Security Games with Surveillance Cost and Optimal Timing of Attack Execution

Bo An[1], Matthew Brown[2], Yevgeniy Vorobeychik[3], Milind Tambe[2]
[1]The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
[2]University of Southern California, Los Angeles, CA, 90089, USA
[3]Sandia National Laboratories*, Livermore, CA 94550, USA
[1]boan@ict.ac.cn, [2]{mattheab,tambe}@usc.edu, [3]yvorobe@sandia.gov

ABSTRACT

Stackelberg games have been used in several deployed applications to allocate limited resources for critical infrastructure protection. These resource allocation strategies are randomized to prevent a strategic attacker from using surveillance to learn and exploit patterns in the allocation. Past work has typically assumed that the attacker has perfect knowledge of the defender's randomized strategy or can learn the defender's strategy after conducting a fixed period of surveillance. In consideration of surveillance cost, these assumptions are clearly simplistic since attackers may act with partial knowledge of the defender's strategies and may dynamically decide whether to attack or conduct more surveillance. In this paper, we propose a natural model of limited surveillance in which the attacker dynamically determine a place to stop surveillance in consideration of his updated belief based on observed actions and surveillance cost. We show an upper bound on the maximum number of observations the attacker can make and show that the attacker's optimal stopping problem can be formulated as a finite state space MDP. We give mathematical programs to compute optimal attacker and defender strategies. We compare our approaches with the best known previous solutions and experimental results show that the defender can achieve significant improvement in expected utility by taking the attacker's optimal stopping decision into account, validating the motivation of our work.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: [Multiagent Systems]

General Terms

Algorithm, Security

Keywords

Game Theory, Security, Optimization, Stackelberg Games

1. INTRODUCTION

Stackelberg security games have been used in several deployed applications for protecting critical infrastructure including LAX Airport, US Coast Guard, and the Federal Air Marshals Service [4, 7, 5, 14, 3, 12, 1]. A Stackelberg security game models a sequential interaction between a defender and an attacker [6]. The defender first commits to a randomized security policy, and the attacker uses surveillance to learn about the policy before attacking. A Stackelberg equilibrium of this game yields an optimal security policy for the defender, based on the assumption that the attacker will observe this strategy and respond optimally.

Terrorists conduct surveillance to select potential targets and gain strong situational awareness of targets' vulnerabilities and security operations [13]. Most existing work on security games, including deployed applications, assumes that the attacker is able to observe the defender's strategy perfectly or can learn the defender's strategy after conducting a fixed period of surveillance. These assumptions are a useful first-level approximation, but it is clearly simplistic. In reality, the attacker may have more limited observation capabilities since surveillance is costly and delays an attack. Attackers may also wish to reduce the number of observations due to the risk of being detected by security forces while conducting surveillance [13]. Therefore, it is essential to consider the attacker's dynamic decision making while conducting limited surveillance.

While there has been some related work that relaxes the perfect observation assumption in security games, the proposed approaches have some fundamental drawbacks. The COBRA algorithm [10] focuses on human perception of probability distributions by applying support theory [15]. RECON [18] considers imperfect observation 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 how the beliefs are formed. Both RECON and COBRA require hand-tuned parameters to model observations errors, which make them less applicable. Korzhyk *et al.* [8] also consider imperfect observation but only consider perfect observation and no observation. In practice, an attacker may have partial knowledge of the defender's strategy rather than the two extreme situations. Generally, Stackelberg equilibria and Nash equilibria in security games are different [19], and the defender's optimal strategy with limited surveillance may be different from both the Stackelberg and Nash equilibria. An *et. al* [2] propose a model wherein an attacker updates his belief based on a limited number of observed actions. But the model assumes that the defender can perfectly estimate the number of observations the attacker will make, which is unrealistic. 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 [9, 16].

In this paper, we propose a natural model of limited surveillance in which the attacker dynamically determines whether to make more observations or to attack his best target immediately. The attacker's optimal stopping decision after each observation takes into

*Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013),* Ito, Jonker, Gini, and Shehory (eds.), May 6–10, 2013, Saint Paul, Minnesota, USA.

account both his updated belief based on observed defender actions and surveillance cost. Such an optimal stopping model for limited surveillance does not assume the knowledge about the defender's strategy nor the number of observations the attacker will make.

We investigate the model both theoretically and experimentally. We make the following key contributions: (1) We introduce a model of security games with limited surveillance in which attacker dynamically decides when to attack. (2) We show that the attacker's optimal stopping problem can be formulated as a discrete state space MDP. (3) We show an upper bound on the maximum number of observations the attacker can make and thus the stopping problem is equivalent to a finite state MDP. (4) We give mathematical programs for computing optimal attacker and defender strategies. (5) Experimental results show that the defender can gain significantly higher utility by considering the optimal stopping decision.

2. STACKELBERG SECURITY GAMES

A Stackelberg security game [6] has two players, a defender who uses m identical resources to protect a set of targets $T = \{1, 2, \ldots, n\}$ ($m < n$), and an attacker who selects a single target to attack. The defender has N pure strategies $\mathcal{A}$, each a coverage vector representing which m targets are covered. Our model can handle more general security settings in which there may exist scheduling constraints on the assignment of resources. In that case, $\mathcal{A}$ represents feasible assignments. We write $A_i = 1$ if target i is covered in strategy $A \in \mathcal{A}$, and $A_i = 0$ otherwise. Each target i is covered by some pure strategies. The defender can choose a randomized strategy $\mathbf{x}$, with $x_A \geq 0$ the probability of playing a strategy A. A defender strategy can be represented more compactly using a marginal coverage vector $\mathbf{c}(\mathbf{x}) = \langle c_i(\mathbf{x})\rangle$ where $c_i(\mathbf{x}) = \sum_{A\in\mathcal{A}} x_A A_i$ is the probability that target i is covered by some defender resource [6]. The attacker's strategy is a vector $\mathbf{a} = \langle a_i\rangle$ where a_i is the probability of attacking target i. Since the attacker always has an optimal pure-strategy response, we restrict the attacker's strategies to pure strategies without loss of generality.

The payoffs for each player depend on which target is attacked and the probability that the target is covered. If the attacker attacks target i, there are two cases: If target i is covered, the defender receives a reward R_i^d and the attacker receives a penalty P_i^a. Otherwise, the payoffs for the defender and attacker are P_i^d and R_i^a, respectively. We assume that $R_i^d \geq P_i^d$ and $R_i^a \geq P_i^a$ in order to model that the defender would always prefer the attack to fail, while the attacker would prefer it to succeed. For a strategy profile $\langle \mathbf{c}, \mathbf{a}\rangle$, the expected utilities for both agents are given by:

$$U^d(\mathbf{c},\mathbf{a}) = \sum_{i\in T} a_i U^d(c_i, i), \text{ where } U^d(c_i, i) = c_i R_i^d + (1 - c_i) P_i^d$$

$$U^a(\mathbf{c},\mathbf{a}) = \sum_{i\in T} a_i U^a(c_i, i), \text{ where } U^a(c_i, i) = c_i P_i^a + (1 - c_i) R_i^a$$

In a Stackelberg game, the defender moves first, choosing $\mathbf{x}$, while the attacker observes $\mathbf{x}$ and plays an optimal response $\mathbf{a}$ to it. The standard solution concept is strong Stackelberg equilibrium (SSE) [17]. In an SSE, the defender chooses an optimal strategy $\mathbf{x}$, accounting for the attacker's best response $\mathbf{a}$, under the assumption that the attacker breaks ties in the defender's favor.

3. OPTIMAL STOPPING SECURITY GAMES

In this work we depart from the typical Stackelberg assumption that the attacker has full knowledge of the defender strategy $\mathbf{x}$, and also relax the assumption made by An et al. [2] that the defender knows how many observations the attacker will make. Instead, we model the attacker as a Bayesian decision maker who optimally solves the following sequential decision problem: for each sequence of observed defender moves, decide whether to attack now, or to make another observation of the defender, at some fixed cost. We refer to our model as OPTS (OPtimal sTopping Security games). Specifically, OPTS assumes the following sequence of moves:

1. The defender chooses a mixed strategy $\mathbf{x}$. We assume that when choosing a strategy, the defender has knowledge of the attacker's prior beliefs.
2. The attacker decides whether to make an observation or to attack immediately. After making one observation, the attacker updates his belief and decides whether to continue observing the defender based on his posterior belief about the defender's strategy, and so on. The game ends when the attacker attacks a target.

Consider the LAX airport example based on the ARMOR application [2, 11]. The police at LAX deploy checkpoints on the entrance roads to LAX according to a randomized strategy computed by ARMOR. Prior to an attack, attackers typically engage in surveillance [13] which can take the form of driving around the different airport entrances, but will ultimately launch an attack based on a finite number of observations of the checkpoint locations.[1] More salient to our work is that a rational attacker can, and will, dynamically choose whether to attack or to continue surveillance, depending on the particular sequence of observed checkpoint locations, as well as his prior belief and observation costs.

Formally, in an OPTS model the attacker has a prior belief about the defender's strategy, updates this belief upon observing actual defense realizations, and dynamically decides whether to stop surveillance after each observation based on this posterior belief. Suppose that the attacker makes a sequence of $\tau \geq 0$ observations, $\sigma = \{\sigma^1, \ldots, \sigma^\tau\}$, where each observation σ^i corresponds to the defender's pure strategy realization A, drawn i.i.d. from the defender's mixed strategy. Such a sequence of observations σ can be compactly represented by an observation vector $\mathbf{o} = \langle o_A\rangle$ in which o_A is the number of times pure strategy A was observed. Thus, a single observation vector $\mathbf{o}$ represents $\frac{\tau!}{\prod_{A\in\mathcal{A}} o_A!}$ different observation sequences. Let $\mathcal{O} = \cup_{\tau\in\mathbb{Z}_{\geq 0}} \mathcal{O}_\tau$ denote the set of all possible observation vectors, with $\mathcal{O}_\tau = \{\mathbf{o} : o_A \in \{0, \ldots, \tau\}, \sum_{A\in\mathcal{A}} o_A = \tau\}$ the set of all observation vectors with length (the number of observations) exactly τ. Additionally, we let $\mathbf{o} = \langle o_A = 0\rangle$ be an "empty" observation vector corresponding to the intial attacker state prior to surveillance activity.

The attacker starts the decision problem with a prior belief, which is also known to the defender. We assume that the attacker's belief is represented as a Dirichlet distribution with support $\mathcal{S} = \{\mathbf{x} : \sum_{A\in\mathcal{A}} x_A = 1, x_A \geq 0, \forall A \in \mathcal{A}\}$. A Dirichlet distribution $f(\mathbf{x})$ is characterized by a parameter vector $\alpha = \langle \alpha_A\rangle$ with $\alpha_A > -1$ for all $A \in \mathcal{A}$. It assigns probability $\beta \prod_{A\in\mathcal{A}} (x_A)^{\alpha_A}$ to the defender's mixed strategy $\mathbf{x}$, where $\beta = \frac{\Gamma(\sum_{A\in\mathcal{A}} \alpha_A + |\mathcal{A}|)}{\prod_{A\in\mathcal{A}} \Gamma(\alpha_A + 1)}$ is a normalization constant expressed in terms of the gamma function Γ. The prior belief can be represented as follows:

$$f(\mathbf{x}) = \frac{\Gamma(\sum_{A\in\mathcal{A}} \alpha_A + |\mathcal{A}|)}{\prod_{A\in\mathcal{A}} \Gamma(\alpha_A + 1)} \prod_{A\in\mathcal{A}} (x_A)^{\alpha_A}.$$

If the defender's *actual* mixed strategy is $\mathbf{x}$ and the attacker makes τ observations, the probability that the attacker observes $\mathbf{o} \in \mathcal{O}_\tau$ is $f(\mathbf{o}|\mathbf{x}) = \frac{\tau!}{\prod_{A\in\mathcal{A}} o_A!} \prod_{A\in\mathcal{A}} (x_A)^{o_A}$. By applying

[1] An alternative model could be developed where the attacker picks a subset of targets to observe, and will therefore only partially observe the strategy realization of the defender in each observation. We leave this model for future work.

Bayes' rule given the observation vector $\mathbf{o}$, we can calculate the posterior distribution as:

$$f(\mathbf{x}|\mathbf{o}) = \frac{\Gamma(\sum_{A\in\mathcal{A}}\alpha_A + |\mathcal{A}| + \tau)}{\prod_{A\in\mathcal{A}}\Gamma(\alpha_A + o_A + 1)} \prod_{A\in\mathcal{A}} (x_A)^{\alpha_A + o_A}.$$

Having observed $\mathbf{o}$, the attacker believes that the probability with which the defender chooses a pure strategy A is

$$\Pr(A|\mathbf{o}) = \int_{\mathcal{S}} x_A f(\mathbf{x}|\mathbf{o})\mathrm{d}\mathbf{x} = \frac{\alpha_A + o_A + 1}{\sum_{A'\in\mathcal{A}}\alpha_{A'} + |\mathcal{A}| + \tau}.$$

Finally, the marginal coverage of target i according to the posterior belief $f(\mathbf{x}|\mathbf{o})$ is

$$c_i^{\mathbf{o}} = \sum_{A\in\mathcal{A}} A_i \cdot \Pr(A|\mathbf{o}) = \frac{\sum_{A\in\mathcal{A}} A_i(\alpha_A + o_A + 1)}{\sum_{A\in\mathcal{A}}\alpha_A + |\mathcal{A}| + \tau} > 0.$$

In the OPTS model, for any observation vector $\mathbf{o}$, the attacker can either attack a target immediately, choosing the target that maximizes his expected utility with respect to the posterior belief $f(\mathbf{x}|\mathbf{o})$, or continue surveillance. If the attacker chooses to attack a target i, he obtains the expected utility of $U^a(c_i^{\mathbf{o}}, i)$ as his immediate reward, while the defender receives the expected utility of $U^d(c_i, i)$ (where c_i is the actual coverage of i), and the game ends. On the other hand, if he chooses to make another observation, he has to "pay" a cost $\lambda > 0$, which represents the opportunity cost of delaying an attack, such as increasing the probability that the attacker is captured before an attack can be carried out.

4. ATTACKER'S DECISION PROBLEM

Since the defender decides her strategy before the surveillance phase and the attacker will decide whether to continue to observe after each observation, the attacker's optimal stopping problem can be formulated as a Markov Decision Process (MDP) in which states are the observation vectors $\mathbf{o}$. The attacker's optimal stopping problem can be solved without knowing the defender's true strategy $\mathbf{x}$. Therefore, we can first compute the attacker's optimal policy, and then use it to compute the optimal mixed strategy commitment for the defender.

Observe that the MDP is in fact a directed acyclic graph (DAG) if we connect states with only non-zero transition probabilities, since there is an edge from state $\mathbf{o}$ to state $\mathbf{o}'$ if and only if $\mathbf{o}' = \mathbf{o} \cup \{A\}$ for an $A \in \mathcal{A}$. Therefore, an observation vector with observation length τ is connected to only $|\mathcal{A}|$ observation vectors with length $\tau + 1$. (The initial state in this DAG represents the state before any observations have been made.)

If the attacker attacks his best target $\psi(\mathbf{o})$ at state with observation vector $\mathbf{o}$, he will gain an immediate utility[2]

$$W(\mathbf{o}) = U^a(\mathbf{o}) - \lambda \cdot \Delta(\mathbf{o}),$$

where $U^a(\mathbf{o}) = c^{\mathbf{o}}_{\psi(\mathbf{o})}(P^a_{\psi(\mathbf{o})} - R^a_{\psi(\mathbf{o})}) + R^a_{\psi(\mathbf{o})}$ is the attacker's utility without considering observation cost, $c^{\mathbf{o}}_{\psi(\mathbf{o})}$ is the marginal coverage of target $\psi(\mathbf{o})$ according to the posterior belief $f(\mathbf{x}|\mathbf{o})$, and $\Delta(\mathbf{o}) = \sum_{A\in\mathcal{A}} o_A$ is the length of observation vector $\mathbf{o}$.

The attacker can also make another $\tau' > 0$ observations after he observes $\mathbf{o}$. If the attacker's expected utility from making more observations is lower than $W(\mathbf{o})$, he will just attack his best target $\psi(\mathbf{o})$. Formally, we define a value function $V(\mathbf{o})$ for each observation vector $\mathbf{o}$, which represents the attacker's expected utility when his observation vector is $\mathbf{o}$ and he follows the optimal policy afterwards. At each state, the attacker can either attack the best target $\psi(\mathbf{o})$ and gain a utility $W(\mathbf{o})$ or make another observation, reaching state $\mathbf{o}' = \mathbf{o} \cup \{A\}$ with probability $\Pr(A|\mathbf{o})$. The optimal value function $V(\mathbf{o})$ thus satisfies the following dynamic programming recursion:

$$V(\mathbf{o}) = \max\big\{W(\mathbf{o}), \sum_{A\in\mathcal{A}} \Pr(A|\mathbf{o}) V(\mathbf{o} \cup \{A\})\big\}.$$

Without loss of generality, we assume that the attacker always chooses to attack when he is indifferent between attacking and making another observation.

The countable-state MDP is still a challenge due to its infinite horizon. However, we now present a series of results that demonstrate that, in fact, it suffices to consider observations of bounded length, which implies that we need only consider an MDP with a finite state space and a finite horizon. The intuition behind this result is that after many observations have been made, new observations do not change the posterior belief very much, so the value of new information is low, whereas the cost of making additional observations remains fixed for all time. Therefore, eventually the value of making additional observations will fall, and permanently remain, below the cost of making them, and the attacker will attack once that point is reached. What we proceed to show is that there is a uniform bound on the number of observations made by the attacker beyond which he will always attack.

First, we bound the most that the attacker can gain by taking a single observation and then attacking, rather than attacking immediately. We call this quantity

$$\mathbf{MV}(\mathbf{o}) = \max_{A\in\mathcal{A}} U^a(\mathbf{o} \cup \{A\}) - U^a(\mathbf{o}).$$

LEMMA 1. *For any* $\epsilon > 0$ *and for all* $\mathbf{o}$ *with* $\Delta(\mathbf{o}) = \tau > \frac{M}{\epsilon} - \sum_{A\in\mathcal{A}}\alpha_A - |\mathcal{A}| - 1$, $\mathbf{MV}(\mathbf{o}) < \epsilon$.

PROOF. Let $\tau = \Delta(\mathbf{o})$ and $\max_{j\in T}(R^a_j - P^a_j) = M \geq 0$. For any A, let $\psi_A = \psi(\mathbf{o} \cup \{A\})$ be the optimal target to attack after observing $\mathbf{o} \cup \{A\}$. Define $C_i = \sum_{A\in\mathcal{A}} A_i(\alpha_A + o_A + 1)$ and $\bar{C} = \sum_{A\in\mathcal{A}}\alpha_A + |\mathcal{A}| + \tau$, and note that $C_i/\bar{C} \leq 1$.

Without loss of generality, let $k = \psi(\mathbf{o})$, which implies that $U^a(\mathbf{o}) = U^a(c^{\mathbf{o}}_k, k) \geq U^a(c^{\mathbf{o}}_j, j)$ for all targets $j \in T$, i.e., $U^a(\mathbf{o}) = c^{\mathbf{o}}_k(P^a_k - R^a_k) + R^a_k = \frac{\sum_{A\in\mathcal{A}} A_k(\alpha_A + o_A + 1)}{\sum_{A\in\mathcal{A}}\alpha_A + |\mathcal{A}| + \Delta(\mathbf{o})}(P^a_k - R^a_k) + R^a_k = \frac{C_k}{\bar{C}}(P^a_k - R^a_k) + R^a_k \geq \frac{C_j}{\bar{C}}(P^a_j - R^a_j) + R^a_j$ for all j.

$$\begin{aligned}
\mathbf{MV}(\mathbf{o}) &= \max_{A\in\mathcal{A}}\Big(U^a(\mathbf{o} \cup \{A\}) - U^a(\mathbf{o})\Big) \\
&\leq \max_{A\in\mathcal{A}}\left(\frac{C_{\psi_A}}{\bar{C}+1}(P^a_{\psi_A} - R^a_{\psi_A}) + R^a_{\psi_A} - U^a(\mathbf{o})\right)^{3} \\
&\leq \max_{A\in\mathcal{A}}\Bigg(\frac{C_{\psi_A}}{\bar{C}+1}(P^a_{\psi_A} - R^a_{\psi_A}) + R^a_{\psi_A} \\
&\qquad - \Big(\frac{C_k}{\bar{C}}(P^a_k - R^a_k) + R^a_k\Big)\Bigg) \\
&\leq \max_{A\in\mathcal{A}}\Bigg(\frac{C_{\psi_A}}{\bar{C}+1}(P^a_{\psi_A} - R^a_{\psi_A}) + R^a_{\psi_A} \\
&\qquad - \Big(\frac{C_{\psi_A}}{\bar{C}}(P^a_{\psi_A} - R^a_{\psi_A}) + R^a_{\psi_A}\Big)\Bigg) \\
&\leq M \max_{A\in\mathcal{A}} C_{\psi_A}\Big(\frac{1}{\bar{C}} - \frac{1}{\bar{C}+1}\Big) \\
&\leq \frac{M}{\sum_{A\in\mathcal{A}}\alpha_A + |\mathcal{A}| + \tau + 1}
\end{aligned}$$

[2] Note that the expected utility is from the attacker's perspective and is based on his posterior belief. The *real* attacker utility depends on the defender's strategy which is unknown to the attacker.

[3] If $A_{\psi_A} = 1$, $U^a(\mathbf{o} \cup \{A\}) = \frac{C_{\psi_A}+1}{\bar{C}+1}(P^a_{\psi_A} - R^a_{\psi_A}) + R^a_{\psi_A} \leq \frac{C_{\psi_A}}{\bar{C}+1}(P^a_{\psi_A} - R^a_{\psi_A}) + R^a_{\psi_A}$. If $A_{\psi_A} = 0$, $U^a(\mathbf{o} \cup \{A\}) = \frac{C_{\psi_A}}{\bar{C}+1}(P^a_{\psi_A} - R^a_{\psi_A}) + R^a_{\psi_A}$.

Therefore, for any $\epsilon > 0$, letting $\tau > \frac{M}{\epsilon} - \sum_{A \in \mathcal{A}} \alpha_A - |\mathcal{A}| - 1$ implies that $\mathbf{MV}(\mathbf{o}) < \epsilon$ for any $\mathbf{o}$ with $\Delta(\mathbf{o}) = \tau$. □

The next lemma uses the bound we obtained on $\mathbf{MV}(\mathbf{o})$ for any $\mathbf{o}$ to show that when the number of observations is sufficiently large, the attacker will always attack immediately. The key is that this bound is only in terms of the length of an observation vector, τ, and is therefore uniform across all observation vectors with length at least τ.

LEMMA 2. *Suppose that* $\Delta(\mathbf{o}) > \frac{M}{\lambda} - \sum_{A \in \mathcal{A}} \alpha_A - |\mathcal{A}| - 1$. *Then* $V(\mathbf{o}) = W(\mathbf{o})$*, i.e., the attacker attacks immediately.*

Collecting Lemmas 1, and 2, we have proved the following theorem.

THEOREM 3. *The infinite horizon MDP is equivalent to an MDP with a finite state space.*

The power of this theorem is that we can now solve the bounded observation space MDP using backward induction. The problem that arises, however, is that the state space, though finite, is exponentially large in the upper bound on the number of observations. We consider the associated algorithmic questions in the following section.

5. COMPUTING AN OPTIMAL ATTACKER POLICY

Given the optimal value $V(\mathbf{o})$ for each state $\mathbf{o}$, we can decide the optimal policy (i.e., stopping rule) of the attacker as follows: with observation vector $\mathbf{o}$, the attacker will make at least another observation if and only if $W(\mathbf{o}) < V(\mathbf{o})$. The form of the optimal attacker policy that will be useful as an input to the defender's problem of computing the best commitment strategy is that of an *observation graph* $\mathcal{O}^*$, which is comprised of a set of observation vectors at which the attacker attacks. In constructing the observation graph, we must be careful not to include any observation vectors $\mathbf{o}$ that cannot be reached in the sense that the attacker already attacks at all other, shorter, observation vectors which must precede $\mathbf{o}$.

DEFINITION 4. *An observation vector* $\mathbf{o}$ *is reachable if and only if there exists a sequence of observation vectors* $\{\mathbf{o}^1, \ldots, \mathbf{o}^m\}$ *such that 1)* $\mathbf{o}^1 = \langle o^1_A = 0 \rangle$ *and* $\mathbf{o}^m = \mathbf{o}$*; 2) for each* $1 < i \le m$, $\mathbf{o}^i = \mathbf{o}^{i-1} \cup \{A\}$ *where* $A \in \mathcal{A}$*; and 3)* $V(\mathbf{o}^i) > W(\mathbf{o}^i)$ *for each* $1 \le i < m$*. Let the set of reachable observation vectors be* $\mathcal{O}^*$*.*

To construct $\mathcal{O}^*$, we initially set $\mathcal{O}^* = \{\mathbf{o} = \langle o_A = 0 \rangle\}$. Then, for each state $\mathbf{o} \in \mathcal{O}^*$ such that $W(\mathbf{o}) < \sum_{A \in \mathcal{A}} \Pr(A|\mathbf{o}) V(\mathbf{o} \cup \{A\})$ (i.e., $W(\mathbf{o}) < V(\mathbf{o})$), add the state $\mathbf{o}' = \mathbf{o} \cup \{A\}$ to $\mathcal{O}^*$ for each $A \in \mathcal{A}$. This process continues until no states can be added to $\mathcal{O}^*$. The height of an observation graph $\mathcal{O}^*$ is the maximum length of all the observation vectors in $\mathcal{O}^*$.

5.1 Backward Induction

Computing an optimal solution to the attacker's MDP amounts to computing the value function for all $\mathbf{o}$. Since our MDP has a finite horizon, it can be solved using backward induction, starting at all observation vectors with length $\tau_{max} = \lfloor \frac{M}{\lambda} - \sum_{A \in \mathcal{A}} \alpha_A - |\mathcal{A}| - 1 \rfloor + 1$ computed in Lemma 2, and working backwards towards the initial state. Then, for any observation vector $\mathbf{o}$ with $\Delta(\mathbf{o}) > \tau_{max}$, we set $V(\mathbf{o}) = W(\mathbf{o})$.

5.2 Backward Induction with Forward Search

The bound τ_{max} used in the naive backward induction approach above may not be tight in practice. If in fact the attacker always attacks even for $\tau < \tau_{max}$, using τ_{max} as the upper bound will result in an exponentially larger MDP that we must solve. Here we present an incremental algorithm which gradually considers larger observation vectors until the optimal policy is found. The challenge in constructing a forward search algorithm is that in order to compute the value of a given observation vector, we need to know the values of all observation vectors that can possibly follow it. We resolve this challenge by constructing an upper and lower bound on the entire value function, and using the convergence of these functions to each other to check when an optimal solution has been reached.

Suppose that we start backward induction from observation vectors with length $\tau^b \le \tau_{max}$. Two issues arise: how do we set the value $V(\mathbf{o})$ for each observation $\mathbf{o} \in \mathcal{O}_{\tau^b}$ and how it will affect the values of each $\mathbf{o}' \in \mathcal{O}_\tau$ with $\tau < \tau^b$. First, we bound the optimal value that the attacker can receive in any state $\mathbf{o}$.

LEMMA 5. $V(\mathbf{o}) \le R^a_{max} - \lambda \cdot \Delta(\mathbf{o})$ *where* $R^a_{max} = \max_{i \in T} R^a_i$ *is the attacker's maximum reward.*

This lemma implies that for $\mathbf{o} \in \mathcal{O}_{\tau^b}$, $W(\mathbf{o}) \le V(\mathbf{o}) \le R^a_{max} - \lambda \cdot \Delta(\mathbf{o})$. In addition, since $c^{\mathbf{o}}_i > 0$ for any observation vector $\mathbf{o}$, it follows that $U^a(\mathbf{o}) < R^a_{\max}$ and thus $W(\mathbf{o}) < R^a_{\max} - \lambda \cdot \Delta(\mathbf{o})$.

Let the optimal value of observation vector $\mathbf{o}$ be $V^{\tau^b}_{min}(\mathbf{o})$ when we set $V(\mathbf{o}) = W(\mathbf{o})$ for each $\mathbf{o} \in \mathcal{O}_{\tau^b}$. We compute the optimal value of $\mathbf{o} \in \mathcal{O}_{\le \tau^b}$ by applying backward induction as follows:

$$V^{\tau^b}_{min}(\mathbf{o}) = \begin{cases} W(\mathbf{o}) & \text{if } \mathbf{o} \in \mathcal{O}_{\tau^b} \\ \max\{W(\mathbf{o}), \sum_{A \in \mathcal{A}} \Pr(A|\mathbf{o}) V^{\tau^b}_{min}(\mathbf{o} \cup \{A\})\} & \text{if } \mathbf{o} \in \mathcal{O}_{<\tau^b} \end{cases}$$

Similarly, define $V^{\tau^b}_{max}(\mathbf{o})$ as the optimal value function when we set $V(\mathbf{o}) = R^a_{max} - \lambda \cdot \Delta(\mathbf{o})$ for each $\mathbf{o} \in \mathcal{O}_{\tau^b}$. The optimal value $V^{\tau^b}_{max}(\mathbf{o})$ of observation vector $\mathbf{o} \in \mathcal{O}_{\le \tau^b}$ in this case can be computed recursively by:

$$V^{\tau^b}_{max}(\mathbf{o}) = \begin{cases} R^a_{max} - \lambda \cdot \Delta(\mathbf{o}) & \text{if } \mathbf{o} \in \mathcal{O}_{\tau^b} \\ \max\{W(\mathbf{o}), \sum_{A \in \mathcal{A}} \Pr(A|\mathbf{o}) V^{\tau^b}_{max}(\mathbf{o} \cup \{A\})\} & \text{if } \mathbf{o} \in \mathcal{O}_{<\tau^b} \end{cases}$$

It is easy to see that $V^*(\mathbf{o}) = V^{\tau_{max}}_{min}(\mathbf{o})$ for $\mathbf{o} \in \mathcal{O}_{\le \tau^b}$ since the optimal policy can be computed by starting the backward induction from observation vectors with length τ_{max}. The following proposition shows that for any $\mathbf{o} \in \mathcal{O}_{\le \tau^b}$, $V^{\tau^b}_{min}(\mathbf{o})$ and $V^{\tau^b}_{max}(\mathbf{o})$ are in fact lower and upper bounds on $V^*(\mathbf{o})$, respectively.

PROPOSITION 6. $V^{\tau^b}_{min}(\mathbf{o}) \le V^*(\mathbf{o}) \le V^{\tau^b}_{max}(\mathbf{o})$ *for each observation vector* $\mathbf{o}$ *with length* $\Delta(\mathbf{o}) \le \tau^b$.

Next, we show that as we increase τ_b, the above bounds become tighter.

PROPOSITION 7. *For any* $\tau^b < \tau \le \tau_{max}$*, it follows that* $V^{\tau^b}_{min}(\mathbf{o}) \le V^{\tau}_{min}(\mathbf{o}) \le V^*(\mathbf{o}) \le V^{\tau}_{max}(\mathbf{o}) \le V^{\tau^b}_{max}(\mathbf{o})$ *for any* $\mathbf{o} \in \mathcal{O}_{\le \tau^b}$.

Given the values $V^{\tau^b}_{min}(\mathbf{o})$ and $V^{\tau^b}_{max}(\mathbf{o})$ for all observation vectors $\mathbf{o} \in \mathcal{O}_{\le \tau^b}$, we can form observation graph $\mathcal{O}^*_{min}(\tau^b)$ and $\mathcal{O}^*_{max}(\tau^b)$ for $\mathcal{O}_{\le \tau^b}$, respectively. The next lemma presents an intuitive fact about the relationship between these.

LEMMA 8. $\mathcal{O}^*_{min}(\tau^b) \subseteq \mathcal{O}^*_{max}(\tau^b)$.

The final two results in this section then establish that it suffices to check whether $V^{\tau^b}_{min}(\mathbf{o}) = V^{\tau^b}_{max}(\mathbf{o})$ *only at the initial state* $\mathbf{o} = \langle o_A = 0 \rangle$, rather than for the entire observation graph, providing us with the final building block for the *backward induction with forward search (BI-FS)* algorithm.

LEMMA 9. *If $\mathcal{O}^*_{min}(\tau^b) \subset \mathcal{O}^*_{max}(\tau^b)$, $V^{\tau^b}_{min}(\mathbf{o}) < V^{\tau^b}_{max}(\mathbf{o})$ for the initial state $\mathbf{o} = \langle o_A = 0\rangle$.*

PROOF. We first define sub-observation graphs. $\mathcal{O}^*_{min}(\tau^b)$'s sub-observation graph $\mathcal{O}^*_{min}(\tau^b, \mathbf{o})$ with initial state $\mathbf{o}$ can be constructed in the same way as the construction of $\mathcal{O}^*_{min}(\tau^b)$ except that the construction starts from state $\mathbf{o}$.

We now prove the result by induction on the height of the sub-observation graph $\mathcal{O}^*_{min}(\tau^b, \mathbf{o})$, which is the maximum difference of lengths of observation vectors $\mathcal{O}^*_{min}(\tau^b, \mathbf{o})$. Assume that $\mathcal{O}^*_{min}(\tau^b, \mathbf{o})$ includes only one state $\mathbf{o}$ and $\mathcal{O}^*_{min}(\tau^b, \mathbf{o}) \subset \mathcal{O}^*_{max}(\tau^b, \mathbf{o})$. The value of the initial state of $\mathcal{O}^*_{min}(\tau^b, \mathbf{o})$ is $W(\mathbf{o})$, which is smaller than $\mathcal{O}^*_{max}(\tau^b, \mathbf{o})$'s initial state value $V^{\tau^b}_{max}(\mathbf{o})$ since the attacker decides to make more observations. Assume the result is true for any sub-observation graph $\mathcal{O}^*_{min}(\tau^b, \mathbf{o})$ with height $h > 1$. Consider a sub-observation graph $\mathcal{O}^*_{min}(\tau^b, \mathbf{o})$ with height $h + 1$. Since $\mathcal{O}^*_{min}(\tau^b, \mathbf{o}) \subset \mathcal{O}^*_{max}(\tau^b, \mathbf{o})$, it follows that $\mathcal{O}^*_{min}(\tau^b, \mathbf{o} \cup \{A\}) \subset \mathcal{O}^*_{max}(\tau^b, \mathbf{o} \cup \{A\})$ for some $A \in \mathcal{A}$. It then follows that

$$\begin{aligned} V^{\tau^b}_{min}(\mathbf{o}) &= \sum_{A \in \mathcal{A}} \Pr(A|\mathbf{o}) V^{\tau^b}_{min}(\mathbf{o} \cup \{A\}) \\ &< \sum_{A \in \mathcal{A}} \Pr(A|\mathbf{o}) V^{\tau^b}_{max}(\mathbf{o} \cup \{A\}) = V^{\tau^b}_{max}(\mathbf{o}), \end{aligned}$$

which completes the proof. □

We are also able to check whether they are the same as the observation graph $\mathcal{O}^*$.

PROPOSITION 10. *If $V^{\tau^b}_{min}(\mathbf{o}) = V^{\tau^b}_{max}(\mathbf{o})$ for the initial state $\mathbf{o} = \langle o_A = 0\rangle$, the approximate observation graph $\mathcal{O}^*_{min}(\tau^b)$ is the same as the observation graph $\mathcal{O}^*$.*

PROOF. Given Lemma 8 and Lemma 9, it follows that $\mathcal{O}^*_{min}(\tau^b) = \mathcal{O}^*_{max}(\tau^b)$ since otherwise, $V^{\tau^b}_{min}(\mathbf{o}) < V^{\tau^b}_{max}(\mathbf{o})$ for the initial state $\mathbf{o} = \langle o_A = 0\rangle$.

Claim 1: $V^{\tau^b}_{min}(\mathbf{o}) = V^{\tau^b}_{max}(\mathbf{o})$ for each $\mathbf{o} \in \mathcal{O}^*_{min}(\tau^b)$.

Proof of Claim 1: We show this by contradiction. By Propositions 6, $V^{\tau^b}_{min}(\mathbf{o}) \leq V^{\tau^b}_{max}(\mathbf{o})$ for each $\mathbf{o} \in \mathcal{O}^*_{min}(\tau^b)$. Suppose that $V^{\tau^b}_{min}(\mathbf{o}) < V^{\tau^b}_{max}(\mathbf{o})$ for one state $\mathbf{o} \in \mathcal{O}^*_{min}(\tau^b) \cap \mathcal{O}_\tau$. For one state $\mathbf{o}' \in \mathcal{O}^*_{min}(\tau^b) \cap \mathcal{O}_{\tau-1}$ such that $\mathbf{o} = \mathbf{o}' \cup \{A\}$ for an $A \in \mathcal{A}$, it follows that $V^{\tau^b}_{min}(\mathbf{o}') < V^{\tau^b}_{max}(\mathbf{o}')$ given the backward induction definition. Continuing this process, we can get that $V^{\tau^b}_{min}(\mathbf{o}) < V^{\tau^b}_{max}(\mathbf{o})$ for the initial state $\mathbf{o} = \langle o_A = 0\rangle$, a contradiction. □

Claim 2: $\max_{\mathbf{o} \in \mathcal{O}^*_{min}(\tau^b)} \Delta(\mathbf{o}) < \tau^b$.

Proof of Claim 2: We show this by contradiction. Assume that an observation vector $\mathbf{o} \in \mathcal{O}_{\tau^b}$ is contained in both $\mathcal{O}^*_{min}(\tau^b)$ and $\mathcal{O}^*_{max}(\tau^b)$. By definition, we have $V^{\tau^b}_{min}(\mathbf{o}) = W(\mathbf{o})$ and $V^{\tau^b}_{max}(\mathbf{o}) = R^a_{max} - \lambda \cdot \Delta(\mathbf{o})$. Given *Claim 1*, it then follows that $W(\mathbf{o}) = R^a_{max} - \lambda \cdot \Delta(\mathbf{o})$, which contradicts to the fact that $W(\mathbf{o}) < R^a_{max} - \lambda \cdot \Delta(\mathbf{o})$. □

Since $V^*(\mathbf{o}) \leq V^{\tau^b}_{max}(\mathbf{o})$ for each observation vector $\mathbf{o}$ with length $\Delta(\mathbf{o}) \leq \tau^b$, the observation graph $\mathcal{O}^*$'s observation vectors with length no longer than τ^b should be a subset of $\mathcal{O}^*_{max}(\tau^b)$. By the fact that $\max_{\mathbf{o} \in \mathcal{O}^*_{max}(\tau^b)} \Delta(\mathbf{o}) < \tau^b$, this implies that $\mathcal{O}^* \subseteq \mathcal{O}^*_{max}(\tau^b)$. Similarly, we can show that $\mathcal{O}^*_{min}(\tau^b) \subseteq \mathcal{O}^*$. In consideration of the fact that $\mathcal{O}^*_{min}(\tau^b) = \mathcal{O}^*_{max}(\tau^b)$, it is easy to see that $\mathcal{O}^*_{min}(\tau^b) = \mathcal{O}^*_{max}(\tau^b) = \mathcal{O}^*$. □

Based on Proposition 10, we propose a search heuristic (Algorithm 1) to iteratively increase τ^b to find out the observation graph $\mathcal{O}^*$. The algorithm starts with a small τ^b and checks for convergence. If not, it increases τ^b until reaching the upper bound τ_{max}.

Algorithm 1: Backward Induction with Forward Search

```
$\tau^b \leftarrow 1$;
while $\tau^b < \tau_{max}$ do
    if $V^{\tau^b}_{min}(\mathbf{o}) = V^{\tau^b}_{max}(\mathbf{o})$ for the initial state $\mathbf{o} = \langle o_A = 0\rangle$ then
        return $\mathcal{O}^*_{min}(\tau^b)$
    end
    else $\tau^b \leftarrow 2\tau^b$ ;
end
return $\mathcal{O}^*_{min}(\tau_{max})$;
```

5.3 Approximation Approach

In the worst case, the forward search approach still needs to start backward induction from very long observation vectors, since the number of observation vectors (double) exponentially increases with observation length and the number of resources and targets. Here we propose a heuristic approach called *approximate forward search (A-BI-FS)* (Algorithm 2). The key difference of this heuristic from Algorithm 1 is that we only keep track of convergence of $V^{\tau^b}_{min}(\mathbf{o})$, which no longer guarantees optimality.

Algorithm 2: Approximate Forward Search

```
$\tau^b \leftarrow 1, \tau \leftarrow \lceil \beta\tau^b \rceil$ where $\beta > 1$;
while $\tau < \tau_{max}$ do
    Compute $V^{\tau^b}_{min}(\mathbf{o})$ and $V^{\tau}_{min}(\mathbf{o})$ for all $\mathbf{o} \in \mathcal{O}_{\leq\tau^b}$;
    if $\max_{\mathbf{o} \in \mathcal{O}_{\leq\tau^b}} |V^{\tau^b}_{min}(\mathbf{o}) - V^{\tau}_{min}(\mathbf{o})| < \epsilon$ then
        return $\mathcal{O}^*_{min}(\tau^b)$
    end
    else $\tau^b \leftarrow \tau, \tau \leftarrow \lceil \beta\tau \rceil$;
end
return $\mathcal{O}^*_{min}(\tau_{max})$;
```

6. OPTIMAL DEFENSE STRATEGY

After solving the attacker's optimal stopping problem, we obtain an observation graph with states $\mathcal{O}^*$. Let the leaves of graph be $\mathcal{O}^{*l}$ which represent the set of observation vectors for each of which the attacker will choose to attack its best target.

We now introduce an exact (but nonconvex) mathematical program for computing the defender's optimal strategy $\mathbf{x}$, assuming that $\psi(\mathbf{o})$ are pre-computed for all $\mathbf{o} \in \mathcal{O}^{*l}$. This is similar to the MILP formulation for security games presented in [2] except that in our case observation vectors in $\mathcal{O}^{*l}$ may vary in length.

DF-OPT:

$$\max \quad \sum_{\mathbf{o} \in \mathcal{O}^{*l}} \frac{\Delta(\mathbf{o})!}{\prod_{A \in \mathcal{A}} o_A!} \prod_{A \in \mathcal{A}} (x_A)^{o_A} d^{\mathbf{o}} \qquad (1)$$

$$\textbf{s.t.} \quad x_A \in [0,1] \qquad \forall A \in \mathcal{A} \quad (2)$$

$$\sum_{A \in \mathcal{A}} x_A = 1 \qquad (3)$$

$$c_i = \sum_{A \in \mathcal{A}} x_A A_i \qquad \forall i \in T \quad (4)$$

$$d^{\mathbf{o}} = c_{\psi(\mathbf{o})} R^d_{\psi(\mathbf{o})} + (1 - c_{\psi(\mathbf{o})}) P^d_{\psi(\mathbf{o})} \qquad \forall \mathbf{o} \in \mathcal{O}^{*l} \quad (5)$$

DF-OPT computes the defender's optimal strategy by considering all possible $\mathbf{o} \in \mathcal{O}^{*l}$ and evaluating her expected utility for each

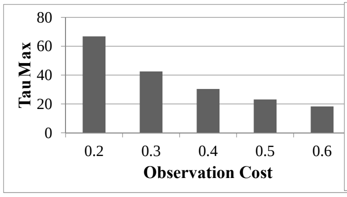

Figure 1: τ_{max}

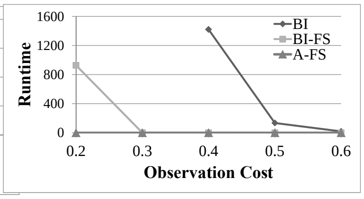

Figure 2: Runtime

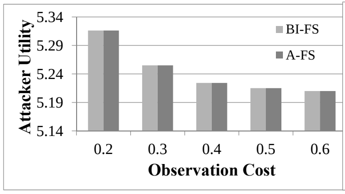

Figure 3: Attacker Utility

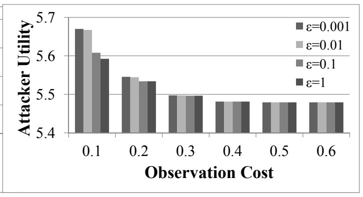

Figure 4: Attacker Utility

observation. Equation (1) is the objective function which maximizes the defender's expected payoff $\sum_{\mathbf{o}\in\mathcal{O}^{*l}} f(\mathbf{o}|\mathbf{x})d^{\mathbf{o}}$ where $d^{\mathbf{o}}$ is the defender's expected utility when the attacker's observation vector is $\mathbf{o}$. Equations (2) and (3) define the feasible strategy space for the defender. Equation (4) defines the marginal coverage for each target given the defender's strategy $\mathbf{x}$. Equation (5) defines the defender's expected payoff $d^{\mathbf{o}} = c_{\psi(\mathbf{o})}R^d_{\psi(\mathbf{o})} + (1 - c_{\psi(\mathbf{o})})P^d_{\psi(\mathbf{o})}$ when the attacker attacks $\psi(\mathbf{o})$ for observation $\mathbf{o}$.

7. EXPERIMENTAL EVALUATION

We compare OPTS against SGLS (in which the attacker takes τ observations of the defender's strategy and the defender plans accordingly) as well as the standard SSE model (in which the attacker has full knowledge of the defender's strategy and the defender plans accordingly). We conduct experiments primarily on randomly-generated instances of security games. R_i^d and R_i^a are drawn independently and uniformly from the range $[0, 10]$. P_i^d and P_i^a are drawn from the range $[-10, 0]$. All experiments are averaged over 20 sample games. Unless otherwise specified, we use 5 targets, 1 defender resource, $\lambda = 0.4$ as the observation cost, and $\alpha_A = 0$ for every $A \in \mathcal{A}$. We use KNITRO version 8.0.0 to solve DF-OPT, SLGS, and SSE.

7.1 Attacker's Decision Making

While OPTS is used to produce the optimal strategy for the defender, this approach first requires the solving of optimal stopping problem for the attacker. We introduced three algorithms for solving the stopping problem (Backward Induction (BI), Backward Induction with Forward Search (BI-FS), and Approximate Forward Search (A-FS)), which will evaluate and compare in this section. For A-FS, the parameters β and ϵ were fixed at 2.0 and 0.001, respectively.

7.1.1 Effect of Observation Cost

The observation cost, λ, is a critical parameter in the attacker's decision making process. For BI, λ directly determines τ_{max}, which is the maximum length of the observation vectors to be explored during backward induction. Similarly, λ influences the convergence rate between $V_{min}^{\tau^b}(\mathbf{o})$ and $V_{max}^{\tau^b}(\mathbf{o})$ for BI-FS as well as between $V_{min}^{\tau^b}(\mathbf{o})$ and $V_{min}^{\tau}(\mathbf{o})$ for A-FS. Thus, we conducted experiments to determine the impact of varying values of λ on our three algorithms with respect to both runtime and attacker utility. For these experiments, we tested λ values of 0.2, 0.3, 0.4, 0.5, and 0.6.

Figure 1 shows how the average value of τ_{max} changes for the different values of λ. These results indicate an exponential increase in τ_{max} as λ is decreased, which is expected given that observation cost is the denominator for the leading term in the equation for τ_{max}. This initial result will help provide insight for the remaining results in this section.

In Figure 2, we evaluate runtime for the λ values with the x-axis indicating the observation cost and the y-axis representing the runtime for computing the observation graph. All three algorithms are able to efficiently compute the observation graph for $\lambda = 0.6$. By decreasing λ from 0.6 to 0.5, we observe a small runtime increase for BI. However, for $\lambda = 0.4$, the runtime dramatically rises by an order of magnitude which indicates that BI cannot scale up for $\lambda < 0.4$. Given the results shown in Figure 1, this runtime increase is to be expected, as the average τ_{max} for $\lambda = 0.4$ is 30. By incrementally increasing τ^b, BI-FS is able to efficiently compute the observation graph for $\lambda = 0.3$, after which point the algorithm experiences a large runtime increase at $\lambda = 0.2$. The runtime for A-FS remains constant for all λ values tested.

The runtime improvement is possible because A-FS does not require either the computation of $V_{max}^{\tau^b}(\mathbf{o})$ or the convergence of $V_{min}^{\tau^b}(\mathbf{o})$ and $V_{max}^{\tau^b}(\mathbf{o})$. There are instances when BI-FS has to compute the observation graph for an unnecessarily large τ^b despite the fact that the attacker's utility did not increase from the previous iteration of the algorithm, as convergence has not yet been reached, i.e., $V_{min}^{\frac{\tau^b}{2}}(\mathbf{o}) = V_{min}^{\tau^b}(\mathbf{o})$ but $V_{min}^{\tau^b}(\mathbf{o}) \neq V_{max}^{\tau^b}(\mathbf{o})$. In such situations, convergence can be slow, depending on the value of λ, as $V_{max}^{\tau^b}(\mathbf{o})$ decreases by at most $\lambda \times \tau^b$ when going from one iteration to the next, while $V_{min}^{\tau^b}(\mathbf{o})$ does not increase. A-FS avoids these unnecessarily large values of τ^b by terminating as soon as the increase in attacker utility from one iteration to the next drops below a threshold, ϵ.

For the same set of λ values, we then compared the attacker utility obtained by the generated observation graphs. Figure 3 shows the results with the x-axis representing the observation cost and the y-axis indicating the attacker's utility, i.e., $V_{min}^{\tau^b}(\mathbf{o})$ for initial state $\mathbf{o} = \langle o_A = 0 \rangle$. From these results, two observations can be made. First, the attacker's utility monotonically increases as the value of λ is decreased. This is resulting from the attacker taking additional observations and gaining a more accurate belief about the defender's strategy. Second, at each value of λ, the attacker's utility is equivalent for both BI-FS and A-FS.

Given the orders of magnitude runtime improvement, it is nonintuitive that A-FS produces no loss in solution quality. To better understand these results, we conducted another set of experiments in which we varied the ϵ parameter for the A-FS algorithm. This parameter specifies the maximum tolerance for determining an approximately optimal solution, $|V_{min}^{\tau^b}(\mathbf{o}) - V_{min}^{\tau}(\mathbf{o})| < \epsilon$. In Figure 4, the x-axis indicates the observation cost, while the y-axis represents the attacker utility for A-FS using different ϵ values. Due to the efficiency of the approximate approach, we included additional data points with $\lambda = 0.1$ These results indicate the A-FS can be suboptimal, and the degree of suboptimality is exaggerated as λ is decreased and ϵ is increased.

7.1.2 Effect of the number of Pure Strategies

Another important factor in the attacker's decision making process is the number of pure strategies for the defender, which is equivalent to the number of observation vectors reachable from any given observation vector $\mathbf{o}$. Thus, as the number of pure strategies is increased, it becomes more difficult to compute the observation

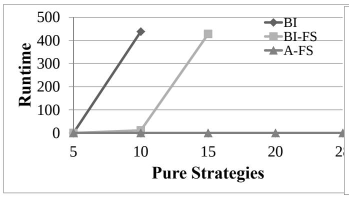

Figure 5: **Runtime**

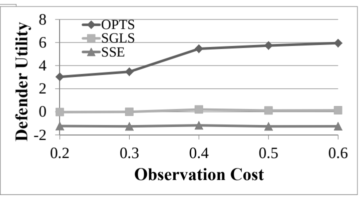

Figure 6: **Defender Utility**

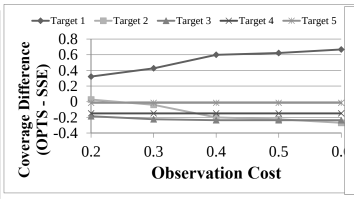

Figure 9: **SSE Comparison**

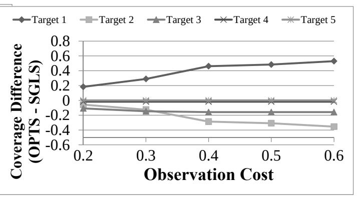

Figure 10: **SGLS Comparison**

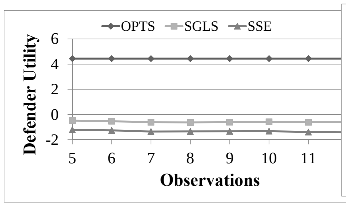

Figure 7: **Defender Utility**

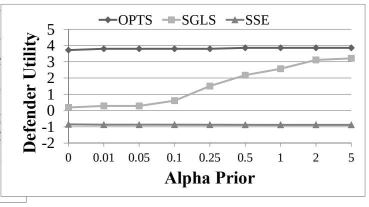

Figure 8: **Defender Utility**

graph. In Figure 5, we confirm this result by calculating the runtime needed to compute the observation graph with $\lambda = 0.6$ for 5 (5 targets, 1 resource), 10 (5 targets, 2 resources), 15 (6 targets, 2 resources), 20 (6 targets, 3 resources), and 28 (8 targets, 2 resources) pure strategies. While all three algorithms have similar runtimes for 5 pure strategies, BI sees a significant runtime increase at 10 pure strategies and is unable to scale up to 15 pure strategies. BI-FS has a similar runtime increase at 15 pure strategies and cannot reach 20 pure strategies. However, A-FS is able to scale all the way up to 28 pure strategies with minimal runtime increase. As described early, the relative efficiency of A-FS is achieved because it terminates at smaller values of τ^b. As the number of pure strategies increases, the number of states in the finite-state MDP, for a given τ^b, increases exponentially. Thus, by A-FS avoiding larger values of τ^b, the disparity in runtime is even further exaggerated as the number of pure strategies is increased.

7.2 Defender's Decision Making

In Section 7.1, we evaluated three algorithms for solving the attacker's stopping problem. We now consider the defender's optimization problem and compare the performance of OPTS using BI-FS against both SGLS and the standard SSE model.

In Figure 6, we compare OPTS, SGLS, and SSE with respect to defender utility, with the x-axis indicating the observation cost λ and the y-axis representing the defender's utility when playing against an attacker who is considering the optimal stopping problem. Based on these results, we see that SGLS consistently outperforms SSE, which in turn is consistently outperformed by OPTS. The constant ordering is indicative of how accurately these different approaches are modeling the type of attacker presented in this paper. The SSE model relies on a number of strong assumptions including that an attacker has perfect knowledge of the defender's strategy. SGLS shows improvement by modeling that the attacker samples the defender's strategy by taking observations. However, by assuming a fixed number of observations, the defender achieves a lower utility by optimizing against observation vectors which are not reachable. In contrast, OPTS determines and optimizes against the exact set of observation vectors which lead to attacks, resulting in significantly higher defender utility.

In Figure 7, instead of changing the observation cost, we evaluate defender utility while varying the number of observations taken in SGLS. The process for incrementing τ^b in BI-FS remains the same. We again observe the same consistent ordering of the three approaches, with OPTS performing the best. For both SGLS and SSE, there is a slight decrease in defender utility as τ is increased because the attacker's belief about the defender's strategy becomes more accurate. The defender utility for OPTS remains constant, as the algorithm and its performance is independent of the number of observations assumed by SGLS.

We performed one last comparison of defender utility for the three approaches, in which we varied the strength of the attacker's prior, represented through the α vector. In these experiments, we assumed a uniform α vector. Figure 8 presents the results for these experiments, in which the x-axis indicates the value assigned to each α_A and the y-axis presents the defender's utility. While OPTS is the top performing approach for all α values tested, its advantage over SGLS diminishes as the value for each α_A is increased. When the weight is high, the attacker will learn very slowly, thus the attacker will start to attack earlier for both OPTS and SGLS in consideration of surveillance cost and thus, the difference is smaller.

In order to better understand the results from Figures 6, 7, and 8 we wanted to analyze the difference in the underlying strategies generated by OPTS, SGLS, and SSE. To accomplish this, we borrowed an experimental setup from [2], in which the defender and attacker are playing a zero-game in which the payoffs for the targets are sorted such that Targets 1 through 5 are valued in decreasing order. If $\lambda = 0$, the attacker would take an infinite number of observations and acquire full knowledge of the defender's strategy. In this situation, the strategies returned by OPTS and SSE would be identical. As λ approaches 0, the attacker will take an increasing number of observations and obtain a more accurate belief about the defender's strategy. Thus, we want to understand how the OPTS strategy converges to the SSE strategy as a function of λ. Figure 9 shows the rate of convergence, with the x-axis indicating the observation cost, while the y-axis represents the difference in coverage between the strategies generated by OPTS and SSE for the five targets. We observed that at $\lambda = 0.6$ there were noticeable differences between the two strategies, with OPTS placing significantly more coverage on the most valuable target, Target 1. However, as λ is decreased we see a noticeable trend toward convergence. We performed a similar set of experiments in Figure 10, where we compared the strategies for OPTS and SGLS. For $\lambda = 0.6$ we still observe differences in the two strategies, though less pronounced than when comparing OPTS and SSE. By $\lambda = 0.2$, the strategies have become quite similar to each other, with OPTS placing more coverage on Target 1 rather than placing it on Targets 2 or 3.

7.3 Robustness Analysis

Up to this point, we have assumed the λ used by OPTS to compute the defender's strategy is, in fact, the true cost of observation incurred by the attacker. However, it is unlikely that this information would be available to the defender and thus would have to be estimated. Therefore, it is important to understand how robust our approach is in the face of this uncertainty over the true value of λ. We perform this analysis by computing the defender strategy (whether produced by OPTS, SGLS, or DOBSS) with a noisy estimation of λ. That strategy is then evaluated against the attacker strategy produced by OPTS using λ (because it is the defender who has uncertainty about the observation cost, not the attacker). The

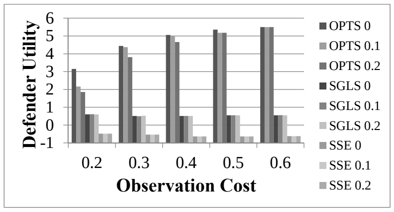

Figure 11: Observation Cost Noise

noisy observation cost, λ_n, is sampled from the uniform distribution $[\lambda, \lambda + p]$, where p is the noise. Only overestimations of λ were considered to avoid the significant runtimes associated with solving OPTS for $\lambda_n \leq 0.2$. We conducted experiments with the three approaches of defender decision making for three values of p (0, 0.1, 0.2) while varying λ. From the results in Figure 11, we again observe that OPTS outperform SGLS and DOBSS for all settings tested. For larger values of λ, the amount of noise has limited impact on the defender utility of OPTS because the attacker is already choosing to attack after minimal observation. As λ is decreased, the presence of noise leads to a decrease in defender utility and this effect is amplified by increased values of p. Since $\lambda_n \geq \lambda$, noise causes the defender to overestimate the value of λ and thus underestimate the amount of observation conducted by the attacker, leading to lower defender utility. So while noisy estimations of λ may hinder OPTS, the algorithm is robust enough to noise such that it surpasses both SGLS and DOBSS even when the latter algorithms have no noise.

8. CONCLUSIONS

This paper provides five key contributions to security games considering attackers' dynamic surveillance decision: (1) We introduce a model of security games with limited surveillance in which attacker dynamically decides when to attack. (2) We show that the attacker's optimal stopping problem can be formulated as a discrete state space MDP. (3) We show an upper bound on the maximum number of observations the attacker can make and thus the stopping problem is equivalent to a finite state MDP. (4) We give mathematical programs to compute optimal attacker and defender strategies. (5) Experimental results show that the defender can gain significantly higher utility by considering the attacker's optimal stopping decision, validating the motivation of our work.

9. ACKNOWLEDGMENTS

We thank David Kempe, Christopher Kiekintveld, Eric Shieh, Satinder Singh, and Xiaoming Sun for their very helpful suggestions. This research was supported by MURI grant W911NF-11-1-0332, National Natural Science Foundation of China (No.61202212, 61035003, 60933004), National Program on Key Basic Research Project (973 Program) (No.2013CB329502), National High-tech R&D Program of China (863 Program) (No.2012AA011003, 2013-AA01A606,), National Science and Technology Support Program (No.2012BA107B02).

10. REFERENCES

[1] B. An, M. Jain, M. Tambe, and C. Kiekintveld. Mixed-initiative optimization in security games: A preliminary report. In *AAAI Spring Symposium on Help Me Help You: Bridging the Gaps in Human-Agent Collaboration*, pages 8–11, 2011.

[2] B. An, D. Kempe, C. Kiekintveld, E. Shieh, S. Singh, M. Tambe, and Y. Vorobeychik. Security games with limited surveillance. In *AAAI*, pages 1241–1248, 2012.

[3] B. An, M. Tambe, F. Ordóñez, E. Shieh, and C. Kiekintveld. Refinement of strong Stackelberg equilibria in security games. In *AAAI*, pages 587–593, 2011.

[4] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, pages 500–503, 2009.

[5] J. P. Dickerson, G. I. Simari, V. S. Subrahmanian, and S. Kraus. A graph-theoretic approach to protect static and moving targets from adversaries. In *AAMAS*, pages 299–306, 2010.

[6] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordóñez. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, pages 689–696, 2009.

[7] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *AAAI*, pages 805–810, 2010.

[8] D. Korzhyk, V. Conitzer, and R. Parr. Solving Stackelberg games with uncertain observability. In *AAMAS*, pages 1013–1020, 2011.

[9] J. Morgan and F. Vardy. The value of commitment in contests and tournaments when observation is costly. *Games and Economic Behavior*, 60(2):326–338, 2007.

[10] J. Pita, M. Jain, M. Tambe, F. Ordóñez, and S. Kraus. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.

[11] J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordóñez, S. Kraus, and P. Parachuri. Deployed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport. In *AAMAS*, pages 125–132, 2008.

[12] E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. PROTECT: A deployed game theoretic system to protect the ports of the United States. In *AAMAS*, 2012.

[13] E. Southers. *LAX - terror target: the history, the reason, the countermeasure*, chapter Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned, pages 27–50. Cambridge University Press, 2011.

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

[15] A. Tversky and D. J. Koehler. Support thoery: A nonextensional representation of subjective probability. *Psychological Review*, 101:547–567, 1994.

[16] E. van Damme and S. Hurkens. Games with imperfectly observable commitment. *Games and Economic Behavior*, 21(1-2):282–308, 1997.

[17] B. von Stengel and S. Zamir. Leadership with commitment to mixed strategies. Technical Report LSE-CDAM-2004-01, CDAM Research Report, 2004.

[18] Z. Yin, M. Jain, M. Tambe, and F. Ordóñez. Risk-averse strategies for security games with execution and observational uncertainty. In *AAAI*, pages 758–763, 2011.

[19] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, , and M. Tambe. Stackelberg vs. nash in security games: interchangeability, equivalence, and uniqueness. In *AAMAS*, pages 1139–1146, 2010.