Markov Security Games: Learning in Spatial Security Problems

Richard Klima[1], **Karl Tuyls**[1], **Frans Oliehoek**[1,2]
[1] University of Liverpool
[2] University of Amsterdam
{richard.klima, k.tuyls, frans.oliehoek}@liverpool.ac.uk

Abstract

In this paper we present a preliminary investigation of modelling spatial aspects of security games within the context of Markov games. Reinforcement learning is a powerful tool for adaptation in unknown environments, however the basic single-agent RL algorithms are unfit to be applied in adversarial scenarios. Therefore, we profit from Adversarial Multi-Armed Bandit (AMAB) methods which are designed for such situations. Based on temporal difference methods we derive two new multi-agent algorithms using AMAB methods for spatial two-player non-cooperative security games.

1 Introduction

Security games have gained a lot of attention in recent years due to their successful application on real-world security threats. Examples include the ARMOR system for airport security [16], the IRIS tool (scheduling Federal Air Marshals) [21], and the PROTECT system for scheduling Coast Guard [18]. Additionally some work has focused on Green Security Games for poaching problems [5]. Some of these security games however, do not consider space or time, i.e. the time it takes the defender to travel to the target node, as part of the model. Proposed strategies need to deal with delayed rewards in dynamic scenario, where the agents move on a map and have different levels of knowledge about the environment, which makes finding optimal strategies very demanding.

The security game framework is defined as a non-cooperative 2-player general-sum game. We call the two players with contradictory goals *defender* and *attacker* based on the model of the Stackelberg Security Game (SSG), widely used in security games. In our preliminary study we model the spatial security game to be played on a 2D graph (grid), which represents a map on which the players move and interact with each other. Solving the game by finding equilibria strategies is often not feasible due to limited knowledge about the model or computational intractability. A second option is to learn an optimal strategy from the interaction with the opponent in the environment. Therefore, in this paper we focus on reinforcement learning methods to approach spatial security games. To model the spatial security game effectively we use the concept of Markov Games (MG), which combines Markov Decision Processes (MDP) with repeated games, well described in [12]. In this paper we introduce an informal class of spatial security games which are a class of Markov Games in which two players, the attacker and defender, take moves in a physical environment represented as a grid, which we call the *Markov Security Game* (MSG). Choosing an optimal strategy (actions) for the defender at each step on the map is a decision-making process where we want to maximize the reward, therefore we can think of these actions as arms of multi-armed bandits with unknown distributions. We can argue that the attacker in security games is often adaptive adversary and thus it is convenient to focus on the Adversarial Multi-Armed Bandit framework, a generalisation of standard multi-armed bandit problems, which has no statistical assumptions on reward generation [1].

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Q-learning has been shown to perform well in a single agent setting, and there have been many efforts to extend this Bellman style reinforcement learning techniques to multi-agent settings [22]. Such approaches have been very successful in zero-sum repeated games, or team repeated games, but less successful in general-sum stochastic games [19]. We present a new approach to using reinforcement learning methods in combination with multi-armed bandit problem techniques to address Markov Games more effectively, applied to the security domain.

The paper is organised as follows: firstly we discuss related work in Section 2, after which we introduce the Spatial (Markov) Security Game model and define some of the key concepts in Section 3. In Section 4 we present two new algorithms and in Section 5 we evaluate both algorithms in the security domain. Finally, we conclude our work and point out several directions for future work.

2 Related work

We base our work on two areas: (i) temporal difference learning methods and (ii) adversarial multi-armed bandit methods. One of the fundamental temporal difference learning algorithms - Q-learning was presented by Watkins in 1989 [24], where he described a significant connection between reinforcement learning and Markov Decision Processes (MDP). Learning with delayed rewards appears in many real world problems where the reward is not immediately known. This is also the case in spatial security games where the movement on the map delays the reward which is obtained in time of apprehending the attacker or reaching the target node. Another temporal-difference algorithm is SARSA (state-action-reward-state-action) presented in [17]. SARSA is on-policy meaning we use the policy for the Q-value update. A standard framework of MDP assumes a single adaptive agent who operates in a stationary environment defined by a probabilistic transition function. Therefore any other agents in the environment must be thought of as part of the environment from the single agent point of view. Consequently, for multi-agent systems we need to come up with more sophisticated algorithms.

There have been several works presenting new algorithms based on Q-learning for multi-agent systems. Littman, 1994 [12] proposed minimax-Q, which substitutes the *max* operator in standard Q-learning update by the *minimax* operator as known in game theory, which can be solved by linear programming. The minimax strategy is the optimal strategy for non-cooperative zero-sum game. This algorithm has some weaknesses; (i) the necessity of using linear programming in each step demands high computational complexity, (ii) many domains (e.g. security games) require general-sum (non-zero-sum) assumption on rewards, therefore this algorithm cannot be safely used in such areas. In [7] the authors present the Nash Q-learning algorithm for general-sum stochastic games which is an intuitive next step from the minimax-Q algorithm, moving from zero-sum to general-sum games. As the name of the algorithm suggests the concept of Nash equilibrium (NE) is used, which is a baseline solution concept in general-sum games. NE is used in the update function of Q-learning replacing the *max* operator. Only in case of both players selecting the same NE the proposed algorithm is proven to converge. Nevertheless this algorithm faces the selection problem of NE (non-uniqueness property of NE), where in case of existence of multiple NE the algorithm might not converge. One of the proposed solutions to tackle the selection NE problem is using correlated equilibrium, however, that is not possible against an adversarial opponent (e.g. security games). Another intuitive step from using the Nash equilibrium concept in the Q-learning algorithm is to use the concept of Strong Stackelberg Equilibrium (SSG) as known in Stackelberg Games and has recently been widely used in Stackelberg Security Games [8]. Q-learning combined with SSG was presented in [10], where the authors discuss the asymmetric learning model with *leader* and *follower* as known in Stackelberg games. By using Stackelberg equilibrium they overcome the selection problem of Nash equilibrium because the Stackelberg equilibrium is unique and thus guarantees stronger convergence properties. The asymmetric learning model is a relaxation of the symmetric model discussed above. However in their model both agents need to accept their roles as leader and follower and keep a copy of the opponent's Q function, which is computationally demanding. In [14] the authors present and compare two strategies - *Bully* and *Godfather* which are based on Stackelberg leader and follower concepts. They compare combinations of these strategies played against each other in some well-known games. There has been more work that takes similar approaches, in which the authors modify Nash Q-learning by using the Stackelberg equilibrium concept [11]. In particular, an algorithm is proposed that chooses between Nash and Stackelberg equilibria based on dominance. Another approach is the friend-or-foe Q-learning (FFQ) algorithm presented in [13]. In case of coordination or adversarial

equilibria FFQ converges to Nash equilibrium. The algorithm requires additional information before the game starts about the relation of the players; in case of 'friend' the algorithm uses a method for cooperative learning and in case of 'foe' the algorithm uses zero-sum learning method. Thus, in cooperative setting the algorithm uses classic max update for Q-values and in adversarial setting the algorithm uses minimax update for Q-values. In security games we face the general-sum property, thus this algorithm cannot be used effectively here. An important temporal-difference style algorithm was presented in [23] called Expected SARSA, based on standard on-policy SARSA. The proposed algorithm reduces the increase in variance compared to standard SARSA. Expected SARSA bases its Q-value update on used-policy expected value $E\{Q(s_{t+1}, a_{t+1})\}$ rather than on $Q(s_{t+1}, a_{t+1})$. This new update rule seems promising in the domain of spatial security games where the policy is a mixed strategy over possible actions, thus the expected value update better reflects the goodness of the state.

In this paper we discuss the use of Adversarial Multi-Armed Bandit (AMAB) methods for temporal difference learning in Markov stochastic games. The AMAB framework was used for learning in some security games, e.g. Border Patrol [9].

Another efficient family of methods deployed to Security Games using random sampling is based on Monte Carlo Tree Search [15]. In [3] the authors use Monte Carlo Tree Search in Markov Games using Multi-Armed Bandit algorithms for the selection policy. However, in this paper we focus on temporal difference methods, which are better suitable for learning the strategy online.

3 Spatial Security Game model

The standard security game model is based on concept of Stackelberg game, deriving Stackelberg Security Games [8]. The main difference to normal form games is a distinction between the players; *leader* (defender) moves first and *follower* (attacker) observes defender strategy (to some extent; in our model the follower knows only about previously visited nodes by the leader) and acts upon that. The Spatial (Markov) Security Game (MSG) is a game model consisting of a grid game which represents a map on which 2 non-cooperative players act. We propose a realistic 2D model which captures the basic scenario of several security domains and propose effective defender strategy against adaptive adversary. Each node on the map (see Figure 1) is a state s, defined by its coordinates and each edge is an action a (up, down, left or right) from each node following the MDP framework notations. We test and compare several methods based on temporal difference (TD) methods and adversarial multi-armed bandit methods. For the opponent we assume intelligent adversarial attacker based on fictitious play.

3.1 Markov games

Markov Games (MG) generalize Markov decision processes and repeated games. MG is a stochastic game model with multiple agents moving in environment defined by states, actions and rewards obtained in each state. The concept of Markov games is described in [25] chapter 14.3.1. A Markov Game is defined as a tuple $(n, S, A_{1,\ldots,n}, R_{1,\ldots,n}, T)$ where n is number of agents in the system, S is a finite set of system states, A_k is the action set of agent k, $R_k : S \times A_1 \times ... \times A_n \times S \to \mathbb{R}$ is the reward function of agent k and $T : S \times A_1 \times ... \times A_n \times S \to \mu(S)$ is the transition function.

3.2 Temporal difference methods

A standard temporal difference off-policy learning method is Q-learning presented in [24]. We follow the description from [20].

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{1}$$

This function directly approximate the optimal action-value function which is independent to the followed policy (e.g. ϵ-greedy).

A modification of standard on-policy SARSA was proposed in [23] called Expected SARSA, where the Q-value update function is defined:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - Q(s_t, a_t)] \tag{2}$$

value function V is defined as $V(s_{t+1}) = \sum_a \pi(s_{t+1}, a)Q(s_{t+1}, a)$ for π to be a chosen policy. For cases where the policy π is greedy, Expected SARSA can be seen as a generalization of Q-learning; the value function $V(s)$ then simplifies to $V(s) = \max_a Q(s, a)$

3.3 Adversarial multi-armed bandit

Multi-armed bandit (MAB) is an important decision-making framework designed to approach well-known exploration vs. exploitation problem, where a player needs to decide between exploiting the best action so far or exploring new actions with potentially high rewards. Standard MAB assumes deterministic or stochastic generation of rewards, which we cannot assume in security game domain where the opponent is adaptive adversary. Therefore we focus on one of the strongest generalisation of bandit problem called Adversarial Multi-Armed Bandit (AMAB) problem where no stochastic assumptions on generation of rewards is made [1]. Algorithm for optimizing cumulative reward in AMAB environment is EXP3 *Exponential-weight algorithm for Exploration and Exploitation* proposed in [2]. We use a numerically more stable formulation introduced by [4]. Formally, a given action i is chosen with probability:

$$p(i) = \frac{1-\epsilon}{\sum\limits_{j \in K} e^{(S(j)-S(i))\frac{\epsilon}{K}}} + \frac{\epsilon}{K}, \tag{3}$$

where ϵ represents the amount of random exploration in the algorithm and K is the number of possible actions, thus $K = |A|$. Vector S represents sums of rewards for each action and is defined by

$$S_t(i) = S_{t-1}(i) + \frac{r_t}{p_t(i)} \tag{4}$$

so the vector S represents sums of rewards obtained in each round divided by probability of playing that action in each round. Thus, actions with higher rewards or rarely played actions are preferred.

4 TD learning using MAB method

We present a new approach to learning in Markov Security Games, where we use EXP3 algorithm as a policy for deciding on action selection using Q-values from temporal difference learning algorithms. Using algorithm EXP3 enables us to face effectively adversarial attacker, nevertheless standard EXP3 algorithm is not designed for MDP framework, thus we make use of temporal difference methods and combine EXP3 sum update with Q-values which makes using EXP3 in Markov Game possible. We propose two such methods, which we call EXP3-Q learning and Expected SARSA-EXP3 learning.

4.1 EXP3-Q

We propose algorithm EXP3-Q which is based on Q-learning update and EXP3 online learning algorithm. Using EXP3 enables us to learn effectively against adversarial opponent and Q-learning update conserves the spatial property of the environment defined by MDP.

The sums in EXP3 algorithm (Equation 3) are updated using the Q-values from Equation 1 modifying the standard EXP3 sum update (Equation 4) by

$$S_t(s, a) = S_{t-1}(s, a) + \frac{Q_t(s, a)}{p(s, a)} \tag{5}$$

4.2 Expected SARSA-EXP3

We use the algorithm Expected SARSA, where for the policy we use EXP3 algorithm stated in Equation 3. For the sum update in EXP3 algorithm we propose to use Q-values from Expected SARSA update function:

$$S_t(s, a) = S_{t-1}(s, a) + \frac{Q_t(s, a)}{p(s, a)} \tag{6}$$

where $Q_t(s, a)$ is a Q-value from Expected SARSA Q-value update function with value function using as policy the EXP3 algorithm:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - Q(s_t, a_t)] \tag{7}$$

where value function V is defined as $V(s_{t+1}) = \sum_a \pi^{EXP3}(s_{t+1}, a)Q(s_{t+1}, a)$ where π^{EXP3} is policy using EXP3 algorithm.

4.3 Attacker behaviour model

The aim of this paper is to propose effective algorithms to play against an intelligent adversarial attacker. We assume the attacker to adapt his strategy according to defender moves on the map and according to probability of successful attack in each particular zone. In our experiments we base the attacker behavior model on fictitious play [6], which can be seen as a realistic simple strategy for the attacker in security games. The attacker chooses the actions according to mixed strategy which is proportional to number of defender visits D and probability of successful attack T in each node. Vectors T and D are normalised to 1. Attacker strategy is then defined by probability of choosing each of the nodes i on the map by $p(i) = T_i - D_i$. In case of negative number we assume the attacker covers the target with 0 probability. The assumption of the attacker knowing all past defender moves comes from many security games where the attacker can observe the defender actions.

5 Experiments

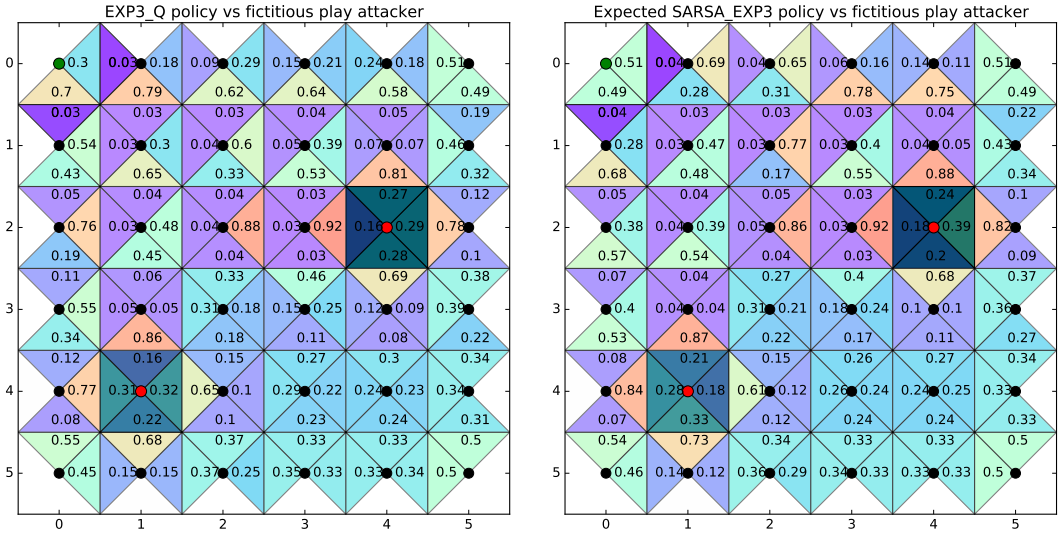
We test classic Q-learning and the proposed algorithms EXP3-Q and Expected SARSA-EXP3 in a grid game scenario. We assume a grid game of size 6x6 (see Figure 1), players have up to 4 actions depending on which node there are in - left, right, up and down (players are not allowed to get out of the map, i.e. in edging nodes). Each dot (black, red or green) is a node in the map. In our experiments an attacker decides at each step between two targets (nodes) based on fictitious play (see Section 4.3). Then the attacker attacks the chosen target with some probability of success given for each target. The targets are located at coordinates [4,1] and [2,4] depicted by shaded tiles and red dots and have probability of successful attack $T_1 = 0.25$ and $T_2 = 0.35$ respectively. The game stops when either (i) defender is in the same node as the attacker; attacker is apprehended or (ii) attacker successfully attacks the target. The defender starts at position [0,0] depicted by green dot. The attacker is assumed to be able to enter the map from any side therefore we do not reflect his transition on the map and only assume his presence in one of the two target nodes. The triangles contain probabilities of actions (mixed strategy vector over the actions - up, down, right, left) for each node for chosen strategy. In our experiments we use a fixed exploration $\epsilon = 0.1$ for all the policies.

We run experiments to tune the other parameters α and γ for each of the algorithms in Table 1. For each setting we run 10 000 games averaged over 500 runs. We provide number of average defender wins in the 10 000 games (this number is relative depending on setting of probabilities of successful attack T). The standard setting for the algorithms is $\alpha = 0.5$, $\gamma = 0.9$ and $\epsilon = 0.1$. In Table 1 we also provide 95% confidence intervals. The maximum number of wins for Q-learning was 696, for EXP3-Q learning 761 wins which is statistically-significantly better and for Expected SARSA-EXP3 learning we obtained 709 defender wins which is also better than Q-learning (in this case not statistical significant difference). We can see that EXP3-Q and Expected SARSA-EXP3 outperforms standard Q-learning. In Figure 1 we show the two proposed algorithms with tuned parameters; EXP3-Q with $\alpha = 0.5$ and $\gamma = 0.6$ and Expected SARSA-EXP3 with $\alpha = 0.5$ and $\gamma = 0.7$. These settings of the algorithms only demonstrate their good-enough behavior and we leave optimal tuning of the parameters (running all combinations of parameter settings) for future work. In Figure 1 we can see how the defender chooses his actions based on mixed strategy vectors in each node. Interesting node is [2,1], where the defender decides whether to cover target 1 at [4,1] or target 2 at [2,4]. For EXP3-Q he chooses with probability 0.45 to go down (target 1) and 0.48 to go right (target 2) however using Expected SARSA-EXP3 he chooses to go down with probability 0.54 (target 1) and with probability 0.39 to go right (target 2). Positive probabilities for going up or left come from exploration part of the algorithm.

Table 1: Defender wins for varying α and γ with 95% confidence interval

α	Q-learning	EXP3-Q	E-S-EXP3
0.1	424 ±24	456 ±15	449 ±10
0.2	471 ±21	**469** ±12	556 ±9
0.3	515 ±20	390 ±12	591 ±10
0.4	**558** ±18	358 ±12	623 ±11
0.5	551 ±18	311 ±12	642 ±11
0.6	536 ±16	264 ±11	653 ±12
0.7	519 ±15	223 ±11	687 ±12
0.8	487 ±13	198 ±11	685 ±12
0.9	430 ±12	163 ±11	**708** ±13

γ	Q-learning	EXP3-Q	E-S-EXP3
0.1	686 ±21	277 ±3	232 ±3
0.2	**696** ±20	460 ±4	372 ±3
0.3	692 ±19	622 ±6	524 ±5
0.4	676 ±19	706 ±8	624 ±6
0.5	678 ±19	759 ±10	677 ±7
0.6	658 ±19	**761** ±11	700 ±9
0.7	665 ±18	740 ±13	**709** ±11
0.8	608 ±19	630 ±13	671 ±11
0.9	548 ±17	307 ±12	645 ±12



(a) EXP3-Q with $\alpha = 0.5, \gamma = 0.6$

(b) Expected SARSA-EXP3 with $\alpha = 0.5, \gamma = 0.7$

Figure 1: EXP3-Q and Expected SARSA-EXP3 against fictitious attacker

6 Conclusion

In this work we proposed Markov Security Games as a model for spatial security problems. MSG are well suited to capture both the spatial component and delayed reward of the studied problems. This paper presents a preliminary study of analysing and describing the concept of MSG. We have proposed two new algorithms EXP3-Q and Expected SARSA-EXP3, which are based on temporal difference learning and on Adversarial Multi-Armed Bandit methods. As a first step we run experiments to show their sensitivity to different parameter settings and compared them to a standard Q-learner. We showed they outperform standard Q-learning, as expected because Q-learning is designed for a stationary environment. We visualised the two proposed algorithms in a grid game scenario representing MSG, in which we show the policy mixed strategies for choosing actions in each node. Our experiments are a first step in exploring MSGs, which shows promising performance of the two new algorithms. In future work we plan to do a more thorough analysis and evaluation of the algorithms w.r.t. spatial constraints and alternative baselines.

Acknowledgments

F.O. is funded by NWO Innovational Research Incentives Scheme Veni #639.021.336.

References

[1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casion: The adversarial multi-armed bandit problem. *NeuroCOLT2*, 1998.

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1), 2001.

[3] Branislav Bošanský, Viliam Lisý, Marc Lanctot, Jiří Čermák, and Mark H.M. Winands. Algorithms for computing strategies in two-player simultaneous move games. *Artificial Intelligence*, 237:1 – 40, 2016.

[4] Peter I. Cowling, Edward J. Powley, and Daniel Whitehouse. Information set Monte Carlo tree search. *IEEE Transaction on Computational Intelligence and AI in Games*, 2012.

[5] Fei Fang, Peter Stone, and Milind Tambe. When security games go green: designing defender strategies to prevent poaching and illegal fishing. *IJCAI*, 2015.

[6] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*. The MIT Press, 1998.

[7] Junling Hu and Michael P. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research 4*, 2003.

[8] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordonez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, 2009.

[9] Richard Klima, Viliam Lisy, and Christopher Kiekintveld. Combining online learning and equilibrium computation in security games. *GAMESEC*, 2015.

[10] Ville Kononen. Asymmetric multiagent reinforcement learning. *Web intelligence and Agent Systems*, 2:105–121, 2004.

[11] Julien Laumonier and Chaib-draa Brahim. Multiagent Q-learning: Preliminary study on dominance between the nash and stackelberg equilibriums. *AAAI-workshop*, 2005.

[12] Michael Littman. Markov games as a framework for multi-agent reinforcement learning. Technical report, Brown University, 1994.

[13] Michael Littman. Friend-or-foe Q-learning in general-sum games. *ICML*, 1, 2001.

[14] Michael L. Littman and Peter Stone. Leading best-response strategies in repeated games. In *In Seventeenth Annual International Joint Conference on Artificial Intelligence Workshop on Economic Agents, Models, and Mechanisms*, 2001.

[15] Janusz Marecki, Gerry Tesauro, and Richard Segal. Playing repeated stackelberg games with unknown opponents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pages 821–828, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[16] James Pita, Manish Jain, Craig Western, Christopher Portway, Milind Tambe, Fernando Ordonez, Sarit Kraus, and Praveen Parachuri. Depoloyed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport. In *AAMAS (Industry Track)*, 2008.

[17] G. A. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical report, 1994.

[18] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph Direnzo, Garrett Meyer, Craig W Baldwin, Ben J Maule, and Garrett R Meyer. PROTECT : A Deployed Game Theoretic System to Protect the Ports of the United States. *AAMAS*, 2012.

[19] Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365 – 377, 2007.

[20] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[21] Jason Tsai, Shyamsunder Rathi, Christopher Kiekintveld, Fernando Ordóñez, and Milind Tambe. IRIS - A tools for strategic security allocation in transportation networks. In *AAMAS (Industry Track)*, 2009.

[22] Karl Tuyls and Gerhard Weiss. Multiagent learning: Basics, challenges, and prospects. *AI Magazine*, 33(3):41–52, 2012.

[23] Harm van Seijen, Hado van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of expected SARSA. In *ADPRL 2009: Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 177–184, March 2009.

[24] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, May 1989.

[25] Marco Wiering and Martijn van Otterlo. *Reinforcement Learning: State-of-the-Art*. Springer, 2012.