Regret Minimization Algorithms for the Follower's Behaviour Identification in Leadership Games

Lorenzo Bisi, Giuseppe De Nittis, Francesco Trovò, Marcello Restelli, Nicola Gatti
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Milano, 20133, Italy
lorenzo.bisi@mail.polimi.it, {giuseppe.denittis, francesco1.trovo, marcello.restelli, nicola.gatti}@polimi.it

Abstract

We study for the first time, a leadership game in which one agent, acting as *leader*, faces another agent, acting as *follower*, whose behaviour is not known *a priori* by the leader, being one among a set of possible behavioural profiles. The main motivation is that in real-world applications the common game-theoretical assumption of perfect rationality is rarely met, and any specific assumption on bounded rationality models, if wrong, could lead to a significant loss for the leader. The question we pose is *whether* and *how* the leader can learn the behavioural profile of a follower in leadership games. This is a "natural" *online identification* problem: in fact, the leader aims at identifying the follower's behavioural profile to exploit at best the potential non-rationality of the opponent, while minimizing the regret due to the initial lack of information. We propose two algorithms based on different approaches and we provide a regret analysis. Furthermore, we experimentally evaluate the pseudo-regret of the algorithms in concrete leadership games, showing that our algorithms outperform the online learning algorithms available in the state of the art.

1 INTRODUCTION

The study of scenarios in which multiple strategic agents interact is a challenging problem that is central in Artificial Intelligence from many years. The modelling of these scenarios can be elegantly achieved by means of *non-cooperative game theory* tools (Fudenberg and Tirole, 1991), while the task of solving a game is in many cases an open problem, in which the most suitable techniques to adopt strictly depend on information available to the agents. Two extreme situations can be distinguished: when all the information about the game is common to the players (e.g., utility functions and rationality—either perfect or bounded), the problem is basically an *optimization problem*, solvable by means of techniques from *operations research* (Shoham and Leyton-Brown, 2008), conversely, when players have no information about the opponents, the problem is a *multi-learning problem*, and *learning* techniques are commonly employed (Tuyls and Weiss, 2012). Some attempts were also done to pair these two approaches, allowing agents to play at the equilibrium if the opponent is rational and to play off the equilibrium learning to exploit her at best otherwise (Conitzer and Sandholm, 2007).

Recently, there has been an increasing interest in *leadership* games, where an agent—called *leader*—publicly commits to a strategy and subsequently another agent—called *follower*—observes the commitment and then takes her decision. Such a paradigm has been successfully employed in a number of applications in the security domain (Basilico et al., 2017; Pita et al., 2008; Tsai et al., 2009), where a *defender* (acting as leader) must protect some targets in an environment from an *attacker* (acting as follower), who aims at compromising such targets without being detected. The success of leadership games in real-world applications is due to a number of reasons: committing to a strategy is the best the leader can do, the equilibrium finding problem is conceptually simple since the follower can merely play her best response to the commitment of the leader without any strategic reasoning about the leader's behaviour, and the solution is unique except degeneracy. The crucial issue is that in real-world applications the follower may be not perfectly rational, not necessarily playing her best response to the leader's commitment. For instance, a terrorist could decide either to attack a target that is not patrolled, since she is sure to not be caught, or a target not so valuable itself, but that would cause a huge panic reaction in the population (e.g., this is what happened in November 2015 in Paris attacks at the Bataclan theatre). The same challenge may be faced by a company that aims at planning the production of a product and has to decide when and how it is convenient to enter the market when another company is already the leader in such a market—this is the well-known *Stackelberg duopoly* (Von Stackelberg, 1934). Whenever the assumption of perfect rationality is not met, each agent may in principle exploit her opponent's strategy.

In the present paper, we focus on leadership games in which the follower may be not rational. The literature provides a number of models of bounded rationality (An et al., 2013; Nguyen et al., 2013). Probably, the most elegant one is the Quantal Response (QR) (McFadden, 1984), which fixes the probability distribution over the non-optimal actions of an agent on the basis of their optimality gap. The crucial issue is that all the works on bounded rationality make an assumption about the specific behaviour of the opponent and this assumption could be never met in real-world applications. In that case, such an assumption may lead to an arbitrarily loss for the leader. Differently from the existing literature, we study the original single-agent-learning problem in which the behaviour of the follower is one among a set of possible behavioural profiles—e.g., the rational one (i.e., best response), a rationally bounded one based on the QR, a stochastic strategy—and the leader does not initially know it, but she can learn it by exploiting the opponent's behaviour at best. Our goal is to design online learning techniques capable to identify the behaviour of the follower while minimizing the regret due to the initial lack of information. We propose a set of algorithms based on sequential learning techniques (Bubeck et al., 2012) that are able to infer the behaviour of the follower the leader is playing against exploiting the repeated interactions between the two players.

Original contributions The main original contributions we provide in this paper are as follows. We define a novel scenario in which a leader plays against a follower whose behaviour is unknown but it belongs to a set of known profiles. We show that state-of-the-art bandit and expert algorithms—suitable for our problem—suffers from a linear and logarithmic regret, respectively, in the length of the time horizon. Thus, we introduce two novel approaches to deal with our problem, bridging together game-theoretical techniques and online learning tools. In the first approach, the leader has a *belief* about the follower and updates it during the game. We name the algorithm *Follow the Belief* (FB) and we provide a finite-time analysis showing that the regret of the algorithm is constant in the length of the time horizon. In the second approach, namely *Follow the Regret* (FR), the learning policy is driven directly by the estimated expected regret and is based on a backward induction procedure. Finally, we provide a thorough experimental evaluation in concrete leadership settings inspired to security domains, comparing our algorithms with the main algorithms available in the state of the art of the online learning field and showing that our approaches provide a remarkable improvement in terms of expected pseudo-regret minimization.

2 RELATED WORKS

Here, we mention the main works related to ours. We mainly refer to the literature on security games since most of the works on leadership games with bounded rationality and/or learning deal with these games. Security games model the problem of finding the optimal schedule of scarce resources when facing strategic adversaries. Many of them deal with real-world problems, e.g., in Pita et al. (2008) game theoretic techniques have been applied to ensure the security of the Los Angeles International Airport (LAX), in Tsai et al. (2009) the authors exploit the Stackelberg paradigm to study how to schedule undercover federal air marshals on domestic U.S. flights, while in Pita et al. (2011) such paradigm is employed to allocate the Transportation Security Administration (TSA) scarce resources to provide protection within several U.S. airports. A higher degree of interaction among the agents is captured in Basilico et al. (2017), where an alarm system to detect potential attacks is introduced. The main issue is that such works only deal with a fully rational attacker while in real-life scenarios attackers might be rationally bounded.

Bounded rationality has been introduced in security games models in the so called Green Security Games (GSGs), a generalization of Stackelberg games (Fang et al., 2015). A remarkable example is Qian et al. (2014), in which the problem of protecting natural resources from illegal extraction is studied: since such extractions are frequent, it is possible for the defender to *learn* the distribution of the resources analyzing the attacker's behavior. A recent application in which an *ad hoc* adaption of the QR function, named *Subjective Utility Quantal Response* (SUQR) (Nguyen et al., 2013), has been employed is the prevention from poachers, who hunt endangered species (Ford et al., 2014; Yang et al., 2014). Here, the QR is employed to model the non-rational behavior of the poachers. In a similar setting, Qian et al. (2016) analyze the problem in which the defender is aware only of the attack activities at targets they protect, modeling it with Restless Multi-Armed Bandit and using Whittle index policy to compute patrol strategies.

In security games, Balcan et al. (2015); Blum et al.

(2015); Paruchuri et al. (2008) deal with a single rational attacker whose preferences may be of multiple types in Bayesian fashion. Specifically, the different attackers are discriminated according to the evaluations they give to the targets, thus leading to the problem of solving Bayesian Stackelberg Games.

The main limitation of all the aforementioned works is that the defender plays against an attacker whose behavioral profile is *a priori* known, while in real-world situation it may be unknown. When dealing with sequential decision learning problems, a customary approach consists in exploiting Multi-Armed Bandit (MAB) techniques, as done by Klíma et al. (2014) and Xu et al. (2016). Even though both works focus on minimizing the expected regret, the different actions corresponding to the arms are the possible targets that may be chosen, while in our work we are discriminating among different attacker types.

3 PROBLEM FORMULATION

Although our work can be employed in principle for any leadership scenario, for the sake of clarity, we focus on security domains, thus referring to the leader as *defender* and to the follower as *attacker*.

Let us consider a 2-player normal-form repeated game $\mathcal{G}_N$ defined over a finite number of rounds $N \in \mathbb{N}$, where a defender D and an attacker A play against each other in some environment with some valuable targets $\mathcal{M} = \{1, \dots, M\}$, characterized by values $\mathbf{v} = (v_1, \dots, v_M)^T, v_m \in (0, 1]$. The goal of the defender D is to protect such targets while the attacker A aims at compromising them. The space of actions of D and A is given by the set of targets such that D chooses the target to protect, while A chooses the target to attack. The course of the game is represented in Figure 1. Specifically, at each round $n \in \{1, \dots, N\}$, the defender D announces the strategy she commits to $\boldsymbol{\sigma}_{D,n} \in \Delta_M$ (Line 1), where Δ_M denotes the M-dimensional simplex, while A observes such a commitment (Line 2). Then, they concurrently play their action over the target space (Line 3), i.e., the defender plays actions $i_{D,n} \in \mathcal{M}$ according to $\boldsymbol{\sigma}_{D,n}$ while A, the follower, plays $i_{A,n} \in \mathcal{M}$ according to some attacker model $\boldsymbol{\sigma}_A(\boldsymbol{\sigma}_{D,n}) \in \Delta_M$. The game is zero-sum: if D and A choose the same target at round n, they both get a utility equal to 0, conversely, if A attacks the i-th target while D decides to protect the j-th one, A gets v_i and D gets $-v_i$ since she lost the target. More concisely, the defender incurs in the *loss* (Line 4):

$$l_n := v_{i_{A,n}} \mathcal{I}\{i_{A,n} \neq i_{D,n}\}, \tag{1}$$

not suffering from any loss if both players select the same target.[1] Hereafter, we assume that the defender is able to compute the best response strategy $\boldsymbol{\sigma}^*_D(A) \in \Delta_M$ if she is given the attacker model she is playing against. Similarly, we denote with $\boldsymbol{\sigma}^*_A(\boldsymbol{\sigma}_D) \in \Delta_M$ the best response A plays against strategy $\boldsymbol{\sigma}_D$ of D. According to such assumption, we can compute the expected loss of D against a generic attacker A as:

$$L(A) := \sum_{m \in \mathcal{M}} \sigma_A(\boldsymbol{\sigma}^*_D(A))_m \, v_m \, (1 - \sigma^*_D(A)_m), \tag{2}$$

where $\sigma_\cdot(\cdot)_m$ is the probability associated with target m by the strategy.

```
for each $n \in \{1, \dots, N\}$
    1. $D$ publicly commits to a strategy $\boldsymbol{\sigma}_{D,n}$
    2. $A$ observes the strategy $D$ committed to
    3. $D$ and $A$ play $i_{D,n}$ and $i_{A,n}$, respectively
    4. $D$ incurs in loss $l_n$ according to Equation (1)
```

Figure 1: Leader-follower interaction

The problem we study in this work is defined as follows.

Definition 1. *The Follower's Behaviour Identification in Security Games (FBI-SG) problem is a tuple $(\mathcal{G}_N, \mathcal{A}, A_{k^*})$, where $\mathcal{G}_N$ is a 2-player normal-form repeated game and $\mathcal{A} = \{A_1, \dots, A_K\}$ is a set of possible attacker behavioural profiles, with $A_{k^*} \in \mathcal{A}$ denoting the actual profile of the attacker in $\mathcal{G}_N$, unknown to the defender D.*

In this work, we cast the FBI-SG as a sequential decision learning problem, where, at each round n, the defender aims at selecting her best response to the attacker in order to identify the actual attacker profile $A_{k^*} \in \mathcal{A}$ while minimizing the loss suffered during the learning process.

Definition 2. *A* policy $\mathfrak{U}$ *is an algorithm able to provide at each round n a strategy profile $\boldsymbol{\sigma}_{D,n}$ for the defender D. Formally:*

$$\mathfrak{U}(h_n) := \boldsymbol{\sigma}_{D,n},$$

where h_n is the history collected so far, i.e., all the strategies declared by the defender $\{\boldsymbol{\sigma}_{D,1}, \dots, \boldsymbol{\sigma}_{D,n-1}\}$, the actions played by the two players $\{i_{D,1}, i_{A,1}, \dots, i_{D,n-1}, i_{A,n-1}\}$ in the past rounds and the corresponding losses $\{l_1, \dots, l_{n-1}\}$.

We evaluate the performance of a given policy $\mathfrak{U}$ over a finite-time horizon of N rounds by means of the expected

[1] Hereafter, we denote with $\mathcal{I}\{E\}$ the indicator function of a generic event E.

cumulative *pseudo-regret*, defined as:

$$R_N(\mathfrak{U}) = \mathbb{E}\left[\sum_{n=1}^{N} l_n\right] - L^* N,$$

where $L^* := L(A_{k^*})$ is the expected loss incurred by the defender if she plays the best response to the actual attacker A_{k^*}, l_n is the loss incurred by using the policy $\mathfrak{U}$ at round n and the expectation $\mathbb{E}[\cdot]$ is taken w.r.t. the stochasticity of the attacker strategy, the defender policy and the policy $\mathfrak{U}$. The goal of a generic policy $\mathfrak{U}$ is to minimize the pseudo-regret $R_N(\mathfrak{U})$ incurred while learning the true attacker's profile.

4 ANALYSED ATTACKER PROFILES

In this section, we describe the different attacker profiles we study in this work and formalize the definition of the attacker strategy $\boldsymbol{\sigma}_{A_{k^*}}(\cdot)$ for two sets of attackers, grouped depending on their ability to change their behavior w.r.t. the strategy D commits to. Specifically, on one side, we take into account stochastic attackers, which disregard the strategy of D, on the other, we focus on strategy-aware attackers, able to modify their strategies depending on the defender announced strategy $\boldsymbol{\sigma}_{D,n}$.

4.1 STOCHASTIC ATTACKER

The first class of attackers is the *Stochastic* (Sto) one, where the attacking player does not take into account the strategy $\boldsymbol{\sigma}_{D,n}$ announced by the defender D and thus has a fixed probability over time to attack the available targets. This class of attackers models opponents focused on specific targets and whose preferences are not influenced by the defender behaviour. At round n, a stochastic attacker Sto plays according to the strategy:

$$\boldsymbol{\sigma}_{Sto}(\boldsymbol{\sigma}) = \mathbf{p}(Sto) \quad \forall \boldsymbol{\sigma} \in \Delta_M,$$

where $\mathbf{p}(Sto) \in \Delta_M$ is a probability distribution over the targets, which is known to D. In this case, the defender best response $\boldsymbol{\sigma}^*_D(\boldsymbol{\sigma}_{Sto})$ is defined as:

$$\sigma^*_D(Sto)_m = \begin{cases} 1 & \text{if } m = \arg\max_{i \in \mathcal{M}} \{v_i\, p(Sto)_i\} \\ 0 & \text{otherwise} \end{cases}.$$

4.2 STRATEGY AWARE ATTACKER

The second class of attackers we examine in this paper consists of strategy aware attackers, corresponding to followers able to modify their strategy depending on the strategy of the defender D. In particular, we study *Stackelberg* (Sta) attackers (Von Stackelberg, 1934), who are able to exploit the information provided by strategy profile declared by the defender D and optimally respond to it, and SUQR attackers (Nguyen et al., 2013), having bounded rationality and being capable to partially exploit the information provided by the defender, disregarding heavily patrolled targets.

Stackelberg Attacker Given a strategy profile declaration $\boldsymbol{\sigma}_{D,n}$, a Stackelberg attacker Sta responds with:

$$\boldsymbol{\sigma}_{Sta}(\boldsymbol{\sigma}) = \arg\max_{\boldsymbol{\sigma}' \in \Delta_M} \sum_{m \in \mathcal{M}} \sigma'_m\, v_m\, (1 - \sigma_m)$$

and the defender best-responds to this attacker is:

$$\boldsymbol{\sigma}^*_D(Sta) = \arg\min_{\boldsymbol{\sigma}' \in \Delta_M} \max_{\boldsymbol{\sigma} \in \Delta_M} \sum_{m \in \mathcal{M}} \sigma'_m\, v_m\, (1 - \sigma_m),$$

as reported in (Conitzer and Sandholm, 2006), where it is proved that, for 2-player zero-sum games, the optimal mixed strategy for the leader to commit to is equivalent to computing the minmax strategy, i.e., to minimize the maximum expected utility that the opponent can obtain.

SUQR Attacker The SUQR attacker responds to the commitment $\boldsymbol{\sigma}_{D,n}$ as:

$$\sigma_{SUQR}(\boldsymbol{\sigma})_m = \frac{\exp\{-\alpha\sigma_m + \beta v_m + \gamma\}}{\sum_{h=1}^{M} \exp\{-\alpha\sigma_h + \beta v_h + \gamma\}},$$

where $\alpha \in \mathbb{R}^+$, $\beta, \gamma \in \mathbb{R}$ are parameters known to the defender, characterizing the attacker and depending on the underlying application. In this case, we do not have a closed form for the best response, but we can compute the minmax solution to the problem following the steps taken in (Yang et al., 2011). We will refer to $\boldsymbol{\sigma}^*_D(SUQR)$ as the best response to an attacker with a SUQR profile.

5 IDENTIFYING THE ATTACKER

Initially, we describe how the state-of-the-art techniques can be adapted to address the FBI-SG problem. Direct approaches are provided by MAB (Bubeck et al., 2012) and expert (Cesa-Bianchi and Lugosi, 2006) algorithms, where arms/experts represent the different attacker behavioural profiles in $\mathcal{A}$. These are general-purpose techniques not exploiting the structure of the problem we are tackling. Summarily, MAB algorithms do not use the expert feedback to learn the attacker behaviour, while expert algorithms do not differentiate among feedbacks received after the defender committed to different strategies. We show below the regret obtained when these algorithms are used in a FBI-SG problem.

When using MAB algorithms, we are able to directly apply the derivation of an upper bound over the pseudo-regret available in the literature to our problem. We can state the following result for the case of UCB1 (Auer et al., 2002).

Theorem 1 (UCB1 Pseudo-regret upper bound)**.** *Let us consider an instance of the FBI-SG problem and apply the UCB1 algorithm, where each possible behavioural profile $A_k \in \mathcal{A}$ is an arm which receives reward $-l_n$ if played. Then, we incur in the following pseudo-regret:*

$$R_N(\mathfrak{U}) \leq 8 \sum_{k \neq k^*} \frac{\ln N}{(\Delta L_k)} + \left(1 + \frac{\pi^2}{3}\right) \sum_{k \neq k^*} \Delta L_k,$$

where $\Delta L_k = \sum_{m=1}^{M} \sigma_{A_{k^*}}(\boldsymbol{\sigma}_D^*(A_k))_m \, v_m \, (1 - \sigma_D^*(A_k)_m) - L^*$ is the expected regret of playing the best response to attacker A_k when the real attacker is A_{k^*}.

When using an expert algorithm, for instance Follow the Perturbed Leader (FPL) (Cesa-Bianchi and Lugosi, 2006), we could exploit an (expert) feedback over all arms since we can compute the expected loss also for the attacker profiles that have not been played at turn n. Nevertheless, if the attacker is strategy aware and we adopt an expert feedback, D incurs in a linear regret. We formally state this result in the following theorem.

Theorem 2 (Expert pseudo-regret upper bound)**.** *Let us consider an instance of the FBI-SG problem and apply the FPL algorithm, where each possible profile A_k is an expert and receives, at round n, an expert reward equal to minus the loss she would have incurred observing $i_{A_{k^*},n}$ by playing the best response to the attacker A_k. Then, there always exists an attacker set $\mathcal{A}$ s.t. the defender D incurs in an expected pseudo-regret of:*

$$R_N(\mathfrak{U}) \propto \Delta L_k \, N.$$

The proof of Theorem 2 is reported in Appendix A for reasons of space. The above results show that MAB techniques provide, in the general case, better guarantees than expert algorithms, assuring a worst-case pseudo-regret of $O(\ln\, N)$ vs. $O(N)$.

In the following, we propose two different techniques that effectively exploit the information both on stochastic and strategy aware attackers, providing better guarantees over the worst-case pseudo-regret. The first algorithm, *Follow the Belief* (`FB`), conducts the learning process taking into account the belief of the learner about the different behavioural profiles. The second method, *Follow the Regret* (`FR`), is based on a value iteration algorithm over the belief space that minimizes the expected regret over the next rounds.

5.1 FOLLOW THE BELIEF

Algorithm 1 `FB`

```
1: $\mathcal{P} = \mathcal{A}$
2: for all $A' \in \mathcal{P}$ do
3:     $b_1(A') = \frac{1}{K}$
4: for all $n \in \{1, \dots, N\}$ do
5:     Select $A_{k_n} = \arg\max_{A' \in \mathcal{P}} b_n(A')$
6:     Play $\boldsymbol{\sigma}_D^*(A_{k_n})$
7:     Observe attacker action $i_{A_{k^*},n}$
8:     for all $A' \in \mathcal{P}$ do
9:         if $\sigma_{A'}(\boldsymbol{\sigma}_D^*(A_{k_n}))_{i_{A_{k^*},n}} = 0$ then
10:            $\mathcal{P} \leftarrow \mathcal{P} \setminus A'$
11:        else
12:            Compute $b_{n+1}(A')$ with Equation (3)
```

The pseudo-code of `FB` is presented in Algorithm 1. At the beginning, `FB` initializes a set of active attackers $\mathcal{P} = \mathcal{A}$ and a belief $b_1(A_k) = 1/K$ for all the attacker profiles $A_k \in \mathcal{P}$ (Lines 1-3). At each round n, the algorithm selects the attacker A_{k_n} for which the belief is the largest one (where ties are broken arbitrarily), best responds with the strategy $\boldsymbol{\sigma}_D^*(A_{k_n})$ and observes the action actually played by the attacker $i_{A_{k^*},n}$ (Lines 4-7). After that, the belief is updated as follows:

$$b_{n+1}(A') = \frac{w_n(A')}{\sum_{A \in \mathcal{P}} w_n(A)}, \tag{3}$$

where $w_n(A) = b_n(A') \, \sigma_{A_{k^*}}(\boldsymbol{\sigma}_D^*(A_{k_n}))_{i_{A_{k^*},n}}$ (Lines 8-12). In other words, the algorithm updates the likelihood of the sequence of the actions for each profile in $A' \in \mathcal{P}$ according to the observed action $i_{A_{k^*},n}$ at round n (Line 12). If the realization $i_{A_{k^*},n}$ is not consistent for attacker A' (zero likelihood), profile A' is removed from $\mathcal{P}$ (Line 10).

Let $b_{kj,t} := \mathbb{E}_{\sigma_D^*(A_j)}[B_{k,t}]$, be the expected value of the belief we get for attacker A_k when we are best responding to A_j and the true type is $A_{k^*} \neq A_k$ and denote with $\Delta b_k := \min_{j | A_j \in \mathcal{A}} \ln(b_{k^* j,t}) - \ln(b_{kj,t})$ the minimum difference of such values. We can upper bound the regret of `FB` algorithm as stated by the following theorem.

Theorem 3 (`FB` pseudo-regret upper bound)**.** *Given an instance of the FBI-SG problem s.t. $\Delta b_k > 0$ for each $A_k \in \mathcal{A}$ and applying* `FB`*, the defender incurs in a pseudo-regret of:*

$$R_N(\mathfrak{U}) \leq \sum_{k=1}^{K} \frac{2(\lambda_k^2 + \lambda_{k^*}^2)\Delta L_k}{(\Delta b_k)^2},$$

where $\lambda_k := \max_{m \in \mathcal{M}} \max_{\boldsymbol{\sigma} \in \mathcal{S}} \ln(\sigma_{A_k}(\boldsymbol{\sigma})_m) - \min_{m \in \mathcal{M}} \min_{\boldsymbol{\sigma} \in \mathcal{S}} \ln(\sigma_{A_k}(\boldsymbol{\sigma})_m) \mathcal{I}\{\sigma_{A_k}(\boldsymbol{\sigma})_m \neq 0\}$ is the range where the logarithm of the beliefs realizations lies (excluding realizations equal to zero, which end the exploration of a profile) and $\mathcal{S} := \cup_k \boldsymbol{\sigma}_D^*(A_k)$ is the set of the available best response to the attackers profile.

For space reasons, we report the proof of Theorem 3 in Appendix A. Comparing the derived results, we notice that the FB algorithm presents an upper bound over the pseudo-regret that is strictly better than that of MAB algorithms, i.e., a constant regret $O(1)$ in N vs. a logarithmic one $O(\ln N)$.

5.2 FOLLOW THE REGRET

FB adopts the belief as discriminant factor to select the strategy profile to play in the next round. Conversely, in what follows, we describe the FR algorithm which is driven by a value iteration procedure that directly minimizes the expected regret over the remaining rounds $\{n+1, \dots, N\}$. In principle, one should perform the procedure until the last round N, but, for computational purposes, an approximate solution can be obtained by setting a maximum level of recursion $h_{\max}$ and carry on the optimization only on the rounds $\{n+1, \dots, \min\{n+h_{\max}, N\}\}$.

Algorithm 2 FR($h_{\max}$)

```
1: for all $A_k \in \mathcal{A}$ do
2:     Initialize $b_k^{(1)} = \frac{1}{K}$
3: for all $n \in \{1, \dots, N\}$ do
4:     $\hat{\mathbf{R}}$ = RE($1, \mathbf{b}^{(n)}, h_{\max}$)
5:     Select $A_{k_n}$ s.t. $k_n = \arg\min_t \hat{R}_t$
6:     Play $\boldsymbol{\sigma}_D^*(A_{k_n})$
7:     Observe attacker action $i_{A_{k^*},n}$
8:     for all $A_k \in \mathcal{A}$ do
9:         Compute $b_k^{(n+1)}$ according to Equation (6)
```

Algorithm 3 RE($h, \mathbf{b}, h_{\max}$)

```
1: for all $A_k \in \mathcal{A}$ do
2:     for all $(i, j) \in \mathcal{M}^2$ do
3:         for all $A_t \in \mathcal{A}$ do
4:             $\hat{b}_t \leftarrow b_t \, \sigma_{A_t}(\boldsymbol{\sigma}_D^*(A_k))_j$
5:         $\hat{\mathbf{b}} \leftarrow \frac{\hat{\mathbf{b}}}{\sum_m \hat{b}_m}$
6:         Compute $r_{ij,k}$ according to Equation (4)
7:         if $h < h_{\max}$ then
8:             $\tilde{\mathbf{R}}$ = RE($h+1, \hat{\mathbf{b}}, h_{\max}$)
9:             $r_{ij,k} \leftarrow r_{ij,k} + \min_k \tilde{R}_k$
10:     Compute $\hat{R}_k$ according to Equation (5)
11: Return $\hat{\mathbf{R}}$
```

The pseudo-code of the FR algorithm is presented in Algorithm 2, which recursively exploits the subroutine Algorithm 3. At first, the FR algorithm requires to initialize a belief vector $b_k^{(1)} = \frac{1}{K}$ for each attacker $A_k \in \mathcal{A}$ (Line 2, Alg. 2). At each round n, the algorithm computes the estimated expected regret vector $\hat{\mathbf{R}}$ suffered by D if she plays the best response $\boldsymbol{\sigma}_D^*(A_k)$ to A_k for each attacker profile $A_k \in \mathcal{A}$ (Line 4, Alg. 2), by recursively calling the *Regret Estimator* (RE) algorithm. Here, for every possible attacker $A_k \in \mathcal{A}$ and for every pair of possible actions of the defender and the attacker $(i, j) \in \mathcal{M}^2$, we create a new belief vector $\hat{\mathbf{b}}$ by updating $\mathbf{b}$ according to the information the attacker played action j (Line 4, Alg. 3). After that, we compute $r_{ij,k}$, i.e., the estimated expected loss in the case the defender D plays action $i_{D,n} = i$ and the attacker A_k plays $i_{A_k,n} = j$ averaged over the beliefs $b_n(A)$, as follows:

$$r_{ij,k} = v_j \mathcal{I}\{i \neq j\} - \sum_{t \in \{1, \dots, K\}} \hat{b}_t \, L(A_t). \qquad (4)$$

If the maximum recursion level $h_{\max}$ has been reached, the above value corresponds to the total estimated expected regret, otherwise we recursively compute the regret by calling RE over the following rounds and sum it to the instantaneous one $r_{ij,k}$ (Line 9, Alg. 3). Finally we compute the estimated total regret of choosing a specific attacker A_k for the next turn (Line 10, Alg. 3) as follows:

$$\begin{aligned}\hat{R}_k := &\sum_{i=1}^{M} \sum_{j=1}^{M} r_{ij,k} \, \sigma_D^*(A_k)_i \\ &\cdot \sum_{A_{k'} \in \mathcal{A}} b_{k'} \, \sigma_{A_{k'}}(\boldsymbol{\sigma}_D^*(A_k))_j,\end{aligned} \qquad (5)$$

where the regret $r_{ij,k}$ is weighted with the probabilities that action i is selected by D and action j is selected by A. The defender D plays, for the current round n, the best response to the attacker A_{k_n}, providing the minimum estimated expected regret $\hat{R}_{k_n}$ (Line 6, Alg. 2) and observing action $i_{A_{k^*},n}$ undertaken by the attacker A_{k^*}. Finally, the algorithm updates the beliefs (Line 9, Alg. 2) as follows:

$$b_k^{(n+1)} = \frac{w_{nk}}{\sum_{k' \in \{1, \dots, K\}} w_{nk'}}, \qquad (6)$$

where $w_{nk} = b_k^{(n)} \, \sigma_{A_k}(\boldsymbol{\sigma}_D^*(A_{k_n}))_{i_{A_{k^*},n}}$.

5.3 COMPUTATIONAL COMPLEXITY

In this section, we analyse the proposed algorithms from a computational perspective. FB has complexity $O(KN)$, since it performs a belief update for each of the K attacker profiles, repeating this operation over N rounds. Thus, it results being linear both in the number of profiles and the rounds the game is played. Conversely, FR requires much more computational time. Indeed, for each attacker profile K, we consider M actions for both players and update the expected regret over the K profiles current beliefs. This leads to

Table 1: Number and type of attacker profiles $\mathcal{A}$ used for the experiments and total number of attacker K.

	Sta	Sto	$SUQR$	U	K
C_1	1	1	-	-	2
C_2	1	-	1	-	2
C_3	1	1	1	-	3
C_4	1	5	-	-	6
C_5	1	-	5	-	6
C_6	1	5	5	-	11
C_7	1	5	5	1	12

a cost of $O(M^2K^2)$ for a single round and an overall computational cost of $O(M^2K^2N)$ over the problem horizon N. If we want to employ the strategy from the current round n to the end of the horizon to compute the estimated expected regret $\hat{R}_n(A_k)$ by means of a forward procedure, the computational cost required by FR is $O(M^{2(N-n)}K^{2(N-n)})$ for a round. Thus, the final computational cost required by FR is $\sum_{n=1}^{N} O(M^{2(N-n)}K^{2(N-n)}) = O\left(\frac{(MK)^{2N}-1}{(MK)^2-1}\right) \approx O(M^{2N}K^{2N})$.

6 EXPERIMENTAL EVALUATION

We compare the proposed algorithms FB and FR (with $h_{\max} = 1$) with the state-of-the-art online learning approaches from the MAB (Bubeck et al., 2012) and expert (Kalai and Vempala, 2005) fields. In particular, We evaluate UCB1 algorithm (Auer et al., 2002), from the MAB literature, and the FPL algorithm (Cesa-Bianchi and Lugosi, 2006), from the expert literature.

In the experiments we also analyse the case in which one of the attacker behavioural profiles, namely U, is stochastic and her strategy is unknown to the defender D (to avoid possible misunderstandings, let us notice that the stochastic behaviour we describe in Section 4 is based on the assumption that the defender knows the strategy). In this case, we are still able to allow the leader to commit to a strategy that somehow minimizes the expected loss. Indeed, we can assign:

$$\boldsymbol{\sigma}^*_{D,n}(U) = FPL(h_n),$$

where $FPL(\cdot) \in \Delta_M$ is the pure strategy prescribed by the FPL algorithm. In this case the algorithm suffers from an additional regret due to the fact that, even if it is able to correctly detect the profile, it does not know the best response $\boldsymbol{\sigma}^*_D(U)$, but it needs to learn it over time.

6.1 EXPERIMENTAL SETTING

The experimental setting is as follows. We use a time horizon of $N = 1000$ rounds, with a different amount of targets $M \in \{5, 10\}$ and different profile configurations C_i, listed in Table 1, in which we report also the number of different stochastic, SUQR, and unknown stochastic behavioural profiles for each configuration. The configurations are ordered from the ones with smallest number of behavioural profiles ($K = 2$) to the largest one ($K = 12$). In principle, these problems should be of increasing difficulty, since the algorithms have to identify the actual behaviour among a larger number of options.

The strategies of the stochastic behavioural profiles Sto are drawn from a Dirichlet distribution with $\boldsymbol{\theta} = \mathbf{1}_M$ (uniform distribution over Δ_M) and the target values $\mathbf{v}$ are uniformly sampled in $[0, 1]^M$. The parameters for the SUQR behavioural profiles are drawn from a uniform probability distribution over the intervals $\alpha \in [5, 15]$, $\beta \in [0, 1]$ and $\gamma \in [0, 1]$, whose choice is motivated by the experimental results obtained by Nguyen et al. (2013). For each combination of behavioural profiles and targets size, 10 random configurations (i.e., target values $\mathbf{v}$ and attacker profile sets $\mathcal{A}$) are generated and the actual behavioural profile A_{k^*} is drawn from a uniform probability distribution over the given profiles set $\mathcal{A}$. For each configuration we run 100 independent experiments and we compute the average regret. We evaluate the performance in terms of expected pseudo-regret $R(\mathfrak{U})_n$ with $n \in \{1, \dots, N\}$ and computational time spent by the algorithms to execute a single run ($N = 1000$ rounds). Each component of the noise vector z in FPL is drawn from a uniform probability distribution over the interval $[0, \hat{v}K\sqrt{N}]$, where $\hat{v} = \max_{m \in M} v_m$, as described in Cesa-Bianchi and Lugosi (2006), Chapter 4.

6.2 EXPERIMENTAL RESULTS

We report in Table 2 the empiric pseudo-regret obtained in the experimental results. It can be observed that the algorithms we propose dramatically outperform the baselines provided by the state of the art. Furthermore, there is no strong statistical evidence that one algorithm between FB or FR outperforms the other. We recall that FR is more computationally demanding than FB, thus one might prefer FB for problems with many attacker behavioural profiles, since it has comparable performance w.r.t. FR and is computationally more efficient. Notably, the FPL algorithm generally improves its performance when tested over larger target space $M = 10$. We think this could be induced by the fact that the specific configurations in which the FPL gets linear regret (i.e., the ones considered in Theorem 2) are less likely to occur when we have a larger amount of targets. Remarkably, our algorithms provide good performance also when a stochastic behavioural profile U whose strategy is unknown to the defender is present among the possible ones.

Table 2: Expected pseudo-regret $R_N(\mathfrak{U})$ over $N = 1000$ rounds and corresponding 95% confidence intervals for different configurations (best results are in boldface).

		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$M = 5$	UCB1	14.12 ± 1.88	8.62 ± 3.73	23.92 ± 5.23	45.75 ± 11.68	1.76 ± 0.41	75.82 ± 19.94	62.31 ± 12.22
	FPL	18.71 ± 35.02	11.16 ± 5.98	38.5 ± 27.18	49.8 ± 62.33	0.77 ± 0.12	68.88 ± 64.13	72.5 ± 53.34
	FB	$\mathbf{0.19 \pm 0.13}$	$\mathbf{0.2 \pm 0.18}$	$\mathbf{0.5 \pm 0.24}$	$\mathbf{0.48 \pm 0.2}$	$\mathbf{0.09 \pm 0.03}$	$\mathbf{0.67 \pm 0.2}$	$\mathbf{7.92 \pm 4.87}$
	FR	$\mathbf{0.1 \pm 0.06}$	$\mathbf{0.27 \pm 0.36}$	$\mathbf{0.42 \pm 0.3}$	$\mathbf{0.62 \pm 0.24}$	$\mathbf{0.07 \pm 0.04}$	$\mathbf{1.07 \pm 1.1}$	$\mathbf{4.84 \pm 3.32}$
$M = 10$	UCB1	16.77 ± 1.2	5.24 ± 2.79	21.2 ± 3.76	60.58 ± 8.89	4.24 ± 5.02	61.52 ± 22.48	58.93 ± 17.42
	FPL	1.08 ± 0.2	5.97 ± 3.5	12.06 ± 4.31	2.63 ± 0.99	3.24 ± 3.96	17.69 ± 16.03	$\mathbf{22.49 \pm 12.26}$
	FB	$\mathbf{0.13 \pm 0.03}$	$\mathbf{0.1 \pm 0.02}$	$\mathbf{0.33 \pm 0.16}$	$\mathbf{0.57 \pm 0.17}$	$\mathbf{0.05 \pm 0.01}$	$\mathbf{0.58 \pm 0.14}$	$\mathbf{16.06 \pm 6.89}$
	FR	$\mathbf{0.06 \pm 0.05}$	$\mathbf{0.12 \pm 0.21}$	$\mathbf{0.21 \pm 0.12}$	$\mathbf{0.43 \pm 0.19}$	$\mathbf{0.02 \pm 0.02}$	$\mathbf{0.6 \pm 0.43}$	$\mathbf{14.65 \pm 8.1}$

In Figures 2 to 7 we show how the pseudo-regret $R_n(\mathfrak{U})$ evolves during the time horizon in the most challenging configurations, namely C_5, C_6 and C_7. The results in other configurations, omitted due to reasons of space, confirm the results obtained in C_5, C_6, C_7 and are reported in Appendix B. The plots are in a semilogarithmic scale for a better comprehension. In all the presented configurations, except in C_7 with $M = 10$, there is statistical significance that the FB and FR algorithms outperform the baselines on average since the confidence intervals do not overlap after the first ≈ 50 rounds. In configuration C_7 with $M = 10$, our algorithms outperform the baselines only on average.

Finally, we analyze the computational effort required by our algorithms to solve instances over $N = 1000$ rounds and $M \in \{5, 10, 20, 40\}$ targets.[2] The average computational times are reported in Table 3 (the full version of Table 3 with confidence intervals is reported in Appendix B). There are three observations we can make. First, we could not report the values for $M \in \{20, 40\}$ for FR since the required computational cost is too high (≥ 3600 seconds). Second, both FB and FR present the same trend w.r.t. the configurations: in fact, when the behavioral profile of the opponent can only be either Sta or Sto, both algorithms are twice more efficient than in cases in which SUQR adversaries are introduced. This is due to the fact that both Sta and SUQR models exploit the strategy the defender commits to, making more difficult to distinguish among them. The most difficult configuration is C_7, where the presence of a stochastic unknown adversary make things even worse since the distribution must also be estimated. Finally, as expected, we notice that FB is *always* faster than FR: in fact, while they are both polynomial in the actions available to the players, i.e., the number of targets, the former is linear while the latter quadratic (since we set $h_{\max} = 1$).

Table 3: Computational time in seconds needed by FB and FR to solve an instance over $N = 1000$ rounds.

		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$M = 5$	FB	6	11	12	4	24	15	15
	FR	77	121	170	146	652	1029	1114
$M = 10$	FB	10	22	23	7	63	47	48
	FR	356	679	887	960	4402	7527	7292
$M = 20$	FB	33	222	138	34	485	227	229
	FR	–	–	–	–	–	–	–
$M = 40$	FB	105	2061	1412	129	2348	1634	1643
	FR	–	–	–	–	–	–	–

7 CONCLUSIONS AND FUTURE RESEARCH

In this work, we study for the first time, a novel leadership game in which the leader plays against a follower whose behaviour is unknown, but it belongs to a set of known profiles. We provide two novel approaches to tackle this problem, namely FB and FR, bridging together game-theoretical techniques and online learning tools. In the FB algorithm the leader is driven by the beliefs on the possible follower profiles, while the FR one is based on a learning policy directly driven by the estimated expected regret, computed according to a value iteration procedure. For the first approach, we provide also a finite-time analysis, showing that the regret of the algorithm is constant in the number of rounds, while bandit and expert algorithms available in the state of the art suffer from a logarithmic and linear regret, respectively. Finally, we experimentally evaluate the performance of our algorithms in leadership settings inspired by concrete security domains, showing that our approaches provide a remarkable improvement in terms of empirical pseudo-regret minimization w.r.t. the main algorithms available in the state of the art of the online learning field.

In the future, we will study an upper bound over the regret of the FR algorithm. Furthermore, we will include new types of attacker profiles and we will extend the framework towards a multi-agent-learning setting, allowing the attacker to exploit a finite/infinite memory.

[2] The computational times for the UCB1 and FPL algorithm are omitted since they are in line with the one of FB.

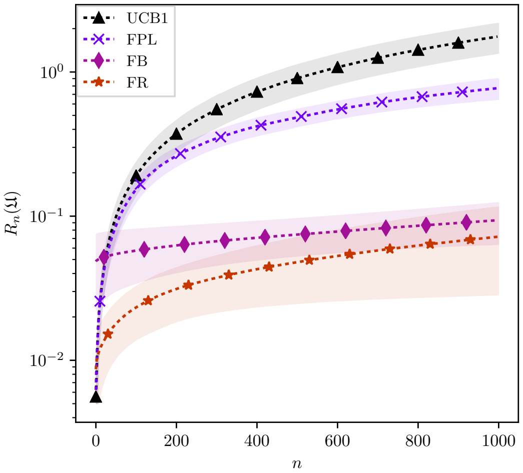

Figure 2: Expected pseudo-regret for the configuration C_5 with $M = 5$ targets.

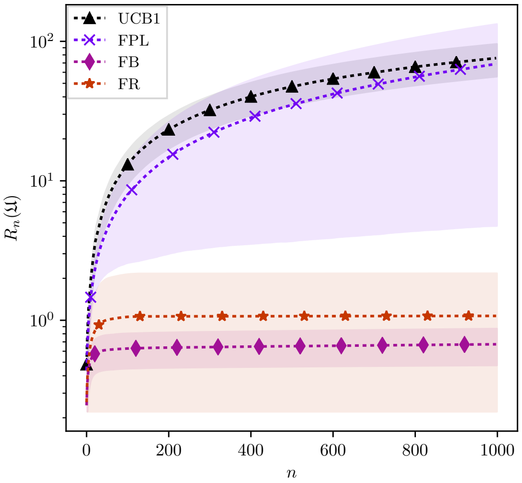

Figure 3: Expected pseudo-regret for the configuration C_6 with $M = 5$ targets.

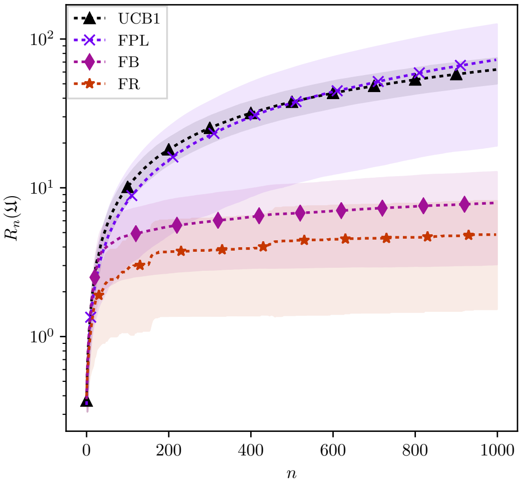

Figure 4: Expected pseudo-regret for the configuration C_7 with $M = 5$ targets.

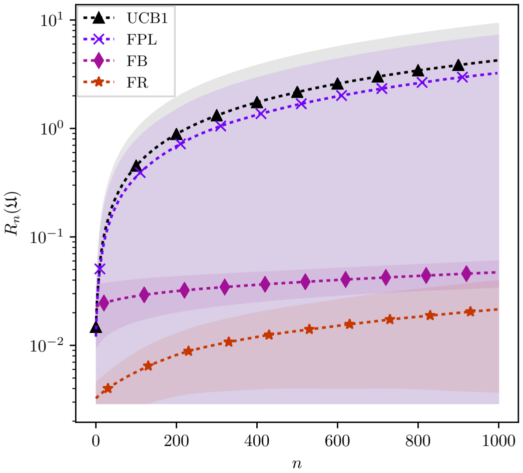

Figure 5: Expected pseudo-regret for the configuration C_5 with $M = 10$ targets.

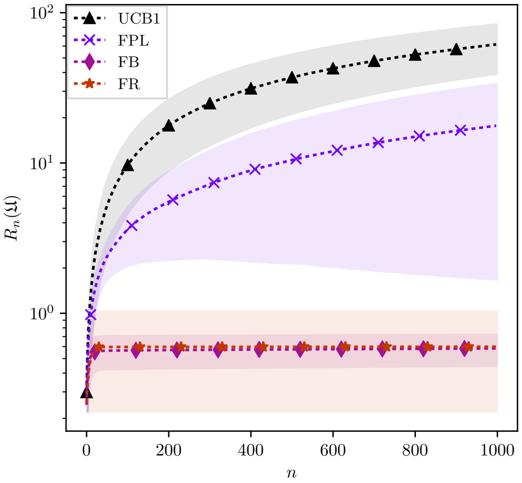

Figure 6: Expected pseudo-regret for the configuration C_6 with $M = 10$ targets.

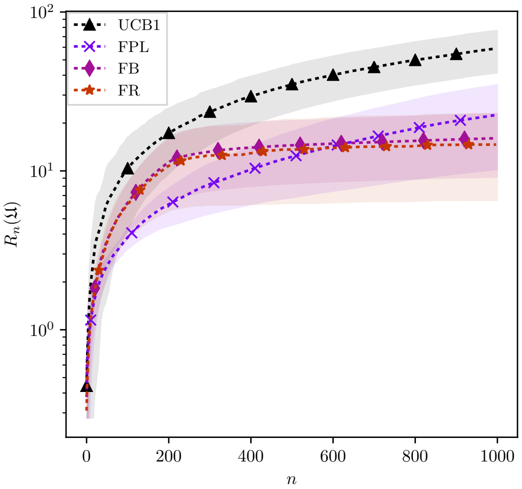

Figure 7: Expected pseudo-regret for the configuration C_7 with $M = 10$ targets.

References

An, B., Brown, M., Vorobeychik, Y., and Tambe, M. (2013). Security games with surveillance cost and optimal timing of attack execution. In *AAMAS*, pages 223–230.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *MACH LEARN*, 47(2):235–256.

Balcan, M., Blum, A., Haghtalab, N., and Procaccia, A. D. (2015). Commitment without regrets: Online learning in Stackelberg security games. In *EC*, pages 61–78.

Basilico, N., De Nittis, G., and Gatti, N. (2017). Adversarial patrolling with spatially uncertain alarm signals. *ART INT*, 246:220–257.

Blum, A., Haghtalab, N., and Procaccia, A. D. (2015). Learning to play Stackelberg security games. Technical report, Carnegie Mellon University, Computer Science Department.

Bubeck, S., Cesa-Bianchi, N., et al. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Conitzer, V. and Sandholm, T. (2006). Computing the optimal strategy to commit to. In *EC*, pages 82–90.

Conitzer, V. and Sandholm, T. (2007). Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *MACH LEARN*, 67(1-2):23–43.

Fang, F., Stone, P., and Tambe, M. (2015). When security games go green: designing defender strategies to prevent poaching and illegal fishing. In *IJCAI*, pages 2589–2595.

Ford, B., Kar, D., Delle Fave, F. M., Yang, R., and Tambe, M. (2014). PAWS: adaptive game-theoretic patrolling for wildlife protection. In *AAMAS*, pages 1641–1642.

Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT Press.

Kalai, A. and Vempala, S. (2005). Efficient algorithms for online decision problems. *J COMPUT SYST SCI*, 71(3):291–307.

Klíma, R., Kiekintveld, C., and Lisỳ, V. (2014). Online learning methods for border patrol resource allocation. In *GameSec*, pages 340–349.

McDiarmid, C. (1989). On the method of bounded differences. *LOND MATH S*, 141(1):148–188.

McFadden, D. L. (1984). Econometric analysis of qualitative response models. *Handbook of econometrics*, 2:1395–1457.

Nguyen, T., Yang, R., Azaria, A., Kraus, S., and Tambe, M. (2013). Analyzing the effectiveness of adversary modeling in security games. In *AAAI*, pages 718–724.

Paruchuri, P., Pearce, J. P., Marecki, J., Tambe, M., Ordóñez, F., and Kraus, S. (2008). Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *AAMAS*, pages 895–902.

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

Pita, J., Tambe, M., Kiekintveld, C., Cullen, S., and Steigerwald, E. (2011). Guards: game theoretic security allocation on a national scale. In *AAMAS*, pages 37–44.

Qian, Y., Haskell, W. B., Jiang, A. X., and Tambe, M. (2014). Online planning for optimal protector strategies in resource conservation games. In *AAMAS*, pages 733–740.

Qian, Y., Zhang, C., Krishnamachari, B., and Tambe, M. (2016). Restless poachers: Handling exploration-exploitation tradeoffs in security domains. In *AAMAS*, pages 123–131.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Tsai, J., Rathi, S., Kiekintveld, C., Ordóñez, F., and Tambe, M. (2009). IRIS-A tool for strategic security allocation in transportation networks. In *AAMAS*, pages 1327–1334.

Tuyls, K. and Weiss, G. (2012). Multiagent learning: Basics, challenges, and prospects. *AI MAG*, 33(3):41.

Von Stackelberg, H. (1934). *Marktform und gleichgewicht*. J. Springer.

Xu, H., Tran-Thanh, L., and Jennings, N. R. (2016). Playing repeated security games with no prior knowledge. In *AAMAS*, pages 104–112.

Yang, R., Ford, B., Tambe, M., and Lemieux, A. (2014). Adaptive resource allocation for wildlife protection against illegal poachers. In *AAMAS*, pages 453–460.

Yang, R., Kiekintveld, C., Ordóñez, F., Tambe, M., and John, R. (2011). Improving resource allocation strategy against human adversaries in security games. In *IJCAI*, pages 458–464.