Restless Poachers: Handling Exploration-Exploitation Tradeoffs in Security Domains

Yundi Qian, Chao Zhang, Bhaskar Krishnamachari, Milind Tambe
University of Southern California, Los Angeles, CA, 90089
{yundi.qian, zhan661, bkrishna, tambe}@usc.edu

ABSTRACT

The success of Stackelberg Security Games (SSGs) in counter-terrorism domains has inspired researchers' interest in applying game-theoretic models to other security domains with frequent interactions between defenders and attackers, e.g., wildlife protection. Previous research optimizes defenders' strategies by modeling this problem as a repeated Stackelberg game, capturing the special property in this domain — frequent interactions between defenders and attackers. However, this research fails to handle exploration-exploitation tradeoff in this domain caused by the fact that defenders only have knowledge of attack activities at targets they protect. This paper addresses this shortcoming and provides the following contributions: (i) We formulate the problem as a restless multi-armed bandit (RMAB) model to address this challenge. (ii) To use Whittle index policy to plan for patrol strategies in the RMAB, we provide two sufficient conditions for indexability and an algorithm to numerically evaluate indexability. (iii) Given indexability, we propose a binary search based algorithm to find Whittle index policy efficiently.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Security, Algorithms, Performance

Keywords

Exploration-exploitation tradeoff, Restless multi-armed bandit, Whittle index policy, POMDP

1. INTRODUCTION

Given the increasing need for security around the globe, optimizing the allocation of a limited number of security resources remains a crucial challenge. The successful applications of Stackelberg Security Games (SSGs) for security resource allocation in counter-terrorism domains [16] have inspired researchers' interest in applying game-theoretic models to new "frequent interaction" security domains with repeated interactions between defenders and attackers, e.g., wildlife protection domain. However, these two domains are different. In wildlife protection domain, attack (poaching) happens frequently so that it gives defenders (patrollers) the opportunity to learn attackers' (poachers') behavioral patterns from their previous actions and then to plan patrol strategies accordingly; while this learning opportunity does not arise in the counter-terrorism domain. Previous research [19, 5] has taken advantage of this opportunity and has modeled the wildlife protection domain as a repeated Stackelberg game. However, this work assumes that defenders have knowledge of all poaching activities throughout the wildlife protected area (we will discuss more about this related work in Section 7). Unfortunately, given vast geographic areas for wildlife protection, defenders do not have knowledge of poaching activities in areas they do not protect. Thus, defenders are faced with the exploration-exploitation tradeoff — whether to protect the targets that are already known to have a lot of poaching activities or to explore the targets that haven't been protected for a long time.

The exploration-exploitation tradeoff here is different from that in the non-Bayesian stochastic multi-armed bandit problem [2]. In stochastic multi-armed bandit problems, the rewards of every arm are random variables with a stationary unknown distribution. However, in our problem, patrol affects attack activities — more patrol is likely to decrease attack activities and less patrol is likely to increase attack activities. Thus, the random variable distribution is changing depending on player's choice — more selection (patrol) leads to lower reward (less attack activities) and less selection (patrol) leads to higher reward (more attack activities). On the other hand, adversarial multi-armed bandit problem [3] is also not an appropriate model for this domain. In adversarial multi-armed bandit problems, the reward can arbitrarily change while the attack activities in our problem are unlikely to change rapidly in a short period. This makes the adversarial multi-armed bandit model inappropriate for our domain.

In reality, how patrol affects attack activities would be reasonably assumed to follow a consistent pattern that can be learned from historical data (defenders' historical observations). We model this pattern as a Markov process and provide the following contributions in this paper. First, we formulate the problem into a restless multi-armed bandit (RMAB) model to handle the limited observability challenge — defenders do not have observations for arms they do not activate (targets they do not protect). Second, we propose an EM based learning algorithm to learn the RMAB model from defenders' historical observations. Third, we use the solution concept of Whittle index policy to solve the RMAB model to plan for defenders' patrol strategies. However, indexability is required for the existence of Whittle index, so we provide two sufficient conditions for indexability and an algorithm to numerically evaluate indexability. Fourth, we propose a binary search based algorithm to find the Whittle index policy efficiently.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016),*
J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.),
May 9–13, 2016, Singapore.

2. MODEL

2.1 Motivating Domains and their Properties

Our work is mainly motivated by the domain of wildlife protection such as protecting endangered animals and fish stocks [14, 19, 5]. Other motivating domains include police patrol to catch fare-evaders in a barrier-free transit system [21], border patrol [7, 8], etc. The model we will describe in this paper is based on the following assumptions about the nature of interactions between defenders and attackers in these domains. Except the frequent interactions between defenders (patrollers/police) and attackers (poachers/fare-evaders/smugglers), these domains share another two important properties: (i) patrol affects attacking activities (poaching/fare evasion/smuggling); (ii) limited/partial observability. We will next use the wildlife protection domain as the example to illustrate these two properties.

Poaching activity is a dynamic process affected by patrol. If patrollers patrol in a certain location frequently, it is very likely that the poachers poaching in this location will switch to other locations for poaching. On the other hand, if a location hasn't been patrolled for a long time, poachers may gradually notice that and switch to this location for poaching.

In the wildlife protection domain, both patrollers and poachers do not have perfect observation of their opponents' actions. This observation imperfection lies in two aspects: (i) limited observability — patrollers/poachers do not know what happens at locations they do not patrol/poach; (ii) partial observability — patrollers/poachers do not have perfect observation even at locations they patrol/poach — the location might be large (e.g., a $2km \times 2km$ area) so that it is possible that patrollers and poachers do not see each other even if they are at the same location.

These two properties make it extremely difficult for defenders to optimally plan their patrol strategies. For example, defenders may find a target with a large number of attack activities at the beginning so they may start to protect this target frequently. After a period of time, attack activities at this target may start to decrease due to the frequent patrol. At this time, defenders have to decide whether to keep protecting this target (exploitation) or to switch to other targets (exploration). However, defenders do not have knowledge of attack activities at other targets at that moment, which makes this decision making extremely difficult for defenders.

Fortunately, the frequent interactions between defenders and attackers make it possible for defenders to learn the effect of patrol on attackers from the historical data. With this learned effect, defenders are able to estimate attack activities at targets they do not protect. Based on this concept, we model these domains as a restless multi-armed bandit problem and use the solution concept of Whittle index policy to plan for defenders' strategies.

2.2 Formal Model

We now formalize the story in Section 2.1 into a mathematical model that can be formulated as a restless multi-armed bandit problem. There are n targets that are indexed by $\mathbb{N} \triangleq \{1, \ldots, n\}$. Defenders have k patrol resources that can be deployed to these n targets. At every round, defenders choose k targets to protect. After that, defenders will have an observation of the number of attack activities for targets they protect, and no information for targets they do not protect. The objective for defenders is to decide which k targets to protect at every round to catch as many attackers as possible.

Due to the partial observability on defenders' side — defenders' observation of attack activities is not perfect even for targets they protect, we introduce a hidden variable attack intensity, which represents the true degree of attack intensity at a certain target. Clearly, this hidden variable attack intensity cannot directly be observed by defenders. Instead, defenders' observation is a random variable conditioned on this hidden variable attack intensity, and the larger the attack intensity is, the more likely it is for defenders to observe more attack activities during their patrol.

We discretize the hidden variable attack intensity into n_s levels, denoted by $\mathbf{S} = \{0, 1, \ldots, n_s - 1\}$. Lower i represents lower attack intensity. For a certain target, its attack intensity transitions after every round. If this target is protected, attack intensity transitions according to a $n_s \times n_s$ transition matrix T^1; if this target is not protected, attack intensity transitions according to another $n_s \times n_s$ transition matrix T^0. The transition matrix represents how patrol affects attack intensity — T^1 tends to reduce attack intensity and T^0 tends to increase attack intensity. The randomness in the transition matrix models attackers' partial observability discussed in Section 2.1. Note that different targets may have different transition matrices because some targets may be more attractive to attackers (for example, some locations may have more animal resources in the wildlife protection domain) so that it is more difficult for attack intensity to go down and easier for attack intensity to go up.

We also discretize defenders' observations of attack activities into n_o levels, denoted by $\mathbf{O} = \{0, 1, \ldots, n_o - 1\}$. Lower i represents less attack activities defenders observe. Note that defenders will only have observation for targets they protect. A $n_s \times n_o$ observation matrix O determines how the observation depends on the hidden variable attack intensity. Generally, the larger the attack intensity is, the more likely it is for defenders to observe more attack activities during their patrol. Similar to transition matrices, different target may have different observation matrices.

While defenders get observations of attack activities during their patrol, they also receive rewards for that — arresting poachers/fare-evaders/smugglers bring benefit. Clearly, the reward defenders receive depends on their observation and we thus define the reward function $R(o), o \in \mathbf{O}$ — larger i leads to higher reward $R(i)$. For example, if $o = 0$ represents finding no attack activity and $o = 1$ represents finding attack activities, then $R(0) = 0$, $R(1) = 1$. Note that defenders only get rewards for targets they protect.

To summarize, for the targets defenders protect, defenders get an observation depending on its current attack intensity, get the reward associated with the observation, and then the attack intensity transitions according to T^1; for the targets defenders do not protect, defenders do not have any observation, get reward 0 and the attack intensity transitions according to T^0. Figure 1 demonstrates this process. In this model, the state discretization level n_s, observation discretization level n_o and reward function $R(o)$ are pre-specified by defenders; the transition matrices T^1 and T^0, observation matrix O and initial belief π can be learned from defenders' previous observations. We will briefly discuss the learning algorithm in Section 2.3. After those parameters are learned, this model is formulated into a restless multi-armed bandit model to plan for defenders' strategies.

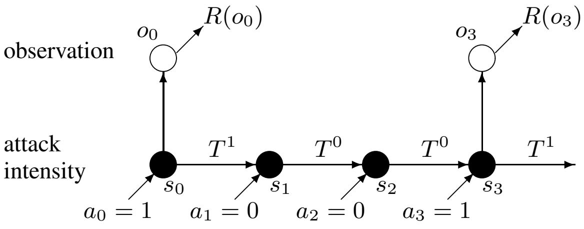


Figure 1: Model Illustration

2.3 Learning Model From Defenders' Previous Observations

Given defenders' action history $\{a_i\}$ and observation history $\{o_i\}$, our objective is to learn the transition matrices T^1 and T^0, observation matrix O and initial belief π. Due to the existence of hidden variables $\{s_i\}$, expectation-maximization (EM) algorithm is used for learning. We show the update steps here and the details are in the online appendix[1].

$$\pi_i^{(d+1)} = P(s_1 = i|x;\theta^d)$$

$$T_{ij}^{1(d+1)} = \frac{\sum_{t=1:a_t=1}^{T-1} P(s_t = i, s_{t+1} = j|x;\theta^d)}{\sum_{t=1:a_t=1}^{T-1} P(s_t = i|x;\theta^d)}$$

$$T_{ij}^{0(d+1)} = \frac{\sum_{t=1:a_t=0}^{T-1} P(s_t = i, s_{t+1} = j|x;\theta^d)}{\sum_{t=1:a_t=0}^{T-1} P(s_t = i|x;\theta^d)}$$

$$O_{ij}^{(d+1)} = \frac{\sum_{t=1:a_t=1}^{T} P(s_t = i|x;\theta^d)I(o_t = j)}{\sum_{t=1:a_t=1}^{T} P(s_t = i|x;\theta^d)}$$

where θ^d is the π, T^1, T^0, O last step and $P(s_t = i|x;\theta^d)$ and $P(s_t = i, s_{t+1} = j|x;\theta^d)$ are computed through forward-backward algorithm.

3. RESTLESS BANDIT FOR PLANNING

In this section, we will formulate the model discussed in Section 2.2 as a restless multi-armed bandit problem and plan defenders' strategies using the solution concept of Whittle index policy.

3.1 Restless Multi-armed Bandit Problems

In this section, we will briefly introduce the restless multi-armed bandit problems (RMABs) and their main solution concept Whittle index policy. In RMABs, each arm represents an independent Markov machine. At every round, the player chooses k out of n arms ($k < n$) to activate and receives the reward determined by the state of the activated arms. After that, the states of *all* arms will transition to new states according to certain Markov transition probabilities. The problem is called "restless" because the states of passive arms will also transition like active arms. The aim of the player is to maximize his cumulative reward by choosing which arms to activate at every round. It has shown by Papadimitriou and Tsitsiklis that it is PSPACE-hard to find the optimal strategy to general RMABs [13].

An index policy assigns an index to each state of each arm to measure how rewarding it is to activate an arm at a particular state. At every round, the index policy chooses to pick the k arms whose current states have the highest indices. Since the index of an arm only depends on the properties of this arm, index policy reduces an n-dimensional problem to n 1-dimensional problems so that the complexity is reduced from exponential with n to linear with n.

Whittle proposed a heuristic index policy for RMABs by considering the Lagrangian relaxation of the problem [18]. It has been shown that Whittle index policy is asymptotically optimal under certain conditions as k and n tend to ∞ with k/n fixed [17]. When k and n are finite, extensive empirical studies have also demonstrated the near-optimal performance of Whittle index policy [1, 6]. Whittle index measures how attractive it is to activate an arm based on the concept of subsidy for passivity. It gives the subsidy m to passive action (not activate) and the smallest m that would make passive action optimal for the current state is defined to be the Whittle index for this arm at this state. Whittle index policy chooses to activate the k arms with the highest Whittle indices. Intuitively, the larger the m is, the larger the gap is between active action (activate) and passive action, the more attractive it is for the player to activate this arm. Mathematically, denote $V_m(x; a = 0)$ ($V_m(x; a = 1)$) to be the maximum cumulative reward the player can achieve until the end if he takes passive (active) action at the first round at the state x with subsidy m. Whittle index $I(x)$ of state x is then defined to be:

$$I(x) \triangleq \inf_m \{m : V_m(x; a = 0) \geq V_m(x : a = 1)\}$$

However, Whittle index only exists and Whittle index policy can only be used when the problem satisfies a property known as indexability, which we define below. Define $\Phi(m)$ to be the set of states for which passive action is the optimal action given subsidy m:

$$\Phi(m) \triangleq \{x : V_m(x; a = 0) \geq V_m(x : a = 1)\}$$

DEFINITION 1. *An arm is indexable if $\Phi(m)$ monotonically increases from $\emptyset$ to the whole state space as m increases from $-\infty$ to $+\infty$. An RMAB is indexable if every arm is indexable.*

Intuitively, indexability requires that for a given state, its optimal action can never switch from passive action to active action with the increase of m. The indexability of an RMAB is often difficult to establish and computing Whittle index can be complex.

3.2 Restless Bandit Formulation

It is straightforward to formulation the model discussed in Section 2.2 into a restless multi-armed bandit problem. Every target is viewed as an arm and defenders choose k arms to activate (k targets to protect) at every round. Consider a single arm (target), it is associated with n_s (hidden) states, n_o observations, $n_s \times n_s$ transition matrices T^1 and T^0, $n_s \times n_o$ observation matrix O and reward function $R(o), o \in \mathbf{O}$ as is described in Section 2.2. For the arm defenders activate, defenders get an observation, get reward associated with the observation, and the state transitions according to T^1. Note that defenders' observation is not the state. Instead, it is a random variable conditioned on the state, and reveals some information about the state. For the arms defenders do not activate, defenders do not have any observation, get reward 0 and the state transitions according to T^0.

Since defenders can not directly observe the state, defenders maintain a belief b of the states for each target, based on which defenders make decisions. The belief is updated according to the Bayesian rules. The following equation shows the belief update when defenders protect this target ($a = 1$) and get observation o or defenders do not protect this target ($a = 0$).

$$b'(s') = \begin{cases} \eta \sum_{s \in \mathbf{S}} b(s) O_{so} T_{ss'}^1, & a = 1 \\ \sum_{s \in \mathbf{S}} b(s) T_{ss'}^0, & a = 0, \end{cases} \tag{1}$$

where η is the normalization factor. When defenders do not protect this target ($a = 0$), defenders do not have any observation, so their belief is updated according to the state transition rule; When defenders protect this target ($a = 1$), their belief is firstly updated according to their observation o ($b_{new}(s) = \eta b(s) O_{so}$ according to Bayes' rule), and then the new belief is then updated according to the state transition rule: $b'(s') = \sum_{s \in \mathbf{S}} b_{new}(s) T_{ss'}^1 = \sum_{s \in \mathbf{S}} \eta b(s) O_{so} T_{ss'}^1 = \eta \sum_{s \in \mathbf{S}} b(s) O_{so} T_{ss'}^1$

We now present the mathematical definition of Whittle index for our problem. Denote $V_m(b)$ to be the value function for belief state b with subsidy m; $V_m(b; a = 0)$ to be the value function for belief state b with subsidy m and defenders take passive action;

[1] http://teamcore.usc.edu/people/yundiqia/web%20page/papers/AAMAS2016Appendix.pdf

$V_m(b; a = 1)$ to be the value function for belief state b with subsidy m and defenders take active action. The following equations show these value functions:

$$
\begin{aligned}
V_m(b; a = 0) &= m + \beta V_m(b_{a=0}) \\
V_m(b; a = 1) &= \sum_{s \in \mathbf{S}} b(s) \sum_{o \in \mathbf{O}} O_{so} R(o) \\
&\quad + \beta \sum_{o \in \mathbf{O}} \sum_{s \in \mathbf{S}} b(s) O_{so} V_m(b^o_{a=1}) \\
V_m(b) &= \max\{V_m(b; a = 0), V_m(b; a = 1)\}
\end{aligned}
$$

When defenders take passive action, they get the immediate reward m and the β-discounted future reward — value function at new belief $b_{a=0}$, which is updated from b according to the case $a = 0$ in Equation 1. When defenders take active action, they get the expected immediate reward $\sum_{s \in \mathbf{S}} b(s) \sum_{o \in \mathbf{O}} O_{so} R(o)$ and the β-discounted future reward. The future reward is composed of different observation cases — $\sum_{s \in \mathbf{S}} b(s) O_{so}$ is defenders' probability to have observation o at belief state b, and $V_m(b^o_{a=1})$ is the value function at new belief $b^o_{a=1}$ that is updated from b according to the case $a = 1$ with observation o in Equation 1. The value function $V_m(b)$ is the maximum of $V_m(b; a = 0)$ and $V_m(b; a = 1)$. Whittle index $I(b)$ of belief state b is then defined to be :

$$I(b) \triangleq \inf_m \{m : V_m(b; a = 0) \geq V_m(b : a = 1)\}$$

The passive action set $\Phi(m)$, which is the set of belief states for which passive action is the optimal action given subsidy m is then defined to be:

$$\Phi(m) \triangleq \{b : V_m(b; a = 0) \geq V_m(b : a = 1)\}$$

3.3 Sufficient Conditions for Indexability

In this section, we provide two sufficient conditions for indexability when $n_o = 2$ and $n_s = 2$. Denote the transition matrices to be T^0 and T^1, observation matrix to be O. Clearly in our problem, $O_{11} > O_{01}$, $O_{00} > O_{10}$ (higher attack intensity leads to higher probability to see attack activities when patrolling); $T^1_{11} > T^1_{01}$, $T^1_{00} > T^1_{10}$; $T^0_{11} > T^0_{01}$, $T^0_{00} > T^0_{10}$ (positively correlated arms).

Define $\alpha \triangleq \max\{T^0_{11} - T^0_{01}, T^1_{11} - T^1_{01}\}$. Since it is a two-state problem with $\mathbf{S} = \{0, 1\}$, we use one variable x to represent the belief state: $x \triangleq b(s = 1)$, which is the probability of being in state 1.

Define $\Gamma_1(x) = xT^1_{11} + (1 - x)T^1_{01}$, which is the belief for the next round if the belief for the current round is x and the active action is taken. Similarly, $\Gamma_0(x) = xT^0_{11} + (1 - x)T^0_{01}$, which is the belief for the next round if the belief for the current round is x and the passive action is taken.

We present below two theorems demonstrating two sufficient conditions for indexability. The proof is in the online appendix.

THEOREM 1. *When $\beta \leq 0.5$, the process is indexable, i.e., for any belief x, if $V_m(x; a = 0) \geq V_m(x; a = 1)$, then $V_{m'}(x; a = 0) \geq V_{m'}(x; a = 1)$, $\forall m' \geq m$*

THEOREM 2. *When $\alpha\beta \leq 0.5$ and $\Gamma_1(1) \leq \Gamma_0(0)$, the process is indexable, i.e., for any belief x, if $V_m(x; a = 0) \geq V_m(x; a = 1)$, then $V_{m'}(x; a = 0) \geq V_{m'}(x; a = 1)$, $\forall m' \geq m$*

3.4 Numerical Evaluation of Indexability

For problems other than those that have been proved to be indexable in Section 3.3, we can numerically evaluate their indexability. We first provide the following proposition.

PROPOSITION 1. *If $m < R(0) - \beta \frac{R(n_o-1)-R(0)}{1-\beta}$, $\Phi(m) = \emptyset$; if $m > R(n_o - 1)$, $\Phi(m)$ is the whole belief state space.*

PROOF. If $m < R(0) - \beta \frac{R(n_o-1)-R(0)}{1-\beta}$, denote $V_m(b; a = 0) = m + \beta W_0$; $V_m(b; a = 1) = R(o) + \beta W_1$, where W_1 and W_0 represent the maximum future reward. Since $W_0 \leq \frac{R(n_o-1)}{1-\beta}$ (achieving reward $R(n_o - 1)$ at every round), $W_1 \geq \frac{R(0)}{1-\beta}$ (achieving reward $R(0)$ at every round), $R(o) \geq R(0)$, we have $V_m(b; a = 1) - V_m(b; a = 0) = R(o) - m + \beta(W_1 - W_0) \geq R(0) - m + \beta \frac{R(0)-R(n_o-1)}{1-\beta} > 0$. Thus, being active is always the optimal action for any state so that $\Phi(m) = \emptyset$.

If $m > R(n_o - 1)$, then the strategy of always being passive dominates other strategies so $\Phi(m)$ is the whole belief state space. □

Thus, we only need to determine whether the set $\Phi(m)$ monotonically increases for $m \subseteq [R(0) - \beta \frac{R(n_o-1)-R(0)}{1-\beta}, R(n_o - 1)]$. Numerically, we can discretize this limited m range and then evaluate if $\Phi(m)$ monotonically increases with the increase of discretized m. Given the subsidy m, $\Phi(m)$ can be determined by solving a special POMDP model whose conditional observation probability is dependent on start state and action. We will discuss the algorithm in detail in Section 4. This algorithm returns a set D which contains n_s-length vectors $d_1, d_2, \ldots, d_{|D|}$. Every vector d_i is associated with an optimal action e_i. Given the belief b, the optimal action is determined by $a^{opt} = e_i$, $i = \arg\max_j b^T d_j$. Thus, $\Phi(m) = \bigcup_{i:e_i=0}\{b : b^T d_i \geq b^T d_j, \forall j\}$.

Given $m_0 < m_1$, our aim is to check whether $\Phi(m_0) \subseteq \Phi(m_1)$. Use the superscript 0 or 1 for set D, vector d, action e to distinguish between the returned solutions with subsidy m_0 and m_1. The following mixed-integer linear program (MILP) can be used to determine whether $\Phi(m_0) \subseteq \Phi(m_1)$.

$$
\begin{aligned}
\min_{b, z^0, z^1, \xi^0, \xi^1} \quad & \sum_{i=1}^{|D^0|} z^0_i e^0_i - \sum_{i=1}^{|D^1|} z^1_i e^1_i \\
s.t. \quad & b_i \in [0, 1], \forall i \in \mathbf{S}, \qquad \sum_{i \in \mathbf{S}} b_i = 1 \\
& z^0_i \in \{0, 1\}, \forall i \in \{1, 2, \ldots, |D^0|\}, \qquad \sum_i z^0_i = 1 \\
& b^T d^0_i \leq \xi^0, \forall i \in \{1, 2, \ldots, |D^0|\} \\
& \xi^0 \leq b^T d^0_i + M(1 - z^0_i), \forall i \in \{1, 2, \ldots, |D^0|\} \\
& z^1_i \in \{0, 1\}, \forall i \in \{1, 2, \ldots, |D^1|\} \qquad \sum_i z^1_i = 1 \\
& b^T d^1_i \leq \xi^1, \forall i \in \{1, 2, \ldots, |D^1|\} \\
& \xi^1 \leq b^T d^1_i + M(1 - z^1_i), \forall i \in \{1, 2, \ldots, |D^1|\}
\end{aligned}
$$

If the result of the above MILP is 0 or 1, $\Phi(m_0) \subseteq \Phi(m_1)$. In the MILP, M is a given large number, b is the belief state, $z^{0/1}_i$ is a binary variable that indicates whether $b^T d^{0/1}_i \geq b^T d^{0/1}_j, \forall j$ (1 indicates yes and 0 indicates no), $\xi^{0/1}$ is an auxiliary variable that equals $\max_i b^T d^{0/1}_i$, $\sum_{i=1}^{|D^{0/1}|} z^{0/1}_i e^{0/1}_i$ is the optimal action for the problem with subsidy $m_{0/1}$. If the result of this MILP is 0 or 1, it means that there does not exist a belief b under which the optimal action for the problem with subsidy m_0 is passive (0) and the optimal action for the problem with subsidy m_1 is active (1). This means $\Phi(m_0) \subseteq \Phi(m_1)$. On the other hand, if the result of the above MILP is -1, $\Phi(m_0) \not\subseteq \Phi(m_1)$.

3.5 Computation of Whittle Index Policy

Given the indexability, Whittle index can be found by doing a binary search within the range $m \subseteq [R(0) - \beta\frac{R(n_o-1)-R(0)}{1-\beta}, R(n_o-1)]$. Given the upper bound ub and lower bound lb, the problem with middle point $\frac{lb+ub}{2}$ as passive subsidy is sent to the special POMDP solver to find the optimal action for the current belief. If the optimal action is active, then the Whittle index is greater than the middle point so $lb \leftarrow \frac{lb+ub}{2}$; or else $ub \leftarrow \frac{lb+ub}{2}$. This binary search algorithm can find Whittle index with arbitrary precision. Naively, we can compute the Whittle index policy by computing the ε-precision indices of all arms and then picking the k arms with the highest indices.

However, since we are actually only interested in which k arms have the highest Whittle index and we do not care what exactly their indices are, we can do better than this naive method, which is demonstrated in Algorithm 1.

Algorithm 1 Algorithm to Compute Whittle Index Policy

```
1: function FindWhittleIndexPolicy
2:     $lb \leftarrow R(0) - \beta\frac{R(n_o-1)-R(0)}{1-\beta}, ub \leftarrow R(n_o-1)$
3:     $A \leftarrow \emptyset, S \leftarrow \{1, 2, \ldots, n\}$
4:     while $|A| < k$ do
5:         $S_1 \leftarrow \emptyset, S_0 \leftarrow \emptyset$
6:         for $i \in S$ do
7:             $a^{opt} \leftarrow$ POMDPSolve($P_i$, $\frac{lb+ub}{2}$)
8:             if $a^{opt} = 1$ then
9:                 $S_1 \leftarrow S_1 \bigcup \{i\}$
10:            else
11:                $S_0 \leftarrow S_0 \bigcup \{i\}$
12:            end if
13:        end for
14:        if $|S_1| \leq k - |A|$ then
15:            $A \leftarrow A \bigcup S_1, S \leftarrow S - S_1$
16:            $ub \leftarrow \frac{lb+ub}{2}$
17:        else
18:            $S \leftarrow S - S_0$
19:            $lb \leftarrow \frac{lb+ub}{2}$
20:        end if
21:    end while
22:    return $A$
23: end function
```

In Algorithm 1, A is Whittle index policy to be returned and is set to be $\emptyset$ at the beginning. S is the set of arms that we have not known whether belong to A or not and is set to be the whole set of arms at the beginning. Before it finds top-k arms (the loop between Line 4 and Line 21), it tests all the arms in S about their optimal action with subsidy $\frac{ub+lb}{2}$. If the optimal action is 1, it means this arm's index is higher than $\frac{ub+lb}{2}$ and we add it to S_1; if the optimal action is 0, it means this arm's index is lower than $\frac{ub+lb}{2}$ and we add it to S_0 (Lines $6-13$). At this moment, we know that all arms in S_1 have higher indices than all arms in S_0. If there is enough space in A to include all arms in S_1, we add S_1 to A, remove them from S and set the upper bound to be $\frac{ub+lb}{2}$ because we already know that S_1 belongs to Whittle index policy set and all the rest arms have the index lower than $\frac{ub+lb}{2}$ (Lines $14-16$). If there is not enough space in A, we remove S_0 from S and set the lower bound to be $\frac{ub+lb}{2}$ because we already know that S_0 does not belong to Whittle index policy set and all the rest arms have the index higher than $\frac{ub+lb}{2}$ (Lines $17-19$)

4. COMPUTATION OF PASSIVE ACTION SET

In this section, we will discuss the algorithm to compute the passive action set $\Phi(m)$ with the subsidy m. This problem can be viewed as solving a special POMDP model whose conditional observation probability is dependent on start state and action while the conditional observation probability is dependent on end state and action in standard POMDPs. Figure 2 demonstrates the difference. The left figure represents special POMDPs and the right figure represents standard POMDPs. In both cases, the original state is s, the agent takes action a, and the state transitions to s' according to $P(s'|s,a)$. However, the observation o the agent get during this process is dependent on s and a in our special POMDPs; while it depends on s' and a in standard POMDPs.

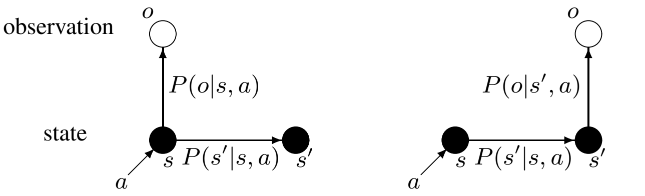


Figure 2: Special POMDPs vs Standard POMDPs

Despite this difference, the solution concept of value iteration algorithm in standard POMDPs can be used to solve our special POMDP formulations with appropriate modifications. We will discuss the special POMDP formulation for our problem in Section 4.1 and present the modified value iteration algorithm in Section 4.2.

4.1 Special POMDP Formulation

The special POMDP formulation for our problem is straightforward.

state space The state space is $\mathbf{S} = \{0, 1, \ldots, n_s - 1\}$.

action space The action space is $\mathbf{A} = \{0, 1\}$, where $a = 0$ represents passive action (do not protect) and $a = 1$ represents active action (protect).

observation space The observation space is $\mathbf{O} = \{-1, 0, 1, \ldots, n_o - 1\}$. It adds a "fake" observation $o = -1$ to represent no observation when taking action $a = 0$. It's called "fake" because defenders have probability 1 to observe $o = -1$ no matter what the state is when they take action $a = 0$, so this observation does not provide any information. When defenders take action $a = 1$, they may observe observations $\mathbf{O} \backslash \{-1\}$.

conditional transition probability The conditional transition probability $P(s'|s,a)$ is defined to be: $P(s' = j|s = i, a = 1) = T^1_{ij}$ and $P(s' = j|s = i, a = 0) = T^0_{ij}$.

conditional observation probability The conditional observation probability $P(o|s,a)$ is defined to be $P(o = -1|s, a = 0) = 1, \forall s \in \mathbf{S}$; $P(o = j|s = i, a = 1) = O_{ij}$. Note that the conditional observation probability here is dependent on the start state s and action a, while it depends on end state s' and action a in standard POMDP models. Intuitively, defenders' observation of attack activities today depends on the attack intensity today, not the transitioned attack intensity tomorrow.

reward function The reward function R is

$$R\left(s, s', a, o\right) = \begin{cases} 0, & a = 0, \\ R(o), & a = 1. \end{cases}$$

With the transition probability and observation probability, $R(s,a)$

can be computed. Note that this formulation is also slightly different due to the different definition of observation probability.

$$R(s,a) = \sum_{s' \in \mathbf{S}} P(s'|s,a) \sum_{o \in \mathbf{O}} P(o|s,a) R(s,s',a,o)$$

4.2 Value Iteration for Our Special POMDP

Different from standard POMDP formulation, the belief update in the special POMDP formulation is

$$b'(s') = \frac{\sum_{s \in \mathbf{S}} b(s) P(o|s,a) P(s'|s,a)}{P(o|b,a)} \tag{2}$$

where

$$P(o|b,a) = \sum_{s' \in \mathbf{S}} \sum_{s \in \mathbf{S}} b(s) P(o|s,a) P(s'|s,a) = \sum_{s \in \mathbf{S}} b(s) P(o|s,a)$$

Note that the belief update process is also consistent with that in Equation 1. Similar to standard POMDP formulation, we have the value function

$$V'(b) = \max_{a \in \mathbf{A}} \left(\sum_{s \in \mathbf{S}} b(s) R(s,a) + \beta \sum_{o \in \mathbf{O}} P(o|b,a) V(b_a^o) \right)$$

which can be broken up to simpler combinations of other value functions:

$$\begin{aligned} V'(b) &= \max_{a \in \mathbf{A}} V_a(b) \\ V_a(b) &= \sum_{o \in \mathbf{O}} V_a^o(b) \\ V_a^o(b) &= \frac{\sum_{s \in \mathbf{S}} b(s) R(s,a)}{|\mathbf{O}|} + \beta P(o|b,a) V(b_a^o) \end{aligned}$$

All the value functions can be represented as $V(b) = \max_{\alpha \in \mathbf{D}} b \cdot \alpha$ since the update process maintains this property, so we only need to update the set D when updating the value function. The set D is updated according to the following process:

$$\begin{aligned} D' &= \text{purge}\left(\bigcup_{a \in \mathbf{A}} D_a \right) \\ D_a &= \text{purge}\left(\bigoplus_{o \in \mathbf{O}} D_a^o \right) \\ D_a^o &= \text{purge}\left(\{\tau(\alpha,a,o) | \alpha \in D\} \right) \end{aligned}$$

where $\tau(\alpha,a,o)$ is the $|D|$-vector given by

$$\tau(\alpha,a,o)(s) = (1/|\mathbf{O}|) R(s,a) + \beta P(o|s,a) \sum_{s' \in \mathbf{S}} \alpha(s') P(s'|s,a)$$

and purge$(\cdot)$ takes a set of vectors and reduces it to its unique minimum form (remove redundant vectors that are dominated by other vectors in the set). $\bigoplus$ represents the cross sum of two sets of vectors: $A \bigoplus B = \{\alpha + \beta | \alpha \in A, \beta \in B\}$.

The update of D' and D_a is intuitive, so we briefly explain the update of D_a^o[2] here:

$$\begin{aligned} P(o|b,a) V(b_a^o) &= P(o|b,a) \max_{\alpha \in D} \sum_{s' \in \mathbf{S}} \alpha(s') P(s'|b,a,o) \\ &= P(o|b,a) \max_{\alpha \in D} \sum_{s' \in \mathbf{S}} \alpha(s') \frac{\sum_{s \in \mathbf{S}} b(s) P(o|s,a) P(s'|s,a)}{P(o|b,a)} \\ &= \max_{\alpha \in D} \sum_{s' \in \mathbf{S}} \alpha(s') \sum_{s \in \mathbf{S}} b(s) P(o|s,a) P(s'|s,a) \\ &= \max_{\alpha \in D} \sum_{s \in \mathbf{S}} b(s) \cdot \left(P(o|s,a) \sum_{s' \in \mathbf{S}} \alpha(s') P(s'|s,a) \right) \end{aligned}$$

Here, $P(s'|b,a,o)$ is the belief of state s' in the next round when the belief in the current round is b, the agent takes action a and get the observation o, which is the $b(s')$ in Equation 2.

5. PLANNING FROM POMDP VIEW

We have discussed in Section 4.1 that every single target can be modeled as a special POMDP model. Given that, we can combine these POMDP models at all targets to form a special POMDP model that describe the whole problem, and solving this special POMDP model leads to defenders' *exact* optimal strategy. Use the superscript i to denote target i. Generally, the POMDP model for the whole problem is the cross product of the single-target POMDP models at all targets with the constraint that only k targets are protected at every round.

state space The state space is $\mathbf{S} = \mathbf{S^1} \times \mathbf{S^2} \times \ldots \times \mathbf{S^n}$. Denote $s = (s^1, s^2, \ldots, s^n)$

action space The action space is $\mathbf{A} = \{(a^1, a^2, \ldots, a^n) | a^j \in \{0,1\}, \forall j \in \mathbb{N}, \sum_{j \in \mathbb{N}} a^j = k\}$, which represents that only k targets can be protected at a round. Denote $a = (a^1, a^2, \ldots, a^n)$

observation space The observation space is $\mathbf{O} = \mathbf{O^1} \times \mathbf{O^2} \times \ldots \times \mathbf{O^n}$. Denote $o = (o^1, o^2, \ldots, o^n)$

conditional transition probability The conditional transition probability is $P(s'|s,a) = \prod_{j \in \mathbb{N}} P^j(s'^j | s^j, a^j)$.

conditional observation probability The conditional observation probability is $P(o|s,a) = \prod_{j \in \mathbb{N}} P^j(o^j | s^j, a^j)$.

reward function The reward function is $R(s,s',a,o) = \sum_{j \in \mathbb{N}} R(s^j, s'^j, a^j, o^j)$

Naively, the modified value iteration algorithm discussed in Section 4.2 can be used to solve this special POMDP formulation. However, this POMDP formulation suffers from curse of dimensionality — the problem size increases exponentially with the number of targets. Thus, the computational cost of value iteration algorithm will soon become unaffordable as the problem size grows.

Silver and Veness [15] have proposed POMCP algorithm, which provides high quality solutions and is scalable to large POMDPs. The POMCP algorithm only requires a simulator of the problem so it also applies to our special POMDPs. At a high level, the POMCP algorithm is composed of two parts: (i) it uses a particle filter to maintain an approximation of the belief state; (ii) it draw state samples from the particle filter and then use MCTS to simulate what will happen next to find the best action. It uses a particle filter to approximate the belief state because it is even computationally impossible in many problems to update belief state due to the extreme large size of the state space. However, in our problem, the all-target POMDP model is the cross product of the single-target POMDP models at all targets. The single-state POMDP model is

[2] Actually the only difference of value iteration algorithm for the special POMDP formulation compared with that for the standard POMDP formulation is the different update of D_a^o.

small so that it is computationally inexpensive to maintain its belief state. Thus, we can easily sample the state s^i at target i from its belief state and then compose them together to get the state sample $s = (s^1, s^2, \ldots, s^n)$ for the all-target POMDP model.

The details of MCTS in POMDP are available in [15] so we omit it here. Although the POMCP algorithm shows better scalability than the exact POMDP algorithm, its scalability is also limited because the action space and observation space are also exponential with k in our problem. Consider the problem instance of $n = 10$, $k = 3$ and $n_o = 2$, the number of actions is $\binom{10}{3} = \frac{10*9*8}{1*2*3} = 120$ and the number of observations is $\binom{10}{3} * 2^3 = 960$. Since actions and observations are the branches in the MCTS, the tree size will soon become extremely large when planning more rounds ahead. This leads to two problems: (i) it will soon run out of memory when planning more rounds ahead; (ii) a huge number of state samples is needed to establish the convergence. Thus, the POMCP algorithm only applies to problem instances with small k. Our experimental evaluation shows that the POMCP algorithm is unable to plan 3 horizons forward (runs out of memory) for the problem instance of $n = 10$, $k = 3$ and $n_o = 2$. It means that for large problem instances, the POMCP algorithm is reduced to myopic policy (only look one round ahead when planning).

6. EXPERIMENTAL EVALUATION

In this section, we will firstly evaluate the Whittle Index Policy in Section 6.1 and then evaluate the RMAB model in Section 6.2. The performance is evaluated in terms of the cumulative reward received within the first 20 rounds with discounting factor $\beta = 0.9$. All results are averaged over 500 simulation runs.

6.1 Evaluation of Whittle Index Policy

We will compare the Whittle Index policy with four baseline algorithms:

Random: The defenders randomly choose k targets to protect at every round.

Myopic: The defenders choose k targets with the highest immediate reward to protect at every round.

Exact POMDP: The defenders uses the modified value iteration algorithm to solve the special POMDP problem discussed in Section 5 to plan for patrol strategies at every round. Note that it only works for small-scale problems and is the *exact* optimal patrol strategy defenders may take

POMCP: The defenders uses POMCP algorithm to solve the special POMDP problem discussed in Section 5 to plan for patrol strategies at every round.

The computation of Whittle Index policy and exact POMDP algorithm involve solving special POMDPs using the modified value iteration algorithm as is discussed in Section 4.2. We implement the modified value iteration algorithm by modifying the POMDP solver written by Anthony R. Cassandra[3]. The detailed algorithm we use for value iteration is the incremental pruning algorithm [4].

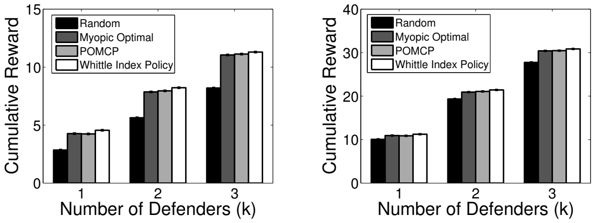
There are two parameters in the POMCP algorithm: the number of state samples and the depth of the tree, i.e., the number of rounds we look ahead when planning. With the increase of the number of state samples, the performance of the POMCP algorithm improves; while the runtime also increases at the same time. Thus, for a fair comparison, during our experiment, we choose the number of state samples so that its runtime is similar to that of Whittle index policy. For the depth of the tree, we choose the one with the largest cumulative reward.

Small Scale: Compare with Exact POMDP Algorithm We then evaluate these five planning algorithms in a small problem instance with $n = 2$, $k = 1$, $n_s = 2$ and $n_o = 2$. The result is shown in Table 1. From the table, we can see that our Whittle index policy and POMCP algorithm perform very close to the optimal Exact POMDP solution and are much better than the myopic optimal policy and random policy, demonstrating their high solution quality.

Table 1: Planning Algorithm Evaluation in Solution Quality for Small-scale Problem Instances

Random	Myopic Optimal	POMCP	Exact POMDP	Whittle Index
2.6534	3.1384	3.1694	3.1798	3.1740

Large Scale: We then evaluate our planning algorithms in a larger problem instance with $n = 10$. Figure 3(a) shows the solution quality comparison when $n_s = 2$ and $n_o = 2$. The x-axis shows the number of defenders (k) and the y-axis shows the cumulative reward. From this figure, we can see that Whittle index policy performs better than the POMCP algorithm and myopic optimal policy, and all of these three algorithms perform much better than the random policy. One thing to note is that the POMCP algorithm shows poor scalability with regard to k — it is unable to plan 3 horizons forward (runs out of memory) with $k = 3$. Figure 3(b) shows the solution quality comparison when $n_s = 3$ and $n_o = 3$, and demonstrates similar patterns as Figure 3(a).



(a) $n_s = 2, n_o = 2$ (b) $n_s = 3, n_o = 3$

Figure 3: Planning Algorithm Evaluation in Solution Quality for Large-scale Problem Instances

An Example When Myopic Policy Fails We can see from Figures 3(a) and 3(b) that the myopic policy performs only slightly worse compared with the Whittle index policy. Here we provide an example where the myopic policy performs significantly worse. Consider the case with 2 targets and 1 defender.

For target 0:

$$T^0 = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} T^1 = \begin{bmatrix} 0.99 & 0.01 \\ 0.1 & 0.9 \end{bmatrix} O = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

For target 1:

$$T^0 = \begin{bmatrix} 0.4 & 0.6 \\ 0.1 & 0.9 \end{bmatrix} \quad T^1 = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \quad O = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$$

Figure 4 shows the performance of different algorithms. In this case, the myopic policy performs similar to the random policy, and is much worse compared with Whittle Index policy.

Runtime Analysis of Whittle Index Policy: Figure 5 analyzes the runtime of Whittle index policy. The x-axis shows the number of targets (n) and the y-axis shows the average runtime. From the figure, we can see that the runtime increases linearly with the number of targets. This is because Whittle index policy reduces an n-dimensional problem to n 1-dimensional problems so that the complexity is linear with n. Another observation is that the number of defenders (k) does not affect the runtime a lot for a given n.

[3] http://pomdp.org/code/

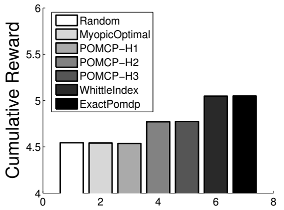


Figure 4: Example when Myopic Policy Fails

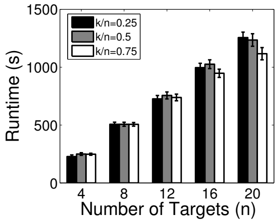


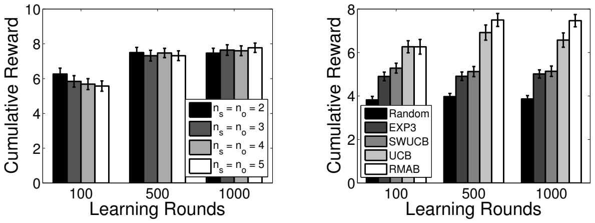
Figure 5: Runtime Analysis of Whittle Index Policy: $n_s = 2, n_o = 2$

6.2 Evaluation of RMAB Modeling

In this section, we will compare our RMAB model with the algorithms (UCB, SWUCB, EXP3) used in [7] with a group of simulated attackers. The performance is evaluated in terms of the cumulative reward received within the first 20 rounds after several rounds learning ($\beta = 0.9$).

Figure 6(a) demonstrates how the performance changes with different learning rounds and $n_s(n_o)$. It shows that when learning rounds is smaller (100), the model with $n_s = n_o = 2$ performs the best. This is because the model with higher $n_s(n_o)$ suffers overfitting with limited data at this time. When the data is relatively abundant (learning rounds = 1000), the model with higher $n_s(n_o)$ performs better. However, we noticed that the difference is not significantly large.

Figure 6(b) shows that comparison of our RMAB model with the Random/UCB/SWUCB/EXP3 algorithms. When learning rounds is smaller (100), our RMAB model performs similar to UCB algorithm, and is better than other algorithms. When learning rounds becomes larger, our RMAB model shows significant advantage over other algorithms.



(a) RMAB (b) $n_s = 2, n_o = 2$

Figure 6: Evaluation of RMAB Modeling

7. RELATED WORK

Previous research in Green Security Games [19, 5, 14] suffers from two major limitations. First, this research fails to capture the defender's *lack of observation of attacks* — in the real world, given a large area to patrol, the defender only has observations of attacks on the limited set of targets she patrolled in any given round. She does not have full knowledge of all of attackers' actions as assumed in [19, 5, 14], leading to an unaddressed exploration-exploitation tradeoff for defenders: informally, should the defender allocate resources to protect targets that have already been visited and have been observed to have suffered a lot of attacks or should she allocate resources to protect targets that have not visited for a long time and hence where there are no observations of attacks. Second, while significant work in security games has focused on uncertainty over attackers' observations of defender actions [20], the reverse problem has received little attention. Specifically, given frequent interactions with multiple attackers, the defender herself faces *observation uncertainty* in observing all of the attacker actions *even in the targets she does patrol*. Addressing this uncertainty in the defender's observation is important when estimating attacks on targets and addressing the exploration-exploitation tradeoff.

The limited observability property and exploration-exploitation tradeoff is also noticed by Klíma in the domain of border patrol where the border is large area [7, 8]. They model the problem as a stochastic/adversarial multi-armed bandit problem and use (sliding-window) UCB algorithm [2]/EXP3 algorithm [3] to plan for patrollers' strategies. However, the stochastic bandit formulation fails to model patrol's effect on attackers' actions while the adversarial bandit formulation fails to capture attackers' behavioral pattern.

Zhang et al. [22] focus on deterring crimes in urban areas by police patrol. They use Dynamic Bayesian Network (DBN) to model criminals' response to police patrol. They learn the DBN model from real-world crime data and then plan police patrol according to the DBN model. Their work does not deal with the exploration-exploitation tradeoff since victims would mostly report crimes to police so the police does not suffer from limited/partial observability issue in this domain.

There is a rich literature on indexability of restless multi-armed bandit problem. Glazebrook et al. [6] provide some indexable families of restless multi-armed bandit problems. Nino-Mora [11] propose PCL-indexability and GCL-indexability and show that they are sufficient conditions for indexability. Liu and Zhao [9] apply the concept of RMABs in dynamic multichannel access. In their model, every arm is a two-state Markov chain and the player only knows the state of the arm he chooses to activate. They prove the indexability of their problem and find the closed-form solution for the Whittle index. In [12], Ny et al. also consider the same class of RMABs but motivated by the application of UAV routing. This problem shares some similarity with our problem but our problem is more difficult in the following aspects: (i) we cannot directly observe the states (POMDP vs. MDP); (ii) different actions lead to different transition matrices in our model; (iii) we allow for more states and observations. A further extension to this work discusses the case with probing errors where the player's observation about the state might be incorrect [10]. This concept is similar to what we assume in our model, but the detailed settings are different.

8. CONCLUSION

This paper presents a new solution framework for security domains with frequent interactions and limited/partial observability. The motivating domains include wildlife protection domain, police patrol to catch fare-evaders, border patrol, etc. We model these domains as an RMAB to handle the limited/partial observability challenge. We provide an EM based learning algorithm to learn the RMAB model and use the solution concept of Whittle index policy to solve the RMAB model for planning optimal patrol strategies.

9. ACKNOWLEDGEMENT

This research is supported by the MURI grant W911NF-11-1-0332.

REFERENCES

[1] P. Ansell, K. D. Glazebrook, J. Niño-Mora, and M. O'Keeffe. Whittle's index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 2003.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 2002.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*. IEEE, 1995.

[4] A. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *UAI*, 1997.

[5] F. Fang, P. Stone, and M. Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *IJCAI*, 2015.

[6] K. Glazebrook, D. Ruiz-Hernandez, and C. Kirkbride. Some indexable families of restless bandit problems. *Advances in Applied Probability*, 2006.

[7] R. Klíma, C. Kiekintveld, and V. Lisỳ. Online learning methods for border patrol resource allocation. In *Decision and Game Theory for Security*. Springer, 2014.

[8] R. Klıma, V. Lisỳ, and C. Kiekintveld. Combining online learning and equilibrium computation in security games. 2015.

[9] K. Liu and Q. Zhao. Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. *Information Theory, IEEE Transactions on*, 2010.

[10] K. Liu, Q. Zhao, and B. Krishnamachari. Dynamic multichannel access with imperfect channel state detection. *Signal Processing, IEEE Transactions on*, 2010.

[11] J. Nino-Mora. Restless bandits, partial conservation laws and indexability. *Advances in Applied Probability*, 2001.

[12] J. L. Ny, M. Dahleh, and E. Feron. Multi-uav dynamic routing with partial observations using restless bandit allocation indices. In *American Control Conference*. IEEE, 2008.

[13] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 1999.

[14] Y. Qian, W. B. Haskell, A. X. Jiang, and M. Tambe. Online planning for optimal protector strategies in resource conservation games. In *AAMAS*, 2014.

[15] D. Silver and J. Veness. Monte-carlo planning in large pomdps. In *NIPS*, 2010.

[16] M. Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press, 2011.

[17] R. R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 1990.

[18] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 1988.

[19] R. Yang, B. Ford, M. Tambe, and A. Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *AAMAS*, 2014.

[20] Z. Yin, M. Jain, M. Tambe, and F. Ordonez. Risk-averse strategies for security games with execution and observational uncertainty. In *Conference on Artificial Intelligence (AAAI)*, 2011.

[21] Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*, 2012.

[22] C. Zhang, A. Sinha, and M. Tambe. Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *AAMAS*, 2015.