Combining online learning and equilibrium computation in security games

Richard Klíma[1], Viliam Lisý[1,2], and Christopher Kiekintveld[3]

[1] Department of Computer Science, FEE, Czech Technical University in Prague
klima.richard@gmail.com, viliam.lisy@agents.fel.cvut.cz,
[2] Department of Computing Science, University of Alberta
[3] Computer Science Department, University of Texas at El Paso
cdkiekintveld@utep.edu

Abstract. Game-theoretic analysis has emerged as an important method for making resource allocation decisions in both infrastructure protection and cyber security domains. However, static equilibrium models defined based on inputs from domain experts have weaknesses; they can be inaccurate, and they do not adapt over time as the situation (and adversary) evolves. In cases where there are frequent interactions with an attacker, using learning to adapt to an adversary revealed behavior may lead to better solutions in the long run. However, learning approaches need a lot of data, may perform poorly at the start, and may not be able to take advantage of expert analysis. We explore ways to combine equilibrium analysis with online learning methods with the goal of gaining the advantages of both approaches. We present several hybrid methods that combine these techniques in different ways, and empirically evaluated the performance of these methods in a game that models a border patrolling scenario.

Keywords: game theory, security games, online learning, Stackelberg game, Stackelberg equilibrium, Nash equilibrium, border patrol, multi-armed bandit problem.

1 Introduction

Game theory has become an important paradigm for modeling resource allocation problems in security [23]. Deciding how to deploy limited resources is a core problem in security, and game theoretic models are particularly useful for finding randomized policies that make it difficult for attackers to exploit predictable patterns in the security. There are several examples of successful decision support systems that have been developed using this methodology, including the ARMOR system for airport security [19], the IRIS tool for scheduling Federal Air Marshals [24], and the PROTECT system for scheduling Coast Guard patrols [22]. All of these examples focus primarily on terrorism, where attacks are very infrequent, the adversaries are highly sophisticated, and the stakes of individual events are extremely high. These factors all lead to constructing game models based mostly on inputs from domain experts.

There are many other security domains that are characterized by much more frequent interactions with lower stakes for individual events. These types of domains include border security, cyber security, and urban policing. When there is enough observable data about the actual behavior of attackers, it makes sense to use this data to construct and continually improve the models used for decision making.

However, pure learning/data-driven approaches also have drawbacks: they are entirely reactive, and cannot anticipate adversaries' reactions, they cannot easily incorporate additional information from experts or intelligence, and they can suffer from very poor initial performance during the initial data collection/exploration phase.

We introduce *hybrid* methods that seek to combine the best features of model-based equilibrium analysis and data-driven machine learning for security games with frequent interactions. By using analysis of (imperfect) game models we can warm-start the learning process, avoiding problems with initial poor performance. Using learning allows us to achieve better long-term performance because we are not limited by inaccuracies in a specified model, and we can also adapt to changes in adversary behaviors over time.

The primary motivating domain for our approach is border security, though we believe that our methods are relevant to many other domains with similar features. Border security is a major national security issue in the United States and many other countries around the world. The Customs and Border Protection agency (CBP) is charged with enforcing border security in the United States. The U.S. has thousands of miles of land and sea borders, so CBP faces a very large-scale resource allocation problem when they decide how to allocate infrastructure and patrolling resources to detect and apprehend illegal border crossings. They also have a large amount of data available to inform these resource allocation decisions; in particular, detailed information is available about all apprehensions, including times and precise locations. In principle, this data allows CBP to identify patterns of activity and adopt risk-based resource allocation policies that deploy mobile resources to the areas with the greatest threat/activity levels. The shift to a more data-driven, risk-based strategy for deploying border patrol resources is a major element of the most CBP strategy plan [1].

We study a game with repeated interactions between an attacker and a defender that is designed to capture several of the main features of the border patrol problem. For this model we introduce several hybrid solution techniques that combine Stackelberg equilibrium analysis with online learning methods drawn from the literature on multi-armed bandits. We also introduce variations of these methods for the realistic case where the defender is allowed to allocate multiple patrolling resources in each round (similar to the case of combinatorial multi-armed bandits). We perform an empirical evaluation of our hybrid methods to show the tradeoffs between equilibrium methods and learning methods, and how our hybrid methods can mitigate these tradeoffs.

2 Related Work

There are several lines of work in the area of security games that acknowledge that the game models and assumption about adversary behaviors are only approximations. These models typically focus on finding equilibrium solutions that are robust to some type approximation error. For example, several works have focused on robustness to errors in estimating payoffs, including both Bayesian and interval models of uncertainty [16,15]. Other works have focused on uncertainty about the surveillance capabilities of the attacker [28,3,2,9], or about the behavior of humans who may act with bounded rationality [20,21,27]. Finally, some recent works have combined multiple types of uncertainty in the same model [18].

Our approach is not focused on simply making equilibrium solutions more robust to modeling error, but on integrating equilibrium models with learning methods based on repeated interactions with an attacker. The learning methods we use are drawn from the literature on online learning in multi-armed bandits (MAB), where the focus is on balancing exploration and exploitation. One well-known method for learning a policy for a MAB with fixed distributions is UCB [4], which has also been modified into Sliding-window UCB [13] for situations with varying underlying distributions. The algorithms that most closely fit our setting are for the adversarial MAB problem, where there are no assumptions about the arms having a fixed distribution of rewards, but instead an adversary can arbitrarily modify the rewards. The EXP3 method is one of the most common learning methods for this case [5]. There have been several other recent works that have considered using learning in the context of security games [17,26,30,6], but these have not considered combining learning with equilibrium models. The most closely related work that considers combining learning and equilibrium models is in Poker, where implicit agent models have been proposed that adopt online learning to select among a portfolio of strategies [7,8].

3 Game model

We introduce a game model that captures several important features of resource allocation for border patrol [1]. The core of the model is similar to the standard Stackelberg security game setting [14,23]. The border is represented by a set of K distinct zones, which represent the possible locations where an attacker can attempt to enter the country illegally. There is a defender (i.e., border patrol), denoted by Θ, who can allocate d resources to patrol a subset of the K zones; there are not enough resources to patrol every area all of the time. The attackers, denoted by Ψ, attempt to cross the border without being detected by avoiding the patrolling agents.

An important difference between our model and the standard security game model is that we consider this a repeated game between the attacker and defender that plays out in a series of rounds. This models the frequent interactions

over time between CBP and organized criminal groups that smuggle people, drugs, and other contraband across the border. Each round $t \in 1 \dots N$ in the game corresponds to a time period (e.g., a shift) for which the defender assigns resources to protect d out of the K zones. Attackers attempt to cross in each round, and each individual attacker selects a single zone to cross at.

The utilities in the game are not zero-sum, but follow the standard security game assumption that the defender prefers to patrol zones that are attacked and the attacker prefers to cross in zones that are not patrolled. More precisely, we assume that for any zone the defender receives payoff $x_c^{\Theta} = 1$ if an attacker chooses the zone and it is patrolled by a resource, and $x_u^{\Theta} = 0$ if it is not selected or not patrolled. We assume that the attacker has a zone preference vector $x_u^{\Psi} = \langle v_1^{\Psi}, \dots, v_K^{\Psi} \rangle$, which describes his payoff for crossing a zone if it is not patrolled. This vector can represent how easy/difficult it is to cross a zone because of specific conditions in the terrain (i.e., without the risk of being caught, an attacker would prefer an easy crossing near a city, rather than a dangerous crossing over miles of open desert). If the attacker is apprehended in zone j, he suffers penalty of $\pi^{\Psi} = 0.5$; hence, his payoff is $x_{c,j}^{\Psi} = v_j^{\Psi} - 0.5$. The goal of each player is to maximize the sum of payoffs obtained over all rounds of the game.

An important characteristic of the border patrol domain is limited observability. In particular, the border patrol only gathers reliable information about the illegal entrants they actually apprehend; they do not observe the complete strategy of all attackers.[4] In our model, we capture this by revealing to the defender only the attackers that are apprehended (i.e., the attacker chooses a zone where the defender currently has a resource patrolling). The defender does not observe the attackers that choose zones that are not patrolled. This leads to a classic exploration vs. exploitation problem, since the defender may need to explore zones that appear to be suboptimal based on the current information to learn more about the attacker's strategy. In a long run of the game we overcome the possibility of high, unnoticed immigrant flows in an unpatrolled zone by an extra exploration, which we use in the defender strategies.

As a simplifying assumption, we assume that the attacker observes the whole patrol history of the defender in all zones but does not know the defender strategy vector. At time t, the attacker knows the number of previous rounds in which the defender protected zone j further denoted $h_j^t = c_j^t * (t-1)$. This can be justified in part by the domain, since border patrol agents are more easily observable (i.e., they are uniformed, drive in marked vehicles, etc.), and smuggling organizations are known to use surveillance to learn border patrolling strategies. We also assume the attackers to cooperate and form a gang or a cartel and thus share fully the gained information about the patrols. However, it also allows us to more easily define a simple but realistic adaptive strategy for the attackers to follow in our model based on fictitious play. We describe this behavior in more detail later on.

[4] This is sometimes described as the problem of estimating the total flow of traffic, rather than just the known or observed flow based on detections and apprehensions.

We do not generally assume that the defender knows the attacker's payoffs (i.e., zone preferences). However, when we consider equilibrium solutions we will assume that the defender is able to estimate with some uncertainty the payoffs for the attacker. Formally, the defender will have an approximation of the attacker's preference vector v'^{Ψ}, such that $|v_j^{\Psi} - v_j'^{\Psi}| < \epsilon$ for each j and some ϵ known to both players.

In Figure 1 we present an example of border patrol game, where the defender chooses first a zone to patrol and then the attacker chooses a zone to cross without knowing which specific zone the defender is currently patrolling. There is the zone preference vector v^{Ψ} and patrol history vector h^{100} at round 100. In this example the attacker is apprehended because she chose the same zone as the defender. If the attacker had chosen zone 1 he would have successfully crossed the border because the defender does not patrol it.

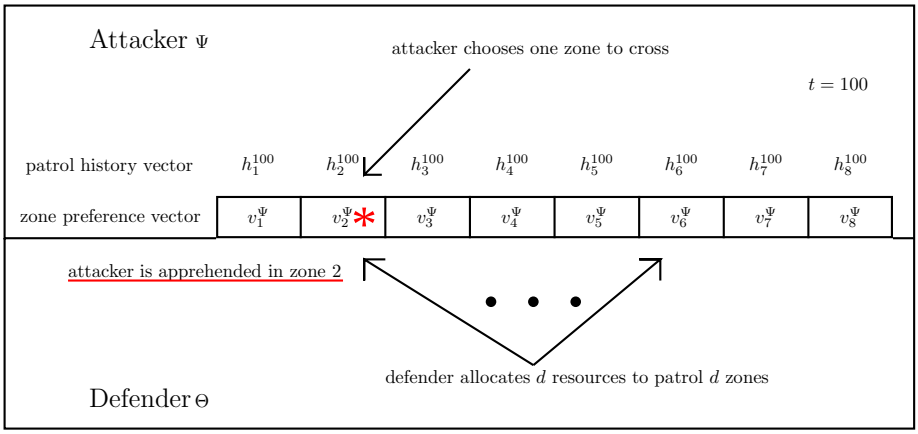


Fig. 1: Border patrol game example

3.1 Attacker Behavior Model

Our main focus in this work will be on designing effective policies for the defender against an adaptive adversary. While there are many ways that the attackers can be modeled as learning and adapting to the defender policy, here we will focus on one simple but natural strategy. We assume that the attackers adapt based on a version of the well-known fictitious play learning policy (e.g., [12]). In fictitious play the player forms beliefs about the opponent strategy and behaves rationally with respect to these beliefs. The standard model of fictitious play assumes the opponent plays a stationary mixed strategy, so the player forms his beliefs about opponent's strategy based on the empirical frequencies of the opponent's play. We define an *adversarial attacker* as an attacker who attacks the zone that maximizes his expected payoff under the assumption that the defender plays a mixed strategy corresponding his normalized patrol history: $j^t = \arg\max_j (v_j^{\Psi} - \pi^{\Psi} * c_j^t)$. This type of the attacker strategy can be seen as the worst-case strategy

compared to some naive attacker strategies, which we also successfully tested our proposed algorithm against. Algorithms for minimizing regret in the adversarial bandit setting are designed to be efficient against any adversary and therefore we expect the proposed combined algorithms to be effective against any attacker's strategy.

We also want to evaluate robustness of the designed strategies to rapid changes of the attacker's behavior. In the real world, these can be introduced by new criminal organization starting operations in the area, or by changes in the demand or tactics used by an organization, such as the adoption of a new smuggling route. Therefore, we introduce also an *adversarial attacker with changes*, which differs form the basic adversarial attacker in having variable preference vector x_u^{Ψ}, that rapidly changes at several points in the game. The defender is not informed about these changes or the time when it happens.

4 Background

4.1 Stackelberg Security Game

Our model of the border patrolling problem is similar to the standard Stackelberg security game model, as described in [14]. The game has two players, the defender Θ and the attacker Ψ. In our model the defender represents the Office of Border Patrol (OBP) and the attacker represents a group of illegal immigrants or a criminal smuggling organization. In security games we usually do not have individuals playing against each other but rather groups of people who have similar or same goal. These groups can represent terrorists, hackers, etc. on the attacker side and officers, authorities, security units etc. on the defender side. These groups use a joint strategy so we can think of the group as an individual player with several resources. The defender has a set of pure strategies, denoted $\sigma_{\Theta} \in \Sigma_{\Theta}$ and the attacker has a set of pure strategies, denoted $\sigma_{\Psi} \in \Sigma_{\Psi}$. We consider a mixed strategy, which allows playing a probability distribution over all pure strategies, denoted $\delta_{\Theta} \in \Delta_{\Theta}$ for the defender and $\delta_{\Psi} \in \Delta_{\Psi}$ for the attacker. We define payoffs for the players over all possible joint pure strategy outcomes by $\Omega_{\Theta} : \Sigma_{\Psi} \times \Sigma_{\Theta} \to \mathbb{R}$ for the defender and $\Omega_{\Psi} : \Sigma_{\Theta} \times \Sigma_{\Psi} \to \mathbb{R}$ for the attacker. The payoffs for the mixed strategies are computed based on the expectations over pure strategy outcomes.

An important concept in Stackelberg security games is the idea of a leader and a follower. This concept is the main difference from the normal-form game. The defender is considered to be the leader and the attacker is the follower. The leader plays first, and then the attacker is able to fully observe the defender strategy before acting. This is quite a strong assumption and it represents very adversarial and intelligent attacker who can fully observe the defender's strategy before deciding how to act. In our model we assume less intelligent attacker who does not know the exact defender strategy as described in Section 3. Formally we can describe the attacker's strategy as a function which chooses a mixed distribution over pure strategies for any defender's strategy: $F_{\Psi} : \Delta_{\Theta} \to \Delta_{\Psi}$.

4.2 Stackelberg equilibrium

Stackelberg equilibrium is a strategy profile where no player can gain by unilaterally deviating to another strategy for the case where the leader moves first, and the follower plays a best response. We follow the standard definition of Stackelberg equilibrium (SE) for security games [14]. This version of Stackelberg equilibrium is known as *strong Stackelberg equilibrium*. The strong SE assumes that in cases of indifference between targets the follower chooses the optimal strategy for the leader. A strong SE exists in every Stackelberg game. The leader can motivate the desired strong equilibrium by choosing a strategy, which is arbitrary close to the equilibrium. This makes the follower strictly better off for playing the preferred strategy.

4.3 Nash equilibrium

Nash equilibrium is a basic concept in game theory for players who move simultaneously. A profile of strategies form a Nash equilibrium if the defender plays a best response s^* that holds $x^{\Theta}(s_i^*, s_{-i}) \geq x^{\Theta}(s_i, s_{-i})$ for all strategies $s_i \in S_i^{\Theta}$ and the attacker plays a best response s^* that holds $x^{\Psi}(s_i^*, s_{-i}) \geq x^{\Psi}(s_i, s_{-i})$ for all strategies $s_i \in S_i^{\Psi}$.

The relationship between strong Stackelberg equilibrium and Nash equilibrium is described in detail in [29]. The authors show that Nash equilibria are interchangeable in security games, avoiding equilibrium selection problems. They also prove that under the SSAS (Subsets of Schedules Are Schedules) restriction on security games, any Stackelberg strategy is also a Nash equilibrium strategy; and furthermore, this strategy is unique in a class of real-world security games.

5 Defender Strategies

The problem the defender faces closely resembles the multi-armed bandit problem, in which each arm represents one of the zones. Therefore, we first explain the online learning algorithms designed for this problem and then we explain how we combine them with game-theoretic solutions.

5.1 Online learning with one resource

First we focus on the problem with a single defender resource ($d = 1$). The defender's problem then directly corresponds to the adversarial multi-armed bandit problem. A standard algorithm for optimizing cumulative reward in this setting is *Exponential-weight algorithm for Exploration and Exploitation* (EXP3), which was introduced in [5]. The algorithm estimates the cumulative sum $s(i)$ of all past rewards the player could have received in each zone using the important sampling correction. If zone i is selected with probability p_i and reward r is received, the estimate of the sum is updated by $s(i) = s(i) + \frac{r}{p_i}$. This ensures that $s(i)$ is an unbiased estimate of the real cumulative sum for that zone. The defender then chooses actions proportionally to the exponential of this cumulative

reward estimate. We use the numerically more stable formulation introduced by [11]. Formally, a given zone i is protected with probability:

$$p_i^{\Theta} = \frac{1-\gamma}{\sum\limits_{j \in K} e^{(s(j)-s(i))\frac{\gamma}{K}}} + \frac{\gamma}{K}, \tag{1}$$

where γ represents the amount of random exploration in the algorithm.

5.2 Online learning with multiple resources

When computing the strategy for multiple defenders ($d > 1$), we could consider each combination of allocations of the resources to be a separate action in a multi-armed bandit problem. It would require the algorithm to learn the quality of each of the exponentially many allocations independently. However, thanks to the clear structure of payoffs from individual zones, the problem can be solved more efficiently as a combinatorial multi-armed bandit problem. We solve it using the COMB-EXP-1 algorithm introduced in [10] and presented here as Algorithm 1.

COMB-EXP-1 algorithm

Initialization: Start with the distribution $q_0(i) = \frac{1}{K}$ and set $\eta = \sqrt{\frac{2d \log K}{KN}}$

for $t = 1, ..., N$ **do**

1. Sample d actions from vector $p_{t-1} = dq_{t-1}$.
2. Obtain the reward vector $X_i(t)$ for all chosen actions i.
3. Set $\bar{X}_i(t) = \frac{1-X_i(t)}{p_{t-1}(i)}$ for all chosen actions i and $\bar{X}_i(t) = 0$ for all other not chosen actions.
4. Update $\bar{q}_t(i) = q_{t-1}(i) \exp\left(-\eta \bar{X}_i(t)\right)$.
5. Compute q_t as a projection of $\bar{q}_t$ to $\mathcal{P} = \{\boldsymbol{x} \in \mathbb{R}^K : \sum_i x_i = 1, x_i \leq \frac{1}{d}\}$ using KL divergence.

end

Algorithm 1: Combinatorial EXP3 learning algorithm

The algorithm starts with a uniform distribution over all zones q_0. In each round, it samples d distinct zones from this distribution using the Algorithm 1, so that the probability of protecting each zone is p_i (line 1). It protects the selected zones and receives reward for each of the selected zones (line 2). It computes the loss vector rescaled by importance sampling (line 3) and updates the probability of protecting individual zones using the exponential weighting (line 4). After the update, vector q_t may not represent a correct probability and not sum to one. Therefore, it must be projected back to the simplex of valid probability distributions (line 5).

Similar to the non-combinatorial EXP3 algorithm, the COMB-EXP-1 algorithm can be numerically unstable if some zone is played with very small probability ($p_t(i) \to 0$). We prevent this instability in our implementation by adding

an uniform vector with very small values (10^{-7}) to the strategy vector q, which bounds the scaled losses $\bar{X}$.

Combinatorial sampling

On line 1, Algorithm 1 samples d zones so that each zone i is protected with probability $p(i)$. We use combinatorial sampling as introduced in [25]. From vector p we create a new cumulative sum vector. For each integer $j \in (1, K)$, let $S_j = \sum_{i<j} p_i$. Based on that we define a disjoint partition of interval $[0, d)$ as $I_j = [S_j, S_j + p_j)$. Interval I_j represents zone j. To sample d zones, we generate single random number y from interval $[0, 1)$ uniformly at random. The selected zones correspond to the intervals that contain points $y, y+1, ..., y+(d-1)$. Since each zone is covered with probability at most 1, no two of these points will be part of the same interval and the probability of hitting interval i is p_i. In Figure 2 there is an example of combinatorial sampling, where we have 6 zones $z_1, ..., z_6$ and 3 resources (defenders). We generate a random number y and sample the intervals created by cumulation from the probability vector p.

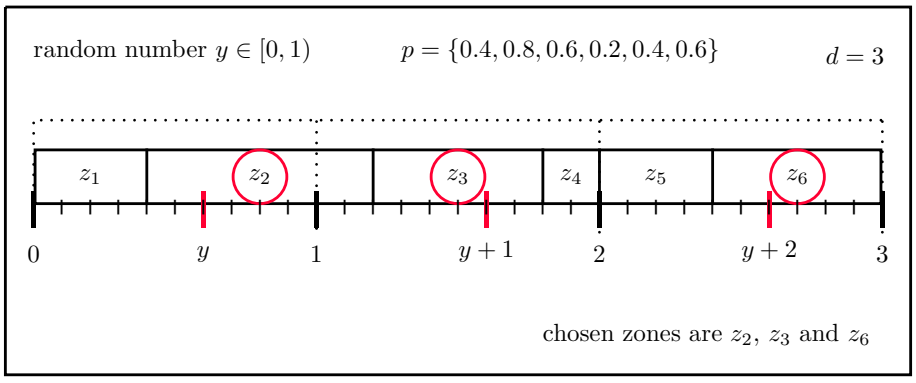


Fig. 2: Combinatorial sampling example

Projection heuristic

The COMB-EXP-1 algorithm requires projection using KL-divergence on line 5. This projection defines a distribution $q \in \mathcal{P}$ which has the minimal KL-divergence from vector $\bar{q}$.

$$q = \arg\min_{p \in \mathcal{P}} KL(p, \bar{q}) \qquad KL(p, q) = \sum_{i \in 1...K} p(i) \log \frac{p(i)}{q(i)} \tag{2}$$

We are not aware of a computationally efficient algorithm for computing such projection. Therefore, we propose a heuristic algorithm H_1, where we decrease all values greater than $1/d$ to $1/d$ and normalize all other values in the vector to sum to $(1 - a/d)$, where a is the number of values in the original vector greater than $1/d$ and d is the number of resources.

We compare our heuristic to another heuristic H_2 where we redistribute the difference value from value in a vector greater than $1/d$ uniformly among other values. In 10000 experiments with randomly generated vectors and with different numbers of resources d, heuristic H_1 was always better than H_2. Further, we tried to randomly perturb the vectors returned by H_1 by small amounts in individual zones, while still ensuring the perturbed vector belongs to $\mathcal{P}$ and we were able to find a better value in less than 1% of cases. We conclude that H_1 is a good approximation of the projection and we use it in the experimental evaluation.

6 Combined Algorithms

In this section we propose four algorithms that combine the online learning algorithms described above with a (possibly inaccurate) game-theoretic solution. The main idea is to start the learning algorithm with some prior information, but allow the algorithms to learn close to optimal solutions even if the initial information is inaccurate.

6.1 Combined algorithm 1

The EXP3 algorithm described in Section 5.1 computes the values $s(j)$ that estimate the cumulative rewards for each zone j. We can initialize the EXP3 algorithm by initializing these values. If both players knew the exact preference vector of the attacker, their optimal static strategy would be the Nash equilibrium (NE) of the game. Due to the security games utility restrictions, this equilibrium is unique [29]. If the attacker was playing an equilibrium attacking zone j with probability NE_j for τ rounds, the cumulative rewards obtainable in individual zones would be $s^{init}(j) = NE_j * \tau$. By using this initialization for the values $s(j)$ in EXP3, it starts from a state similar to the state where it has played τ rounds of the game against the optimal attacker. Since the defender does not have access to the exact preference vector, we compute the approximate Nash equilibrium strategy based on his inaccurate estimate. Algorithm COMB1 than uses it for initialization of $s(i)$ as described above, but otherwise runs the standard EXP3 algorithm.

6.1.1 Combinatorial COMB1

Combinatorial version of the COMB1 algorithms is also based on the intuition of initialization by the estimated equilibrium play. Since COMB-EXP-1 uses the current strategy vector instead of cumulative rewards, we use the defender's strategy for initialization. The algorithms for computing the equilibrium for security games with multiple defender resources, such as [14], directly output the strategy in the form of a coverage vector representing the probability that each

zone will be covered. Let Stackelberg equilibrium (SE) be this coverage vector, than the initial distribution for COMB-EXP-1 is:

$$q_0(i) = \frac{\tau}{d} SE + \frac{1-\tau}{K} \tag{3}$$

where τ is the parameter which sets how confident we are about the Stackelberg equilibrium strategy. The basic setting is $\tau = 0.9$. After initializing the online learning algorithm we continue playing standard combinatorial EXP3.

6.2 Combined algorithm 2

In this combined algorithm, instead of initializing the learning algorithm as if it played based on the equilibrium strategy before the games starts, it actually plays the estimated equilibrium strategy for the first T rounds of the game. Even though the actions are selected based on the equilibrium in these rounds, EXP3 learns form the observed apprehensions. In order to also learn about the zones that are never played in the equilibrium strategy, we add 10% uniform exploration to the strategy.

EXP3 learns by computing the vector of estimates s. This vector is computed from the beginning of the game no matter which strategy the defender uses. For finding the point where to switch from first stage to the second we compute the EXP3 payoff virtually while playing the estimated Nash equilibria. Virtual EXP3 payoff is computed using the importance sampling correction. It gives higher payoff for a strategy with higher probability of visiting a particular zone. If the probability of EXP3 protecting a particular zone with positive payoff is higher than the probability in Nash equilibrium vector, we get a relatively higher payoff for EXP3 than for the NE strategy. In this manner we prioritize the strategy that has the higher estimated payoff. The defender gets covered payoff 1 and uncovered payoff 0 and virtual EXP3 defender covered and uncovered payoff is

$$\bar{x}_c^{\Theta}(t) = \frac{e_i^t}{n_i^t} * 1 \qquad \bar{x}_u^{\Theta}(t) = \frac{e_i^t}{n_i^t} * 0 \tag{4}$$

where e_i^t is the probability of playing zone i in round t by playing EXP3 and n_i^t is the probability of playing zone i in round t by the estimated Nash equilibrium strategy.

We compute the total payoff for both strategies as the sum over all rounds played so far. The algorithm switches to the EXP3 learning algorithm if the cumulative payoff of virtually playing EXP3 exceeds the actual cumulative reward obtained by playing the Nash equilibrium with the additional exploration.

6.2.1 Combinatorial COMB2

Combinatorial COMB2 algorithm is analogous to the standard COMB2 algorithm. We use the estimated Stackelberg equilibrium strategy for multiple resources with 10% extra exploration and combinatorial EXP3 algorithm. We start

with SE strategy and compute virtually expected payoff for playing EXP3. Once the virtual EXP3 payoff becomes greater than actual payoff by playing SE with extra exploration we switch to EXP3 algorithm and use the standard updates.

6.3 Combined algorithm 3

The third combined algorithm is based on a similar concept to the previous one, but in this case we continually switch between two strategies based on which one has the higher current estimated payoff. One of these strategies is based on the estimated equilibrium, and the other is a learning policy. For the strategy we are currently playing we store the total actual payoff and for the other strategy we compute the payoff in the same way we did for virtual play of EXP3 in the previous algorithm. Similar to above, for virtually playing NE strategy the defender gets covered and uncovered payoff

$$\bar{x}_c^{\Theta}(t) = \frac{n_i^t}{e_i^t} * 1 \qquad \bar{x}_u^{\Theta}(t) = \frac{n_i^t}{e_i^t} * 0 \tag{5}$$

Let $\bar{X}_{Alg}^{\Theta}$ be the estimated cumulative payoff of an algorithms, COMB3 plays the estimated Nash equilibria with exploration if $\bar{X}_{EXP3}^{\Theta} < \bar{X}_{NE}^{\Theta}$ or we play EXP3 if $\bar{X}_{EXP3}^{\Theta} > \bar{X}_{NE}^{\Theta}$.

The EXP3 algorithm learns using the expected payoff vector s from all previously played rounds including those rounds when the defender played the NE strategy with exploration.

6.3.1 Combinatorial COMB3

Analogously to the non-combinatorial COMB3 algorithm, combinatorial COMB3 algorithm is a generalization of previous combinatorial COMB2 algorithm. In this COMB3 algorithm we enable the switching between the two strategies arbitrary according to the highest payoff. We compute the virtual SE strategy payoff while playing combinatorial EXP3 algorithm and vice versa.

6.4 Combined algorithm 4

With this algorithm, the defender uses several estimated Nash equilibria corresponding to random modifications of the attacker preference vector by at most ϵ. This models the scenario of building a model based on the input of multiple domain experts, rather than a single expert. There is extra exploration of 10% added to each estimated Nash equilibrium. The main idea is that some of these random variations may be a more accurate estimate of the true preference vector and the algorithm can learn which one from the interaction. COMB4 starts playing with one of the strategies and in parallel computes the expected payoffs for the other estimated Nash strategies and for the EXP3 learning algorithm. In each round, we select an action based on the strategy with the highest current estimate of the cumulative payoff. In our model we did experiments with 3 estimated Nash equilibria (NE).

6.4.1 Combinatorial COMB4

The combinatorial version of this algorithm is practically the same as the non-combinatorial version. The only difference is that the equilibria are computed for multiple defender resources and the learning algorithm is also combinatorial.

7 Experiments

If not otherwise specified, we consider a game model where $K = 8$ (8 zones) and $N = 1000$ (1000 rounds). In the border patrol domain we can consider 1 round as 1 day, so a 1000 round game represents approximately 3 years. All the experiments are run 1000 times to get reliable results. In each of these runs, we generated a new preference vector for the attacker. Each value is i.i.d. and comes from range (0, 1). We compute the estimated preference vector known to the defender by adding a random number from interval $(-\epsilon, \epsilon)$ to each entry. The exploration parameter γ for the learning algorithms has been hand-tuned to $\gamma = 0.2$, i.e., 20% exploration.

7.1 Imprecise Stackelberg equilibrium strategy

We test the influence of different levels of error (ϵ) in the zone preference vector on the performance of the estimated SSE. In Figure 3 we show apprehension rates for different levels of error. We observe the performance of the SSE strategy for $\epsilon \in [0, 0.2]$. The adversarial attacker can learn the strategy and over time the apprehension rate decreases. In particular, for higher values of ϵ there is a large decrease in performance. For $\epsilon \geq 0.15$ we get even worse performance than for playing a random defender strategy, which has the expected payoff 12.5%. For SSE with no error the performance is still very good even after the attacker learns the strategy. In our further experiments we focus on error 0.1, for which the game theoretic strategy is better than random, but there is still room for improvement. The widest mean 95% confidence interval in these experiments is ±0.56% for error 0.1.

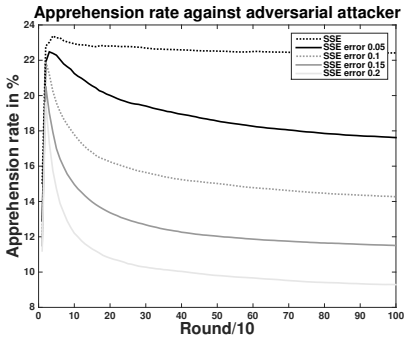
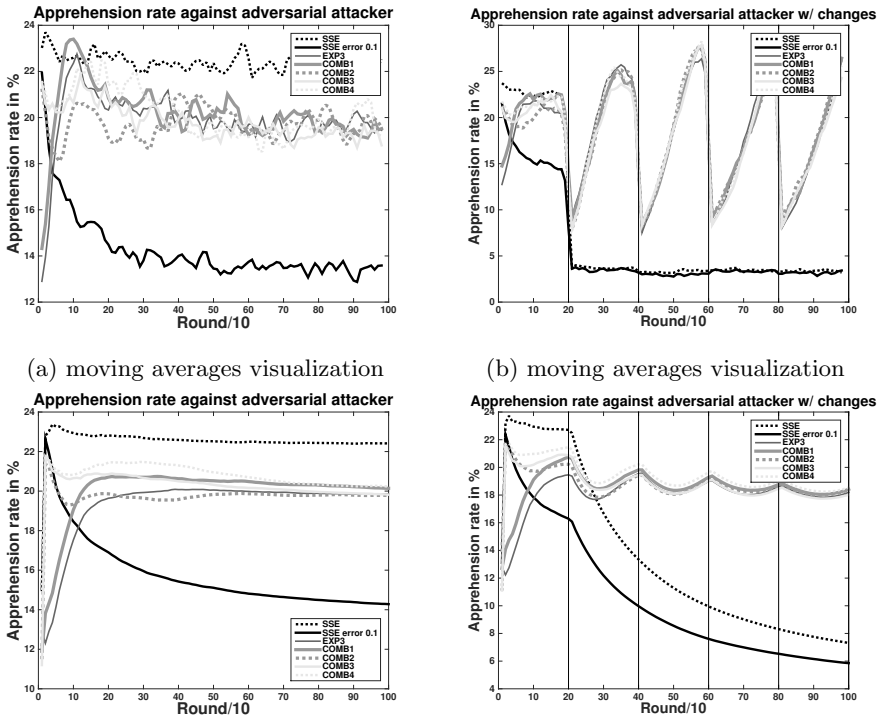


Fig. 3: SSE strategies with different levels of error against an adversarial attacker

7.2 Performance of combined algorithms with one resource

We compare the performance of the EXP3 learning algorithm, Stackelberg equilibrium strategy (SSE), and Stackelberg equilibrium strategy with error (which

is used in the COMB algorithms). For each graph we compute a 95% confidence interval and provide a mean interval width across all rounds.



(a) moving averages visualization

(b) moving averages visualization

(c) adversarial attacker

(d) adversarial attacker w/ changes

Fig. 4: COMB algorithms with 0.1 error against adversarial attacker

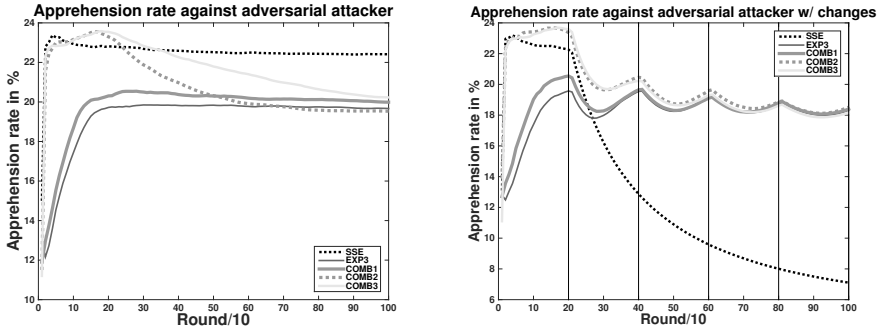
In Figures 4 we use two styles of result visualization to better understand the behavior of the algorithms. One is a moving average of apprehensions in 20 rounds (a,b) and the other is the mean apprehension rate from the beginning of the game (c,d). The moving average better represents the immediate performance of the algorithm and the cumulative performance captures the overall performance of the algorithm. Figure 4a shows the same experiment as Figure 4c and Figure 4b shows the same experiment as Figure 4d. The COMB algorithms use the imprecise game-theoretic solution with error $\epsilon = 0.1$.

In Figure 4c the COMB algorithms have the widest confidence interval $\pm 0.39\%$ and for EXP3 algorithm the width of interval is $\pm 0.30\%$. The mean reward of SSE with error decreases with the attacker learning the strategy. SSE without error gives a very good, stable performance. COMB1 has better but similar performance to EXP3. This comes from the nature of COMB1 algorithm, which is an initialized EXP3. COMB2 algorithm starts with playing SSE with error plus some extra exploration and then switches permanently to EXP3. We can see that

this switch occurs close to the intersection of SSE with error and EXP3 algorithm which is a desired feature of COMB2 algorithm. COMB3 outperforms COMB2, which is caused by better adaptability to the intelligent attacker. COMB4 has the best performance out of all COMB algorithms and also outperforms EXP3 algorithm. COMB2, COMB3 and especially COMB4 algorithms have very good performance for the first half of the game (up to round 500) and outperform EXP3 and SSE with error. At the end of our game COMB algorithms and EXP3 algorithm have similar performance, which is caused by the attacker learning the defender strategy, also the COMB algorithms tend to play EXP3 later in the game.

In games against the adversarial attacker with changes in Figure 4d COMB algorithms have the maximal width of confidence interval $\pm 0.32\%$ and for EXP3 algorithm the width of interval is $\pm 0.26\%$. This figure shows one of the main advantages of the learning algorithm. If we assume that we are not able to detect a change in the attacker payoff and therefore to compute the appropriate game-theoretic solution, we can intuitively expect a poor performance by playing this game-theoretic strategy. In these figures the changes in the attacker's preference vector are highlighted every 200 rounds by black horizontal lines.

The SSE with error strategy and the SSE strategy have almost same performance after the first change in the attacker zone preference vector, because the equilibria are computed for the initial zone preference vector and after the change they have no relation to the real preference vector of the attacker. We can see that COMB algorithms can successfully adapt to these changes in less than 200 rounds and even slightly outperform EXP3 algorithm in the whole run. At the beginning of our game all COMB algorithms are better than the EXP3 algorithm. The COMB algorithms can adapt to these changes because they make use of EXP3 algorithm and can switch to it in case they need to. So the COMB algorithms retain the desired property of learning algorithms.



(a) adversarial attacker (b) adversarial attacker w/ changes

Fig. 5: COMB algorithms with no error against adversarial attacker

In order to separate the behavior of the learning algorithms from the effects of the error in the computed equilibrium, we further evaluate the combined algorithms with precise game-theoretic solution. Figure 5a presents experiments against the adversarial attacker. The widest confidence interval for COMB algorithm is $\pm 0.39\%$ and for EXP3 the width is $\pm 0.29\%$. In this figure we do not visualize COMB4 since it is identical to COMB3 in this case. The COMB2 and COMB3 algorithms get even better than the SSE strategy, because for the attacker it is more difficult to learn the defender strategy if it is not static. This is partly caused by the extra exploration in the COMB algorithm playing SSE, which can confuse the attacker. The attacker learns quite fast against a static defender strategy vector SSE. One can observe that even though COMB2 and COMB3 outperform the SSE strategy for a short period of time, it then drops substantially in performance due to the attacker eventually learning the strategy. The apprehension rate of the COMB algorithms decreases under the SSE strategy even though they use this SSE strategy, because there is the extra 10% exploration added to SSE strategy. Nevertheless we can see that COMB algorithms significantly outperform EXP3 algorithm for the first half of the game and then they all converge to a similar performance.

In Figure 5b we test the COMB algorithms using the precise game-theoretic solution against the adversarial attacker with changes. For COMB algorithms the widest interval is $\pm 0.32\%$ and for EXP3 algorithm the width of interval is $\pm 0.26\%$. The COMB algorithms can react well to changes in the attacker strategy because of the learning algorithm part. If the defender has a precise SSE strategy he might prefer playing it instead of any other strategy in the case of the adversarial attacker however if there are some changes in the attacker payoff matrix the defender would be better off by playing some more sophisticated algorithm like EXP3 or preferably one of the proposed COMB algorithms, because these can adapt to the changes in the attacker behavior over time.

Now we focus on the convergence of the algorithms in a substantially longer time window. Figure 6 presents the COMB algorithms using game-theoretic solution with error against adversarial attacker for 10000 rounds. This experiment is done 100 times for each setting. The maximal mean width of confidence intervals for COMB algorithms is 0.99% and the width of confidence interval for EXP3 is 0.92%. We can see that COMB algorithms and EXP3 algorithm converge to the same performance quite quickly. Playing precise Stackelberg equilibrium strategy has the best performance however the SSE strategy with 0.1 error gives quite poor results. The precise SSE strategy performance increases

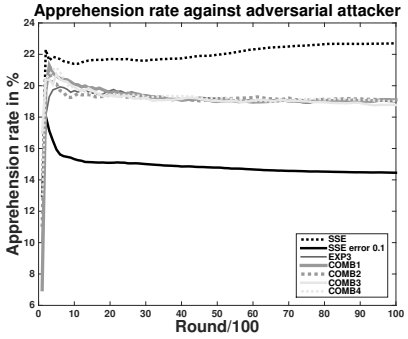
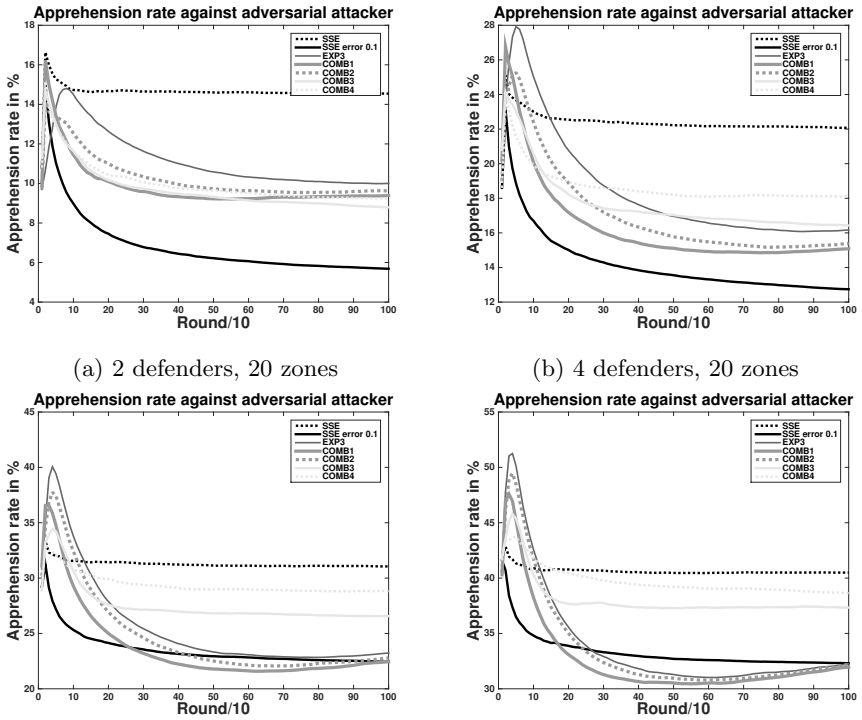


Fig. 6: Convergence of COMB algorithms against adversarial attacker

during the time, which is caused by the attacker learning more precisely the defender strategy and therefore there are more ties in the attacker strategy which the attacker breaks in favor of the defender.

7.3 Combinatorial Combined Algorithms

In this section we focus on the combinatorial case where the defender uses multiple resources so he can patrol d zones in each round where $d > 1$. We test combinatorial variants of COMB algorithms which use combinatorial variant of EXP3 as described in Section 5.2. For brevity, we continue to refer to the combinatorial variant as EXP3. The experiments are done for a larger game model with 20 zones ($K = 20$). We compare the strategies in models with 2, 4, 6 and 8 defender resources ($d = 2, 4, 6, 8$). These experiments are run 1000 times for each setting and each game has 1000 rounds.



(a) 2 defenders, 20 zones

(b) 4 defenders, 20 zones

(c) 6 defenders, 20 zones

(d) 8 defenders, 20 zones

Fig. 7: COMB algorithms against adversarial attacker, 0.1 error in SSE, varying number of defenders

In Figures 7 there are 4 COMB algorithms, SSE with error and SSE without error strategies. The widest mean confidence interval in all the figures is ±0.36%.

We observe in Figure 7a that EXP3 outperforms the COMB algorithms, which is caused by poor performance of the SSE with error strategy. The EXP3 algorithm gives almost 2 times better performance than SSE with error strategy, because there are too few defenders for too many zones and even a small error in the SSE strategy causes a low apprehension rate. Due to this fact, the COMB algorithms have worse performance than EXP3.

When we increase the number of defenders to 4 in Figure 7b, SSE with error does better and so do the COMB algorithms. COMB3 outperforms EXP3 algorithm after the half of the game and COMB4 does even better than COMB3, which comes from the nature of the algorithms. One can observe interesting peaks of the performance curves at the beginning of the game, which are caused by increasing the number of defenders. The attacker needs time to learn effectively against multiple defenders and at the beginning he plays poorly. However by the steepness of the algorithms curves we can see that the attacker learns very quickly after playing very badly at the beginning. These described features are even stronger with increasing number of defenders in Figure 7c and in Figure 7d.

The SSE strategy with error approaches even more closely the performance of EXP3 algorithm because the more defenders there are, the less the error in the SSE strategy vector matters. The defender still chooses the zones with high probabilities even though there are some errors, because these 0.1 errors cannot decrease the real values too much to not be chosen. For the last figure with 8 defenders the SSE with error strategy even outperforms EXP3 algorithm. Nevertheless COMB3 and especially COMB4 algorithms have very strong performance and approach to SSE strategy performance. COMB1 and COMB2 have obvious drawbacks in the limited use of SSE with error strategy. COMB1 use the game-theoretic strategy only to initialize EXP3 and then cannot make use of it anymore and similarly for COMB2 algorithm, which uses the game-theoretic strategy at the beginning and then permanently switches to EXP3 algorithm.

8 Conclusion

We argue that security games with frequent interactions between defenders and attackers require a different approach than the more classical application domains for security games, such as preventing terrorist attacks. Game theoretic models generally require a lot of assumptions about the opponent's motivations and rationality, which are inherently imprecise, and may even change during the course of a long-term interaction. Therefore, it may be more efficient to learn the optimal strategy from the interaction. However, the standard methods for online learning in adversarial environment do not provide ways to incorporate the possibly imprecise knowledge available about the domain.

We propose learning algorithms that are able to take into consideration imprecise domain knowledge that can even become completely invalid at any point during the game. We further show how to efficiently extend these algorithms to allow for the combinatorial case with multiple defender resources. We show that these algorithms achieve significant improvement on the performance of learning

algorithms in the initial stages of the game as well as significant improvement to using only an imprecise game theoretic model in the long run. On the other hand, especially in the combinatorial case, it may be better to use the EXP3 learning algorithm without any knowledge if we expect the performance of imprecise game theoretic solution to be very low. With increasing quality of this solution it is quickly beneficial to use the proposed COMB3 or COMB4 algorithm. Even in the cases where EXP3 outperforms the COMB algorithms, the COMB algorithms still have a very good performance due to using EXP3 as their main component. In a sufficiently long time period all of the COMB algorithms converge to long-run performance of the EXP3 algorithm, and they retain the theoretical guarantees that make EXP3 attractive in adversarial settings.

Future work could focus on a formal analysis of the proposed combined algorithms. For example, it may be possible to derive and prove improved regret bounds which would provide further guarantees on the algorithm performance. Another direction for future work is bringing defender action preferences into the game model, which would better reflect real-world applications.

Acknowledgements

This research was supported by the Office of Naval Research Global (grant no. N62909-13-1-N256) .

References

1. 2012–2016 border patrol strategic plan. U.S. Customs and Border Protection, 2012.
2. B. An, M. Brown, Y. Vorobeychik, and M. Tambe. Security games with surveillance cost and optimal timing of attack execution. *AAMAS*, pages 223–230, 2013.
3. B. An, C. Kiekintveld, E. Shieh, S. Singh, M. Tambe, and Y. Vorobeychik. Security games with limited surveillance. *AAAI*, pages 1241–1248, 2012.
4. P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47:235–256, 2002.
5. P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2001.
6. M.-F. Balcan, A. Blum, N. Haghtalab, and A. D. Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *ACM Conference on Economics and Computation (EC-15)*, pages 61–78, 2015.
7. N. Bard, M. Johanson, N. Burch, and M. Bowling. Online implicit agent modelling. *AAMAS*, pages 255–262, 2013.
8. N. Bard, D. Nicholas, C. Szepesvari, and M. Bowling. Decision-theoretic clustering of strategies. *AAMAS*, pages 17–25, 2015.
9. A. Blum, H. Nika, and A. D. Procaccia. Lazy defenders are almost optimal against diligent attackers. *AAAI*, pages 573–579, 2014.
10. R. Combes, M. Lelarge, A. Proutiere, and M. S. Talebi. Stochastic and adversarial combinatorial bandits. *arXiv:1502.03475*, 2015.
11. P. I. Cowling, E. J. Powley, and D. Whitehouse. Information set monte carlo tree search. *IEEE Transaction on Computational Intelligence and AI in Games*, pages 120–143, 2012.

12. D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. The MIT Press, 1998.
13. A. Garivier and E. Moulines. On upper-confidence bound policies for non-stationary bandit problems. *ALT*, pages 174–188, 2011.
14. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordonez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. *AAMAS*, pages 689–696, 2009.
15. C. Kiekintveld and V. Kreinovich. Efficient approximation for security games with interval uncertainty. *AAAI*, pages 42–45, 2012.
16. C. Kiekintveld, J. Marecki, and M. Tambe. Approximation methods for infinite Bayesian Stackelberg games: Modeling distributional payoff uncertainty. *AAMAS*, pages 1005–1012, 2011.
17. R. Klima, C. Kiekintveld, and V. Lisy. Online learning methods for border patrol resource allocation. *GAMESEC*, pages 340–349, 2014.
18. T. H. Nguyen, A. Jiang, and M. Tambe. Stop the compartmentalization: Unified robust algorithms for handling uncertainties in security games. *AAMAS*, pages 317–324, 2014.
19. J. Pita, M. Jain, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. ARMOR security for los angeles international airport. *AAAI*, pages 1884–1885, 2008.
20. J. Pita, M. Jain, F. Ordonez, M. Tambe, and S. Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence Journal*, 174(15):1142–1171, 2010.
21. J. Pita, R. John, R. Maheswaran, M. Tambe, and S. Kraus. A robust approach to addressing human adversaries in security games. In *European Conference on Artificial Intelligence (ECAI)*, pages 660–665, 2012.
22. E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. Direnzo, G. Meyer, C. W. Baldwin, B. J. Maule, and G. R. Meyer. PROTECT : A Deployed Game Theoretic System to Protect the Ports of the United States. *AAMAS*, pages 13–20, 2012.
23. M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.
24. J. Tsai, S. Rathi, C. Kiekintveld, F. Ordóñez, and M. Tambe. IRIS - A tools for strategic security allocation in transportation networks. *AAMAS*, pages 37–44, 2009.
25. J. Tsai, Z. Yin, J.-y. Kwak, D. Kempe, C. Kiekintveld, and M. Tambe. Urban security: Game-theoretic resource allocation in networked physical domains. *AAAI*, pages 881–886, 2010.
26. R. Yang, B. Ford, M. Tambe, and A. Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. *AAMAS*, pages 453–460, 2014.
27. R. Yang, C. Kiekintvled, F. Ordonez, M. Tambe, and R. John. Improving resource allocation strategies against human adversaries in security games: An extended study. *Artificial Intelligence Journal (AIJ)*, 195:440–469, 2013.
28. Z. Yin, M. Jain, M. Tambe, and F. Ordonez. Risk-averse strategies for security games with execution and observational uncertainty. *AAAI*, pages 758–763, 2011.
29. Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: Interchangeability, equivalence, and uniqueness. *AAMAS*, pages 1139–1146, 2010.
30. C. Zhang, A. Sinha, and M. Tambe. Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. *AAMAS*, pages 1351–1359, 2015.