Security Games with Limited Surveillance: An Initial Report

Bo An, David Kempe
University of Southern California
Los Angeles, CA 90089
boa,d kempe}@usc.edu

Christopher Kiekintveld
University of Texas, El Paso
El Paso, TX 79968
cdkiekintveld@utep.edu

Eric Shieh
University of Southern California
Los Angeles, CA 90089
eshieh@usc.edu

Satinder Singh
University of Michigan
Ann Arbor, MI 48109
baveja@umich.edu

Milind Tambe
University of Southern California
Los Angeles, CA 90089
tambe@usc.edu

Yevgeniy Vorobeychik*
Sandia National Laboratories
Livermore, CA 94550
eug.vorobey@gmail.com

Abstract

Stackelberg games have been used in several deployed applications of game theory to make recommendations for allocating limited resources for protecting critical infrastructure. The resource allocation strategies are randomized to prevent a strategic attacker from using surveillance to learn and exploit patterns in the allocation. An important limitation of previous work on security games is that it typically assumes that attackers have perfect surveillance capabilities, and can learn the exact strategy of the defender. We introduce a new model that explicitly models the process of an attacker observing a sequence of resource allocation decisions and updating his beliefs about the defender’s strategy. For this model we present computational techniques for updating the attacker’s beliefs and computing optimal strategies for both the attacker and defender, given a specific number of observations. We provide multiple formulations for computing the defender’s optimal strategy, including non-convex programming and a convex approximation. We also present an approximate method for computing the optimal length of time for the attacker to observe the defender’s strategy before attacking. Finally, we present experimental results comparing the efficiency and runtime of our methods.

Introduction

Stackelberg games have been used in several deployed applications of game theory to make recommendations for allocating limited resources for protecting critical infrastructure (Basilico, Gatti, and Amigoni 2009; Korzhyk, Conitzer, and Parr 2010; Dickerson et al. 2010; Tambe 2011; An et al. 2011b). A Stackelberg security game models an interaction between an attacker and a defender (Kiekintveld et al. 2009). The defender first commits to a security policy (which may be randomized), and the attacker is able to use surveillance to learn about the defender’s policy before launching an attack. A solution to the game yields an optimal randomized strategy for the defender, based on the assumption that the attacker will observe this strategy and respond optimally. Software decision aids based on Stackelberg games have been implemented in several real-world domains, including LAX (Los Angeles International Airport) (Pita et al. 2008), FAMS (United States Federal Air Marshals Service) (Tsi et al. 2009), TSA (United States Transportation Security Agency) (Pita et al. 2011), and the United States Coast Guard (An et al. 2011a).

Most of the existing work on security games (including the methods used in the deployed applications listed above) assumes that the attacker is able to observe the defender’s strategy perfectly. In reality, the attacker may have more limited observation capabilities, and our goal in this research is to develop models that capture some of these limitations in a more realistic way. Terrorists conduct surveillance to select potential targets and gain strong situational awareness of targets’ vulnerabilities and security operations (Southers 2011). One important limitation is the number of observations an attacker can make; it is not possible to conduct surveillance for an infinite period of time. Attackers may also wish to reduce the number of observations due to the risk of being detected by security forces during surveillance activities (Southers 2011). Therefore, it is important to consider situations where attackers select targets based on limited numbers of observations using explicit belief updates.

There has been some recent work that relaxes the perfect observation assumption in security games. RECON (Yin et al. 2011) takes into account possible observation errors by assuming that the attacker’s observation is within some distance from the defender’s real strategy, but does not address how these errors arise or explicitly model the process of forming beliefs based on limited observations. The COBRA algorithm (Pita et al. 2010) focuses on human perception of probability distributions by applying support theory (Tversky and Koehler 1994) from psychology. Both RECON and COBRA require hand-tuned parameters to model observations errors, which we avoid in this paper. Yin et. al (2010) prove the equivalence of Stackelberg equilibria and Nash equilibria for some classes of security games. In general, however, Stackelberg and Nash equilibria may differ in security games, and the optimal strategy in cases with limited surveillance may be different than both. There also has been some work on understanding the value of commitment for the leader in general Stackelberg games where observations are limited or costly (Bagwell 1995;
The important difference between previous work and the methods we develop in this paper is that we consider a more detailed model of how attackers conduct surveillance operations and update their beliefs about the defender’s strategy. We make the following contributions to this line of work: (1) We introduce a model of security games with strategic surveillance and formulate how the attacker updates his belief given limited observations. (2) We provide multiple formulations for computing the defender’s optimal strategy, including non-convex programming and a convex approximation. (3) We provide an approximate approach for computing the optimal number of observations for the attacker. (4) We present experimental results comparing the efficiency and runtime of the methods we develop.

Stackelberg Security Games

A Stackelberg security game has two players, a defender who decides how to use \( m \) identical resources to protect a set of targets \( T = \{t_1, t_2, \ldots, t_{|T|} \} \) (\( m < |T| \)), and an attacker who selects a single target to attack. The defender’s pure strategies are all possible feasible assignments of the security resources to targets, with at most \( m \) targets from \( T \) protected by a single resource each. The defender’s mixed strategies consist of all probability distributions over these pure strategies. The attacker’s pure strategies coincide with the set of targets that can be attacked (\( T \)). In a Stackelberg game we assume that the attacker is able to (perfectly) observe the defender’s mixed strategy before selecting a target to attack.

Let \( A = \{A_i\} \) be a set of feasible resource assignments, where \( A_i \) is the defender’s \( i^{th} \) pure strategy. If \( A_{ij} = 1 \), target \( t_j \) is covered by the defender in assignment \( A_i \), and \( A_{ij} = 0 \) otherwise. We denote a mixed strategy for the defender by \( x = \langle x_i \rangle \) where \( x_i \) is the probability of choosing \( A_i \). In many cases, we can use a compact representation for this mixed strategy (Kiekintveld et al. 2009). The strategy is represented using a marginal coverage vector \( c = \langle c_j \rangle \) where \( c_j = \sum_{i \in A} x_i A_{ij} \) is the probability that target \( t_j \) is covered by some defender resource. The attacker’s mixed strategy is a vector \( a = \langle a_j \rangle \) where \( a_j \) is the probability of attacking target \( t_j \).

The payoffs for each player depend on which target is attacked and the probability that the target is covered by the defender. Given a target \( t_j \), the defender receives payoff \( R^d_j \) if the adversary attacks \( t_j \) and it is covered; otherwise, the defender receives payoff \( P^d_j \). The attacker receives payoff \( P^a_j \) in the former case and payoff \( R^a_j \) in the latter case. We assume that \( R^d_j > P^d_j \) and \( R^a_j > P^a_j \), so adding resources to cover a target hurts the attacker and helps the defender. For a strategy profile \((c, a)\), the expected utilities for both agents are given by (notations are listed in Table 1):

\[
U_d(c, a) = \sum_{t_j \in T} a_j U_d(c, t_j), \quad U_d(c, t_j) = c_j R^d_j + (1 - c_j) P^d_j
\]
\[
U_a(c, a) = \sum_{t_j \in T} a_j U_a(c, t_j), \quad U_a(c, t_j) = c_j P^a_j + (1 - c_j) R^a_j
\]

In a Stackelberg model, the defender chooses a strategy first, and the attacker chooses a strategy after observing the defender’s strategy. The standard solution concept for Stackelberg games is Strong Stackelberg Equilibrium (SSE) (Breton, Alg. and Haurie 1988; Leitmann 1978; von Stengel and Zamir 2004). An SSE requires that the attacker will choose his best target(s) in response to the defender’s strategy, with ties broken optimally for the defender if there are multiple best responses for the attacker. Since there always exists an optimal pure-strategy response for the attacker, we restrict the attacker’s strategies to pure strategies without loss of generality in this case.

We now introduce a new model that moves away from the Stackelberg model of perfect observation for security games. We call this class of games as ‘security games with strategic surveillance (SGSS)’. In our model, the attacker makes a limited number of observations. The attacker may decide the number of observations to make strategically, considering the cost of conducting surveillance. The sequence of moves in an SGSS is as follows:

1. The attacker first decides how many observations to make (denoted by \( \tau \)), considering the subsequent game and the cost incurred while making observations.
2. Next, the defender chooses a strategy considering the attacker’s prior beliefs about the defender’s strategy and the number of observations the attacker will make.
3. Finally, the attacker makes \( \tau \) observations and selects the optimal target based on his posterior belief about the defender’s strategy.

We assume that the attacker and the defender have common prior beliefs over the set of mixed strategies that the defender may execute. In addition, we introduce a discount factor to model the cost of surveillance. We also assume that the defender does not know the exact times when the attacker will observe the strategy being executed, and therefore cannot strategically change the strategy during times when it could be observed. This is realistic if the defender is operating in a steady state, and does not know when or where surveillance operations could take place for planning a specific attack.

In the rest of the paper we apply a backwards induction approach to analyze SGSS. First we model how the attacker updates his belief and chooses the best target to attack. Then we formulate an optimization problem for the defender’s optimal strategy, given that the attacker will make a known number of observations. Finally, we discuss how the attacker can make a decision on how many observations to make.

Updating Attacker Beliefs

In an SGSS, the attacker updates his beliefs about the defender’s strategy given his prior and \( \tau \) observations, labeled \( O^0, \ldots, O^{\tau-1} \), where each observation corresponds to one of the defender’s pure strategies. The individual observations are drawn independently from the distribution representing the defender’s mixed strategy. We can imagine the belief update proceeding sequentially, with an updated belief calculated after each observation. The attacker begins with a prior belief over the defender’s mixed strategies,
Table 1: Notations used in this paper

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>Number of defender resources</td>
</tr>
<tr>
<td>( T )</td>
<td>Set of targets</td>
</tr>
<tr>
<td>( A )</td>
<td>Set of defender strategies</td>
</tr>
<tr>
<td>( \Phi )</td>
<td>All ways of allocating ( m ) resources on ( T )</td>
</tr>
<tr>
<td>( x )</td>
<td>Defender mixed strategy ( x_i )</td>
</tr>
<tr>
<td>( c )</td>
<td>Defender coverage ( c_j )</td>
</tr>
<tr>
<td>( a )</td>
<td>Attacker coverage ( a_j )</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Parameter of attacker’s prior belief</td>
</tr>
<tr>
<td>( \tau )</td>
<td>Number of observations</td>
</tr>
<tr>
<td>( f^0(x) )</td>
<td>PDF of attacker’s prior belief</td>
</tr>
<tr>
<td>( f^1(x</td>
<td>o) )</td>
</tr>
<tr>
<td>( \alpha^0 )</td>
<td>Attacker’s strategy when his observation is ( o )</td>
</tr>
<tr>
<td>( c^0 )</td>
<td>Attacker’s updated belief about ( t_j )’s coverage given ( o )</td>
</tr>
<tr>
<td>( Z )</td>
<td>Huge positive constant</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Attacker’s utility discount rate</td>
</tr>
</tbody>
</table>

represented by the probability density function \( f^0(x) \) which represents the probability that the defender’s mixed strategy is \( x \). We assume this prior is common knowledge. Given the first observation \( O^0 \), the attacker applies Bayes’ rule to calculate the posterior distribution \( f^1(x|O^0) \) over the defender’s mixed strategies \( x \). The posterior distribution \( f^1(x) \) is then used as the prior belief distribution for observation \( O^1 \). After making \( \tau \) observations, the attacker attacks the target with the highest expected value with respect to the final posterior distribution \( f^\tau(x|O^0,\ldots,O^{\tau-1}) \).

Example 1. We use the LAX airport as an example, based on the ARMOR application. The police at LAX place \( m \) checkpoints on the entrance roads to LAX following a mixed strategy computed using the ARMOR system (Assistant for Randomized Monitoring over Routes) (Pita et al. 2008). Attackers may engage in surveillance prior to an attack. In practice, the attackers will make only a limited number of observations of how the checkpoints are placed before they launch an attack. For example, they might observe placements for 20 days, and then launch an attack a week later after finalizing plans for the attack based on analysis of the security strategy. A single observation in this domain might involve the attacker driving around the different entrances to the airport to determine which ones are covered by checkpoints at any particular time, so each observation gives information about the full strategy of the defender.

For simplicity, we assume in this work that the attacker’s beliefs can be represented as a Dirichlet distribution, which is a conjugate prior for the multinomial distribution. Specifically, the support for the prior distribution \( f^0(x) \) is the simplex \( S = \{ x : \sum_{A_i \in \Phi} x_i = 1, x_i \geq 0, \forall A_i \in \Phi \} \), where \( \Phi \) is the enumeration of all possible ways of allocating \( m \) resources to cover the targets in \( T \). \(^3\) We can consider more general security settings in which there may exist scheduling constraints on assignment of resources, e.g., resources have restrictions on which sets of targets they can cover (Jain et al. 2010). In this case, it follows that \( A \subset \Phi \). If we assume that the attacker has no knowledge of the defender’s scheduling/resource constraints, the attacker will have priors and update beliefs on the set of pure strategies \( \Phi \).

The Dirichlet distribution for \( f^0(x) \) is of the form

\[
f^0(x) = \beta \prod_{A_i \in \Phi} (x_i)^{\alpha_i} \quad \text{where} \quad \alpha = (\alpha_i) \text{ is a parameter of the Dirichlet distribution and} \quad \alpha_i > 0.
\]

By solving the integral \( \int_S f^0(x)dx = 1 \), we have \( \beta = \frac{(\sum_{A_i \in \Phi} \alpha_i + |\Phi| - 1)!}{\prod_{A_i \in \Phi} \alpha_i} \).

The prior belief can then be represented as follows:

\[
f^0(x) = \frac{(\sum_{A_i \in \Phi} \alpha_i + |\Phi| - 1)!}{\prod_{A_i \in \Phi} \alpha_i} \prod_{A_i \in \Phi} (x_i)^{\alpha_i}
\]

The probability that the defender will choose pure strategy \( A_i \) given the attacker’s prior belief \( f^0(x) \) is

\[
f^0(x_i) = \int_S x_i f^0(x)dx = \frac{\alpha_i + 1}{\sum_{A_i \in \Phi} \alpha_i + |\Phi|}
\]

The marginal coverage of target \( t_j \) given prior belief \( f^0(x) \) is

\[
p^0(j) = \sum_{A_i \in \Phi} A_{ij} f^0(x_i) = \frac{\sum_{A_i \in \Phi} A_{ij} (\alpha_i + 1)}{\sum_{A_i \in \Phi} \alpha_i + |\Phi|}
\]

If \( \alpha_i = \alpha_k \) for every \( i, k \), \( f^0(x_i) = \frac{1}{|\Phi|} \) for any strategy \( A_i \in \Phi \). That is, (from the attacker’s perspective) the defender chooses each strategy with the same probability. The probability of strategy \( A_i \) will increase with the increase of \( \alpha_i \).

Next we discuss how the attacker updates his belief given his prior belief and the sequence of observations \( O = \{O^0,\ldots,O^{\tau-1}\} \) where \( O^k \in \Phi \). Let \( \alpha_i(O) \) (or \( \alpha_i \) for short) be the number of times each pure strategy \( A_i \) is executed, with \( \sum_{A_i \in \Phi} \alpha_i = \tau \). If the defender’s mixed strategy is \( x \), the probability that the attacker will observe \( o = (\alpha_i) \) is

\[
f(o|x) = \frac{\tau!}{\prod_{A_i \in \Phi} \alpha_i!} \prod_{A_i \in \Phi} (x_i)^{\alpha_i}.
\]

After the first observation \( O^0 = A_i \), the attacker’s posterior distribution \( f^1(x|O^0) \) can be computed by applying Bayes’ rule as follows:

\[
f^1(x|O^0) = \frac{f(o|x) f^0(x)}{\int_S f(o|x) f^0(x)dx} = \frac{\alpha_i + 1}{\sum_{A_i \in \Phi} \alpha_i + |\Phi|} f^0(x)
\]

\(^3\)We assume that the attacker has prior knowledge about the probability distribution \( f^0(e) \) over the defender’s pure strategies. It is also possible that the attacker has prior belief on targets \( T \)’s marginal coverage (say \( f^0(c) \)). In that case, we can convert \( f^0(e) \) to \( f^0(x) \) by solving a set of linear functions \( c_j = \sum_{A_i \in \Phi} x_i A_{ij}, \forall t_j \in T \).
tacker strategy in response to a defender strategy. Let launching his attack.

After applying Bayes’ rule for all $\tau$ observations, we can calculate the posterior distribution as:

$$f^*(x|\tau) = \frac{(\sum_{A_i \in \Phi} A_{ij}(\alpha_i + |\Phi| + \tau - 1)!}{\sum_{A_i \in \Phi} A_{ij}(\alpha_i + \tau)} \prod_{A_i \in \Phi} \prod_{x_i} (x_i)^{\alpha_i + o_i}$$

The marginal coverage of target $t_j$ given the posterior belief $f^*(x)$ is

$$p^*(j) = \sum_{A_i \in \Phi} A_{ij} f^*(x_i) = \sum_{A_i \in \Phi} A_{ij}(\alpha_i + o_i + 1)$$

After calculating these belief updates for all of the observations, the attacker chooses the best target to attack based on the final posterior belief $f^*(x|\tau)$. The defender’s real strategy $x$ can affect the probability of the attacker’s observations and therefore affect the attacker’s choice of target.

Computing the Defender’s Optimal Strategy

In this section we consider the problem of computing the defender’s optimal strategy $x$ given (1) the attacker’s prior belief $f^0(x)$ represented as a Dirichlet distribution with parameter $\alpha = (\alpha_i)$, and (2) the fact that the attacker will make a known and fixed number of observations ($\tau$) before launching his attack.

**Attacker’s Optimal Strategy**

We first discuss the problem of calculating the optimal attacker strategy in response to a defender strategy. Let $O_\tau$ be the space of possible observations when the attacker makes $\tau$ observations, represented as $O_\tau = \{o : o_i \in \{0, \ldots, \tau\}, \sum_{A_i \in A} o_i = \tau\}$. The space $O_\tau$ is finite and independent of the defender’s strategy $x$.

One important feature of SGSS is that the attacker’s decision about which target to attack is determined by his prior belief and his observation $o$. Therefore, we can compute offline the attacker’s optimal strategy $a^*$ for each observation $o$ by solving the following linear program (LP):

**P1:**

$$\max d^o$$

$$a^o_j \in (0, 1) \quad \forall t_j \in T$$

$$\sum_{t_j \in T} a^o_j = 1$$

$$d^o - c^o_j (R^d - P^d_j) - P^a_j \leq (1 - a^o_j)Z \quad \forall t_j \in T$$

$$c^o_j = \frac{\sum_{A_i \in \Phi} A_{ij}(\alpha_i + o_i + 1)}{\sum_{A_i \in \Phi} A_{ij}(\alpha_i + |\Phi| + \tau)}$$

$$0 \leq c^o_j (P^d_j - R^d_j) - R^a_j \leq (1 - a^o_j)Z \quad \forall t_j \in T$$

The formulation P1 is similar to the MILP formulations for security games presented in (Kiekintveld et al. 2009). Equation (1) is the objective function which maximizes the defender’s expected payoff from the attacker’s perspective. As in Strong Stackelberg equilibrium, we still assume that the attacker breaks ties in favor of the defender. Equations (2) and (3) force the attacker vector to assign a single target probability 1 for each observation $o$. Equation (4) defines the defender’s payoff from the defender’s perspective. Equation (5) defines the defender’s updated belief about the coverage of each target given the observation $o$, $a^*$ represents the attacker’s strategy when his observation is $o$. $Z$ is a huge positive constant. $c^o_j$ is the attacker’s updated belief about the coverage of target $t_j$ if his observation is $o$. $k^o$ is the attacker’s expected utility (from the attacker’s perspective) when his observation is $o$.

In the rest of this section, we provide three mathematical programming formulations for computing the defender’s optimal strategy $x^*$ when the number $\tau$ of observations is known. Throughout, we assume that $a^o$ is known for each potential observation $o$.

**Non-convex Optimization Formulation**

The formulation P2 provides a straightforward approach for computing the defender’s optimal strategy. Equation (7) is the objective function which maximizes the defender’s expected payoff $\sum_{o \in O_\tau} f(o|x) d^o$ where $d^o$ is the defender’s utility when the attacker’s observation is $o$. Equations (8) and (9) restrict the defender’s strategy space $x$. Equation (10) computes each target’s marginal coverage given the defender’s strategy $x$. Equation (11) defines the defender’s expected payoff $d^o$ when the attacker’s observation is $o$.

**P2:**

$$\max \sum_{o \in O_\tau} \prod_{A_i \in A} x_i^{a^o_i} d^o$$

$$x_i \in [0, 1] \quad \forall A_i \in A$$

$$\sum_{A_i \in A} x_i = 1$$

$$c^o_j = \sum_{A_i \in A} x_i A_{ij} \quad \forall t_j \in T$$

$$d^o - c^o_j (R^d - P^d_j) - P^a_j \leq (1 - a^o_j)Z \quad \forall t_j, o \in T \times O_\tau$$

**Convex Approximation**

The objective function (7) in formulation P2 is not convex, and no existing solver can guarantee finding the optimal solution. One approach in this case is to fall back to approximation. In this case, we can approximate the original problem by taking the log inside the summation for the objective function, using the following approximation:

$$\log(\prod_{A_i \in A} x_i^{a^o_i} d^o) \approx \sum_{A_i \in A} \log(x_i) + \sum_{A_i \in A} \log(d^o)$$

The issue can be resolved by adding a large value to each entry in the payoff matrix so $d^o$ will always be positive. Since the equation $\sum_{o \in O_\tau} \log(\prod_{A_i \in A} x_i^{a^o_i} d^o)$ is concave and non-decreasing, we can convert this Hessian approximation problem as follows:
P3:  
\[
\min_{o \in O} \sum_{t \in T} \left( - \log \prod_{A_i \in A} a_i^{o_i} - \sum_{A_i \in A} a_i \log(x_i) - \log d^o \right) \tag{12}
\]

(8) – (11) \tag{13}

We have conducted initial experiments to evaluate the above two formulations P2 and P3. Our experiments use 100 sample game instances with 1 defender resource, a varying numbers of targets, and randomly-generated payoffs. The payoffs are drawn uniformly from the range \([0, 10000000]\), and we then enforce the constraint that rewards are higher than penalties when the uncovered utility is drawn for each player.

Figure 3 compares the expected defender utilities for formulations P2 and P3. It shows that both formulations achieve almost the same defender utility with different number of observations.

The Optimal Number of Observations

This section discusses how the attacker decides the number of observations to make, considering the the cost of surveillance. We model surveillance cost by introducing a discount factor \(\lambda \in (0, 1)\) for the attacker. We can then formulate the attacker’s optimization problem with the discount factor as a bilevel optimization problem P4 by extending the formulation P2:

P4:  
\[
\max_{\tau} \lambda^\tau \sum_{o \in O} \prod_{A_i \in A} a_i^{o_i} \prod_{A_i \in A} (x_i)^{o_i} k^o \tag{14}
\]

\(\tau \in \mathbb{N}\) \tag{15}

\(x = \arg \max_{\tau} \sum_{o \in O} \prod_{A_i \in A} a_i^{o_i} \prod_{A_i \in A} (x_i)^{o_i} d^o\) \tag{16}

\(x_i \in [0, 1] \quad \forall A_i \in A\) \tag{17}

\[\sum_{A_i \in A} x_i = 1\] \tag{18}

\[c_j = \sum_{A_i \in A} x_i a_{ij} \quad \forall j \in T\] \tag{19}

\[d^o - c_j (R^o_j - P^o_j) - P^o_j \leq (1 - o^o_j) Z \quad \forall j, o \in T \times O\] \tag{20}

\[0 \leq k^o - c_j^o (P^o_j - R^o_j) - R^o_j \leq (1 - o^o_j) Z \quad \forall j, o \in T \times O\] \tag{21}

In formulation P4, Equation (14) is the objective function which maximizes the attacker’s expected payoff \(\lambda^\tau \sum_{o \in O} f(o|x) k^o\) when the attacker makes \(\tau\) observations, and the defender takes strategy \(x\). Equation (15) restricts the possible number of observations the attacker can make. Equations (16)-(21) maximize the defender’s expected utility when \(\tau\) is known. \(k^o\) is the attacker’s utility when 1) the attacker makes \(\tau\) observations and 2) the defender takes strategy \(x\).

Bilevel optimization problems are intrinsically hard, and P4 is even more difficult to solve since both the upper-level problem and the second-level problem are not convex. One approach is to try different values of \(\tau\) and solve the defender’s optimization problems using the methods described previously. Intuitively, due to the existence of discount factor \(\lambda\), the attacker’s utility will decrease as \(\tau\) increases for sufficiently large values of \(\tau\). Therefore, we may be able to use some form of intelligent search to find the optimal value of \(\tau\).

Conclusion

This paper explicitly models the attacker’s belief update and strategic surveillance decisions in security games, and presents efficient solution techniques to compute agents’ optimal strategies. Our primary contributions are as follows:

1. We model the security games with strategic surveillance
and formulate how the attacker updates his belief given limited observations. (2) We provide multiple formulations for computing the defender’s optimal strategies, including non-convex programming and convex approximation. (3) We provide an approximate approach for computing the attacker’s optimal surveillance length. (4) We present initial experimental results comparing the efficiency and runtime of our algorithms.

Our future work will focus on designing more efficient algorithms for computing the optimal strategy. Since solving bilevel optimization problems is very difficult, we will also look at some heuristic algorithm such as penalty function methods and trust-region methods. We also plan to conduct more extensive experiments to explore the implications of limited observation on both the strategies and outcomes in security games.

Acknowledgments
This research is supported by MURI grant W911NF-11-1-0332 and ONR grant N00014-08-1-0733.

References


Southers, E. 2011. LAX - terror target: the history, the reason, the countermeasure. Cambridge University Press.


