strategy space) to compute the optimal defender strategy. SCOUT-D can be formulated as follows.

$$\min U \tag{3}$$

$$\mathbf{s.t.} \quad \sum\nolimits_{i\in\mathcal{T}} a_i^0 = m \tag{4}$$

$$a_i^{t_{k+1}} = b_i^{t_k} + \sum\nolimits_{j\in\mathcal{T}} c_{ji}^{\sigma_{ji}^{k+1}} \quad \forall k \in \{1,\ldots,H-1\} \tag{5}$$

$$b_i^{t_k} = a_i^{t_k} - \sum\nolimits_{j\in\mathcal{T}} c_{ij}^{k} \quad \forall k \in \{1,\ldots,H\} \tag{6}$$

$$c_{ij}^k \in \{0,1,\ldots\} \quad \forall i,j\in\mathcal{T}, \forall k \in \{1,\ldots,H\} \tag{7}$$

$$\sum c_{ij}^{t_\eta} = 0 \quad \forall t_\eta \in \phi \text{ and } t_\eta \notin \Phi \tag{8}$$

$$\sum c_{ij}^{\sigma_{ij}^{\eta}} = 0 \quad \forall t_\eta \in \Phi \text{ and } t_\eta \notin \phi \tag{9}$$

$$b_i^{t_k} \geq 0 \quad \forall i\in\mathcal{T}, \forall k \in \{1,\ldots,H\} \tag{10}$$

$$U \geq \max_{t\in[t_k,t_{k+1}]} W_i^{b_i^{t_k}}(t) \quad \forall i\in\mathcal{T}, \forall k \in \{1,\ldots,H-1\} \tag{11}$$

We now describe the SCOUT-D program. Eq.(4) implements feasibility of initial assignment. In Eq.(5), $\sum_{j\in\mathcal{T}} c_{ji}^{\sigma_{ji}^{k+1}}$ represents the number of resources arriving at target i at t_{k+1}. Similarly, in Eq.(6), $\sum_{j\in\mathcal{T}} c_{ij}^k$ represents the number of resources transferred from target i to other targets at t_k. Eq.(7) restricts that the number of transferred resources should be an integer. Eqs.(5)-(7) together enforce the feasibility of resource transfers, much like the flow feasibility constraints in a network flow setting. Eq.(8) deals with time points in ϕ but not in Φ, at which begining transfers are not allowed. Similarly, Eq.(9) deals with time points in Φ but not in ϕ, at which no resources arrive at any target. Eq.(11) forces U to be larger than the optimal attacker utility if the attacker could attack at any time. Since $v_i(t)$ is a continuous function, $\max_{t\in[t_k,t_{k+1}]} W_i^{b_i^{t_k}}(t)$ must exist. The objective of the program and Eq.(11) together guarantee that the defender strategy minimizes the maximum attacker utility when the attacker strategy is continuous.

Computing Exact Equilibrium

When the defender strategy space is continuous, the defender can transfer resources at any time in $[0, t_e]$. However, some transfers may not be beneficial and can be replaced by other transfers without affecting the optimal defender utility. Consequently, it is possible to set time points in Φ appropriately, thus the defender strategy calculated by SCOUT-D is also an equilibrium strategy when the defender strategy space is continuous. In this section, we first prove that for any game with continuous defender strategy space, there must exist an equilibrium defender strategy in which the defender only transfers resources at some time points satisfying specific conditions. We then propose an algorithm to compute these time points, which will be passed to SCOUT-D to compute the optimal strategy.

We consider a continuous value function as a series of monotonic segments. For each target i, let $\xi_1^i, \xi_2^i, \ldots, \xi_{R_i}^i$ represent all time points at which the monotonicity of value function $v_i(t)$ changes (in increasing order). Let $\xi_0^i = 0$ and $\xi_{R_i+1}^i = t_e$. Denote a set $\Xi^i = \{\xi_\rho^i : \rho \in \{0, \ldots R_i + 1\}\}$ in which ρ is an index identifying a particular time point $\xi_\rho^i \in \Xi^i$. Thus for the time period between any adjacent time points in Ξ^i, $v_i(t)$ is monotonic. Denote $\Xi = \{\Xi^i : \forall i \in \mathcal{T}\}$.

If for any optimal defender strategy S of a game with continuous defender strategy space, we can convert S to anther optimal strategy in which transfers only occur at specific time points, then we can compute the optimal defender strategy of such games using SCOUT-D by adding all qualified time points into Φ. Next, we describe the process with which we convert an optimal strategy S. We begin with showing that some transfers in S can be combined without affecting the optimal attacker utility in Proposition 5.

Proposition 5. *If in S, a resource is transferred from target i to target j at time t_1 then from target j to target l at time t_2, if $t_1 + d_{ij} \in [\xi_\rho^j, \xi_{\rho+1}^j)$ and $t_2 \in [t_1 + d_{ij}, \xi_{\rho+1}^j)$, then the optimal attacker utility will not be higher if the resource is directly transferred from target i to target l at time $t_3 \in [t_1, t_2 + d_{jl} - d_{il}]$.*

Proof. Assume that in S, before the transfer at t_1, the number of resources assigned to target i, j, l is a_i, a_j, a_l respectively. In S, the number of resources assigned to these three targets with respect to time is shown in Table 1.

time period	$q_i^t(S)$	$q_j^t(S)$	$q_l^t(S)$
right before t_1	a_i	a_j	a_l
$[t_1, t_1 + d_{ij})$	$a_i - 1$	a_j	a_l
$[t_1 + d_{ij}, t_2)$	$a_i - 1$	$a_j + 1$	a_l
$[t_2, t_2 + d_{jl})$	$a_i - 1$	a_j	a_l
right after $t_2 + d_{jl}$	$a_i - 1$	a_j	$a_l + 1$

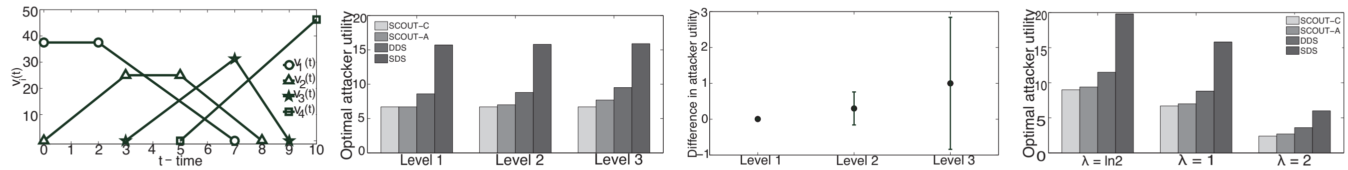
Table 1: Resources assigned to i, j, l in S

Note that $d_{il} \leq d_{ij} + d_{jl}$, otherwise the defender could transfer resources from i to l passing j. Therefore, there always exists $t_3 \in [t_1, t_2 + d_{jl} - d_{il}]$. We now construct a defender strategy S^1. In S^1 everything else is the same as S except that a resource is directly transferred from target i to target l at t_3. Table 2 shows the time periods and $q_i^t(S^1), q_j^t(S^1), q_l^t(S^1)$.

time period	$q_i^t(S^1)$	$q_j^t(S^1)$	$q_l^t(S^1)$
right before t_3	a_i	a_j	a_l
$[t_3, t_3 + d_{il})$	$a_i - 1$	a_j	a_l
right after $t_3 + d_{il}$	$a_i - 1$	a_j	$a_l + 1$

Table 2: Resources assigned to i, j, l in S^1

From Table 1 and Table 2, we can see that only during $[t_1 + d_{ij}, t_2)$, the number of resources assigned to j in S^1 is less than the number of resources assigned to j in S. For any other time t and any



(a) Value functions (b) Varying transfer time (c) SCOUT-C v.s. SCOUT-A (d) Varying λ value

Figure 5: Experimental settings and solution quality

other target x, $q_x^t(S^1) \geq q_x^t(S)$. Therefore, only when $\max_{t\in[t_1+d_{ij},t_2)} W_j^{q_t^j(S^1)}(t) > U^a(f_{tg}(S), f_{tm}(S), S)$, the optimal attacker utility against S^1 can be higher than the optimal attacker utility against S. However, since $v_j(t)$ is monotonic in $[\xi_\rho^j, \xi_{\rho+1}^j]$, thus $\max_{t\in[t_1+d_{ij},t_2)} W_j^{a_j}(t) \leq \max\{W_j^{a_j}(t_1+d_{ij}), W_j^{a_j}(t_2)\} \leq U^a(f_{tg}(S), f_{tm}(S), S)$. Therefore, the optimal attacker utility against S^1 will not be higher than that against S. □

Based on Proposition 5 (proof is given in Appendix), we can convert S to a new optimal strategy S^1. Next, we show that we can replace transfers in S^1 with other transfers which occur at time points satisfying specific conditions.

Assume that in S^1, a resource is transferred from target i at $t_1 \in [\xi_\rho^i, \xi_{\rho+1}^i)$ and this resource arrives at target j at $t_2 \in [\xi_{\rho'}^j, \xi_{\rho'+1}^j)$. Assume that right before t_1, the number of resources assigned to i is a_i. Also assume that right before t_2, the number of resources assigned to j is a_j. Then this transfer can be represented by $Tr = (i, j, a_i, a_j, \rho, \rho')$. To explore how to replace Tr, we first denote a specific time point $\theta(Tr)$ as follows.

$$\theta(Tr) = \operatorname*{argmin}_{t\in E}\left(\max_{t'\in[t,t+d_{ij}]}\{W_i^{a_i-1}(t'), W_j^{a_j}(t')\}\right), \quad (12)$$

where $E = [\xi_\rho^i, \min\{\xi_{\rho+1}^i, \xi_{\rho'+1}^j - d_{ij}\}]$. $\theta(Tr)$ means that, given that a_i resources are assigned to target i and a_j resources are assigned to target j, if a resource is transferred from target i to target j during $[\xi_\rho^i, \xi_{\rho+1}^i]$ and the resource arrives at j during $[\xi_{\rho'}^j, \xi_{\rho'+1}^j]$, then beginning the transfer at $\theta(Tr)$ can minimize the maximum attacker utility if the attacker attacks i or j when the resource is in transfer. Next, we show how to replace a transfer $Tr = (i, j, a_i, a_j, \rho, \rho')$ in S^1 with a transfer starting at $\theta(Tr)$.

Proposition 6. *Keeping all else in S^1 unchanged, the optimal attacker utility will not be higher if the beginning time of transfer Tr in S^1 is changed to $\theta(Tr)$.*

Proof. Assume that in S^1, the beginning time of Tr is t_1. We construct a new strategy S^2 in which everything is the same as S^1 except that the beginning time of Tr is changed to $\theta(Tr)$. Next, we prove Proposition 6 under the assumption that $t_1 < \theta(Tr)$. If $t_1 \geq \theta(Tr)$, it can be proved in a similar way. When $t_1 < \theta(Tr)$, only when $t \in (t_1 + d_{ij}, \theta(Tr) + d_{ij})$, the number of resources assigned to target j is one less in S^2 than that in S^1. For any other target k and any other time t, $q_k^t(S^2) \geq q_k^t(S^1)$. Therefore, the optimal attacker utility in S^2 is higher than that in S^1 only when $\max_{t\in(t_1+d_{ij},\theta(Tr)+d_{ij})} W_j^{q_t^j(S^2)}(t) > U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$.

Given the definition of $\theta(Tr)$, we have $\theta(Tr) + d_{ij} \in [\xi_{\rho'}^j, \xi_{\rho'+1}^j]$. The property of S^1 ensures that no resource is transferred from target j to others during $[t_1+d_{ij}, \xi_{\rho'+1}^j]$. Let ω be the interval of $[t_1+d_{ij}, \theta(Tr)+d_{ij}]$, $q_j^t(S^2) \geq a_j$ when $t \in \omega$. Therefore, $\max_{t\in\omega} W_j^{q_t^j(S^2)}(t) \leq \max_{t\in\omega} W_j^{a_j}(t)$. To show $\max_{t\in\omega} W_j^{a_j}(t) \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$, we consider the following two situations.

Case 1 ($t_1 + d_{ij} \geq \theta(Tr)$): We have $\max_{t\in\omega} W_j^{a_j}(t) \leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} W_j^{a_j}(t) \leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} \{W_i^{a_i-1}(t), W_j^{a_j}(t)\}$. Based on the definition of $\theta(Tr)$, we know that $\max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]}\{W_i^{a_i-1}(t), W_j^{a_j}(t)\} \leq \max_{t\in[t_1,t_1+d_{ij}]}\{W_i^{a_i-1}(t), W_j^{a_j}(t)\} \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$. It then follows that $\max_{t\in\omega} W_j^{q_t^j(S^2)}(t) \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$, the optimal attacker utility against S^2 is not higher than that against S^1.

Case 2 ($t_1+d_{ij} < \theta(Tr)$): If $\max_{t\in[t_1+d_{ij},\theta(Tr)]} W_j^{a_j}(t) \leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} W_j^{a_j}(t)$, then $\max_{t\in\omega} W_j^{a_j}(t) \leq \max_{t\in[\theta(Tr),\theta(Tr)+d_{ij}]} W_j^{a_j}(t)$, thus we can prove $\max_{t\in\omega} W_j^{a_j}(t) \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$ as in **Case 1**. Otherwise, since $v_j(t)$ is monotonic in $[\xi_{\rho'}^j, \xi_{\rho'+1}^j]$, we have $\frac{\mathrm{d}W_j^{a_j}(t)}{\mathrm{d}t} \leq 0$ for $t \in [\xi_{\rho'}^j, \xi_{\rho'+1}^j]$. Thus $\max_{t\in\omega} W_j^{a_j}(t) \leq W_j^{a_j}(t_1 + d_{ij}) \leq U^a(f_{tg}(S^1), f_{tm}(S^1), S^1)$. □

Denote a set $\Theta = \{\theta(Tr) : Tr = (i, j, a_i, a_j, \rho, \rho'), \forall i, j \in \mathcal{T}, \forall a_i, a_j \in \{0, \ldots m\}, \forall \rho \in \{0, \ldots, R_i\}, \forall \rho' \in \{0, \ldots, R_j\}\}$. Proposition 6 implies that we can convert S^1 to another optimal strategy S^2 in which all transfers occur at $t \in \Theta$. In other words, as long as a game with continuous defender strategy space has an equilibrium defender strategy S, it must also have an equilibrium defender strategy in which transfers only begin at $t \in \Theta$. We propose SCOUT-C (Scheduling seCurity resOurces in pUblic evenTs against Continuous strategy space) in Algorithm 2 to compute the optimal defender strategy. SCOUT-C first computes all the time points at which a transfer may begin (Line 5) and a transfer may end (Line 6) in an optimal defender strategy, recording all these time points in a set Ψ. Then SCOUT-D is called to solve the game using Ψ as the time point set, resulting in an optimal defender strategy when the defender strategy space is continuous.

Algorithm 2: SCOUT-C

$\Psi \leftarrow \emptyset$
for $i, j \in \mathcal{T}$ **do**
  for $a_i \in \{0, \ldots, m\}, a_j \in \{0, \ldots, m\}$ **do**
    for $\rho \in \{0, \ldots, R_i\}, \rho' \in \{0, \ldots, R_j\}$ **do**
      $\Psi \leftarrow \Psi \cup \{\theta(Tr)\} \cup \{\theta(Tr) + d_{ij}\}$, where $Tr = (i, j, a_i, a_j, \rho, \rho')$
run SCOUT-D, using Ψ as the time points set

Experimental Evaluation

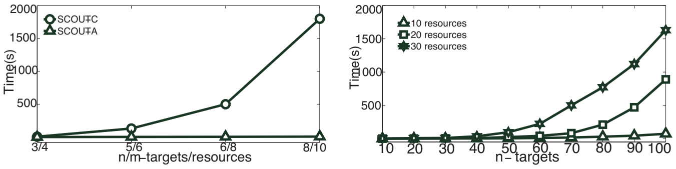
We compare the performance of SCOUT-A and SCOUT-C in terms of attacker utility. As it is a zero-sum game, a lower attacker utility indicates a higher defender utility. We consider two baseline strategies. The first one, SDS, is a static defender strategy in which defender assigns resources to targets at the beginning of the event without transferring them any more. In SDS, the number of resources assigned to each target is in proportion with its maximum value during the event. The second one, which we call DDS, is a dynamic defender strategy with a discrete defender strategy space. We use SCOUT-D to compute a DDS for a game, setting the time point set as $\Phi = \{\frac{n \cdot t_e}{4} : n \in \{0, 1, \ldots 4\}\}$.

All experiments are averaged over 50 sample games. Unless otherwise specified, we use 4 targets, 5 security resources, $t_e = 10$, $\lambda = 1$ to describe the marginal utility of an extra security resource. We use KNITRO version 8.0.0 to solve SCOUT-D. We provide 4 experiment sets that compare the performance of SCOUT-A, SCOUT-C and the baseline strategies against: 1) Different levels of transfer time. 2) Different numbers of targets and resources. 3) Different values of λ. 4) More general forms of targets' value functions.

Figure 5(a) shows an example of the piecewise linear value functions of 4 different targets where the x-axis indicates the time and the y-axis is the value of each target. Each value function has at most three linear segments. We randomly choose a time period in $[0, t_e]$ in which a value function is non-zero. We constrain that the value function is continuous and that the maximum value of each value function is less than 50 while the minimum value is no less than 0.

In the first experiment set, we consider three different levels of transfer time. For level 1, the time needed to transfer resources between any two targets d_{ij} is uniformly generated in $[0, 0.1]$. For level 2, d_{ij} is in $[0, 1]$. For level 3, d_{ij} is in $[0, 5]$. Figure 5(b) shows the solution quality of the algorithms against different transfer time levels. In Figure 5(b), the x-axis indicates three transfer time levels while the y-axis shows the expected attacker utility. The results show that SCOUT-C and SCOUT-A outperform DDS and SDS despite the transfer time level. It is noticeable that arbitrarily discretizing time may lead to significant loss in time-critical domains, e.g., when proceeding time is averagely divided as 4 points (DDS) and transfer time is Level 2, the attacker utility is around 30% higher than that against SCOUT-C, i.e., the loss in defender utility is around 30%.

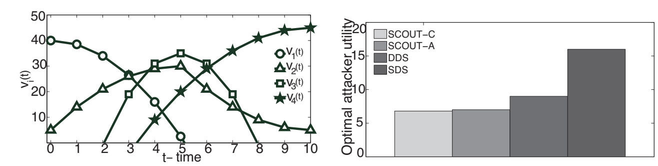
Figure 5(c) shows the average difference (and the error bars) between the solution quality of SCOUT-C and SCOUT-A with different transfer time levels (higher means SCOUT-C is better). The results show that when transfer time is small (Level 1), the solution quality of SCOUT-A is quite close to that of SCOUT-C. As the transfer time increases, solution quality of SCOUT-A becomes worse compared to SCOUT-C. However, from Figures 5(b) and 5(c), we can see that the difference between defender utility against SCOUT-A and the optimal defender utility computed by SCOUT-C is less than 5% under the Level 2 transfer time.



(a) SCOUT-C and SCOUT-A (b) SCOUT-A

Figure 6: Runtime

In the second experiment set, we fix transfer time to Level 2. Figure 6(a) shows the runtime of SCOUT-A and SCOUT-C. The x-axis in Figure 6(a) indicates several combinations of different numbers of targets and resources while the y-axis shows the runtime. Figure 6(b) extends the data out for SCOUT-A, with the x-axis indicating the number of targets and the y-axis indicating runtime. The three marked lines in Figure 6(b) represent the number of resources to be 10, 20 and 30 respectively. For example, the point (100, 890) on the line marked by squares indicates that SCOUT-A has an average runtime of 890s on problems with 100 targets and 20 resources. SCOUT-A can solve a game with 100 targets and 30 resources within 30 minutes.



(a) Quadratic value functions (b) Solution quality

Figure 7: Expansions of general-form value functions

In the third experiment set, shown in Figure 5(d), we show the effect of changing the value of λ. We still fix the number of targets and resources to 4 and 5. Intuitively, a larger λ means that the marginal utility of adding a security resource is higher. There are three sets of results, corresponding to $\lambda = ln2, \lambda = 1, \lambda = 2$ respectively. The expected attacker utility is shown by the y-axis. SCOUT-A and SCOUT-C outperform DDS and SDS in spite of the value of λ.

Finally, we consider value functions of quadratic segments. Figure 7(a) shows an example of value functions of 4 targets. Figure 7(b) shows the solution quality of SCOUT-A, SCOUT-C, DDS, and SDS against such value function forms. SCOUT-A and SCOUT-C still significantly outperform DDS and SDS, similar to the results for piecewise linear value functions.

Conclusions

In this paper, we design a new defender-attacker game model with continuous strategies for both agents to describe dynamic patrolling in domains such as public events. We present SCOUT-A to solve the game when the resources can be relocated without any delay in time. We also propose SCOUT-C to deal with general cases. Both SCOUT-A and SCOUT-C compute the optimal defender strategy despite the continuous strategy space and the time-dependent payoff of an attack. Our detailed experimental results show that both SCOUT-A and SCOUT-C outperform the defender strategies observed in the real world.

Acknowledgements

This work is supported by NSFC grant No. 61202212 and Singapore MOE AcRF Tier 1 grant MOE RG33/13. This research is also supported in part by Interactive and Digital Media Programme Oce, National Research Foundation hosted at Media Development Authority of Singapore (Grant No.: MDA/IDM/2012/8/8-2 VOL 01).

References

Agmon, N.; Urieli, D.; and Stone, P. 2011. Multiagent patrol generalized to complex environmental conditions. In *Proceedings of The 25th AAAI Conference on Artificial Intelligence*, 1090–1095.

An, B.; Kempe, D.; Kiekintveld, C.; Shieh, E.; Singh, S.; Tambe, M.; and Vorobeychik, Y. 2012. Security games with limited surveillance. In *Proceedings of The 26th AAAI Conference on Artificial Intelligence*, 1241–1248.

An, B.; Brown, M.; Vorobeychik, Y.; and Tambe, M. 2013a. Security games with surveillance cost and optimal timing of attack execution. In *Proceedings of The 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 223–230.

An, B.; Ordónez, F.; Tambe, M.; Shieh, E.; Yang, R.; Baldwin, C.; DiRenzo III, J.; Moretti, K.; Maule, B.; and Meyer, G. 2013b. A deployed quantal response-based patrol planning system for the US Coast Guard. *Interfaces* 43(5):400–420.

Basilico, N.; Gatti, N.; and Amigoni, F. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 57–64.

Fang, F.; Jiang, A. X.; and Tambe, M. 2013. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of The 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 957–964.

Kiekintveld, C.; Islam, T.; and Kreinovich, V. 2013. Security games with interval uncertainty. In *Proceedings of The 2013 international conference on Autonomous agents and multi-agent systems (AAMAS)*, 231–238.

Korzhyk, D.; Conitzer, V.; and Parr, R. 2011. Solving Stackelberg games with uncertain observability. In *Proceedings of The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1013–1020.

Letchford, J., and Vorobeychik, Y. 2013. Optimal interdiction of attack plans. In *Proceedings of The 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 199–206.

Nicola Basilico, Nicola Gatti, F. A. 2012. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence* 78–123.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: A deployed game theoretic system to protect the ports of the United States. In *Proceedings of The 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 13–20.

Tambe, M. 2011. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

Varakantham, P.; Lau, H. C.; and Yuan, Z. 2013. Scalable randomized patrolling for securing rapid transit networks. In *Proceedings of The 25th Innovative Applications of Artificial Intelligence Conference (IAAI)*, 1563–1568.

Yang, R.; Ordonez, F.; and Tambe, M. 2012. Computing optimal strategy against quantal response in security games. In *Proceedings of The 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 847–854.

Yin, Z.; Jiang, A. X.; Johnson, M. P.; Kiekintveld, C.; Leyton-Brown, K.; Sandholm, T.; Tambe, M.; and Sullivan, J. P. 2012. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In *Proceedings of The 24th Innovative Applications of Artificial Intelligence (IAAI)*, 2348–2355.