A Patrolling Game for Adversaries with Limited Observation Time

Ahmad Bilal Asghar   Stephen L. Smith

Abstract— **In this paper we consider a robot patrolling scenario on a weighted graph where an intruder can observe the patrolling path and use the information gained by observation to attack the graph's vertices. We pose the problem of finding a patrolling strategy as a multi-stage two player game. The patroller commits to a strategy that is unknown to the intruder. The intruder observes the patroller's actions for a finite amount of time to learn the patroller's strategy and then decides to either attack or renege based on its confidence in the learned strategy. We characterize the expected payoffs for the players and show that finding a k-factor approximation to the optimal patrolling strategy is NP-hard even when the patroller's strategy set is constrained to time homogeneous Markov chains. We propose a search algorithm to find a patrolling policy in such scenarios and illustrate the trade off between hard to learn and hard to attack strategies through simulations.**

I. Introduction

Patrolling locations of importance to protect against possible attacks arises in several security applications including ports, airports, and environmental resources [1]. Limited resources mean that perfect coverage cannot be provided to all locations and hence they have to be patrolled. An adversary can observe the patrolling path over time and use its knowledge about the path to its advantage. Deterministic paths are easier to exploit because they can be learned quickly and the attacker can plan the attack to avoid the patroller. This motivates randomizing of patrolling paths in order to make them less predictable [1], [2]. The focus of this paper is on strategies for generating random patrolling paths. The existing literature usually assumes that the adversary has knowledge of the patrolling strategy with the argument that the adversary can observe the patroller long enough to derive the strategy [3]. This assumption might not always be true in practice [4], as the adversary might not have enough time to perfectly learn the patrolling strategy. Therefore, we focus on the scenarios where the adversary has a limited time to observe the patrolling path to infer the patrolling strategy. It then decides to attack or renege based on its confidence in the learned strategy. Designing a patrolling strategy that takes longer to learn will increase the chances of the adversary reneging. However, the strategy should also minimize the chances of a successful attack in case the adversary attacks.

Related work: There is a substantial amount of work related to design of patrolling policies in robotics, control, and game theory. Random patrolling paths are advocated in [5] by showing that it is hard to find deterministic strategies that satisfy the surveillance criterion of visiting vertices proportional to their importance. Stochastic surveillance is considered in [6] so that the intruders cannot easily exploit the predictability of a deterministic path. Intelligent intruders in a perimeter patrol problem are considered in [7]. Different ways of randomizing paths are used in [8] to empirically show that randomizing the paths can decrease the probability of successful intrusions. In [9] strategic attacker is considered that plays two player zero sum game with the patroller.

Markov chains are often used to model random paths on a graph [5], [6], [10]. A common objective is to minimize mean first passage time of a Markov chain [10]. Different intruder models are considered in [11] to find Markov chains for patrolling in the presence of such intruders. We also restrict the search of the optimal policy to the set of Markov chains.

In game theory, security games are usually posed as Stackelberg games [2], [12]–[15]. In [16], a finite set of Markov chains is used to formulate the patrolling problem as a Bayesian Stackelberg Game where the patroller first picks a mixed optimal strategy and the adversary then picks the region to attack to maximize its payoff. In [17] the patrolling problem is studied on the graphs where traversing edges incurs different costs but the same amount of time and the attacker prefers to attack as soon as possible. One of the intruder models discussed in [11] and the problem setting considered by [3] is closest to ours, however the key difference is that the patrolling strategy is unknown to the attacker beforehand, and is learned through limited observations.

Limited observation time for the adversary leads to it having uncertainty about its own payoffs. In [18] security games where the defender is uncertain about the attacker's payoff is considered. In [4] a player 'Nature' decides whether the follower will observe the leader's actions or not and the leader does not know Nature's choice. We will also use Nature in our game model to pick the time allowed for observation. In [19] the attacker observes the defender for a limited time before attacking. These works, however, consider placing static resources on different locations rather than the problem of a patroller traversing a graph.

Contributions: We present a game model for the scenario where the adversary has a limited time to learn the patroller's strategy (Section II). We devise a strategy for a rational adversary and characterize the expected payoffs for the players (Section III). We set up the problem as a linearly constrained non-convex, non-smooth optimization problem and show that the objective function is locally Lipschitz so that direct search methods guarantee convergence to a Clarke Stationary point (Section IV). We also show that finding a k

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (abasghar@uwaterloo.ca; stephen.smith@uwaterloo.ca)

factor approximation to the optimal patrolling strategy is NP-hard. We propose a heuristic search algorithm (Section IV) and share the experimental results (Section V).

A. Background

We will use time homogeneous Markov chains to model random paths on graphs. The transition matrix P of a Markov chain gives the transition probabilities between its states. A Markov chain is said to be irreducible if it is possible to get to any state from any state. Since we are considering patrolling on a graph with a single patroller, we only consider irreducible Markov chains. We denote the stationary distribution of an n state Markov chain by $\pi \in \mathbb{R}^n$. For a vertex i of a graph, we denote the neighbors of that vertex by $\mathcal{N}(i)$. The element wise product of two matrices A and B is denoted as $A \circ B$. The expression $\mathbf{1}_b(a)$ evaluates to 1 if $a = b$ and 0 otherwise.

II. Problem Definition

A. Patrolling Scenario

The patrolling scenario involves an agent patrolling along the edges of a directed weighted graph $G = (V, E, W)$ where the vertices $V = \{1, 2, ..., n\}$ represent the locations to be monitored. The matrix $W = [w_{ij}]$ represents integer travel times on edges of the graph. Each vertex i of the graph has a value $\phi_i \geq 0$ that represents the value of that vertex. An intruder waits outside the environment observing the patrolling agent's path. It can attack any vertex of the graph and stay there for l time steps[1] to complete the attack. The patrolling agent cannot observe the intruder unless it visits a vertex i while it is being attacked, in which case the intruder is captured and incurs a penalty ψ whereas the patroller receives a reward ρ. Otherwise, the attack is successful and the intruder receives the reward ϕ_i.

Patrolling games are usually modeled as Stackelberg games where the defender chooses a strategy first, and the intruder being the follower knows the patroller's strategy and plays the best response. We consider the case where the intruder does not know the patrolling strategy beforehand. It has a fixed time budget to observe the patrolling path and it uses those observations to learn the patrolling agent's strategy. It then decides whether it will attack the environment or leave without attacking.

B. Game Model

We now model the above mentioned patrolling scenario as a multi-stage game, building on the game in [3] to include the learning aspect of the intruder. The game involves two players, the patrolling agent (defender) and the intruder (attacker). The players act simultaneously at each stage of the game. The attacker can observe the actions of the defender whereas the defender cannot observe the actions of the attacker. We also employ a move by nature to formally model the time allowed for the attacker to learn the defender's strategy. Nature selects a time T that denotes the maximum allowed time for the intruder before it decides whether it will attack or leave without attacking. The attacker knows the time T, however, the defender only knows the probability distribution $q : t \in \mathbb{Z}_+ \to [0, 1]$ representing the probability of nature selecting time t. The game starts at time zero.

Actions: The defender can go to a neighboring vertex of its current vertex i by taking the action *move*(j) where $j \in \mathcal{N}(i)$. Let H represent the set of all possible histories of vertices visited by the patroller, i.e. $h \in H$ is a sequence of vertices of some length. The available actions for the attacker are *obs*, *renege* or *attack-when*(j, h) for $i \in V$ and $h \in H$. Action *obs* corresponds to observing the current action of the defender for the sake of learning its strategy. Note that *obs* cannot be played after time T has elapsed from the start of the game. Also note that time steps are not analogous to stages of the game since the graph is weighted. Traversal of an edge by the defender can take multiple time steps but spans one stage of the game since both the attacker and defender play one action during that time. Playing *attack-when*(j, h) means that the attacker will wait until the defender has followed history h and then start the attack at vertex j. The attacker cannot take any other action after playing *attack-when*(j, h) and the game will conclude either in a successful attack or the capture of the intruder. Playing *renege* means that the attacker will leave without attacking the environment.

Outcomes: In case the intruder plays *renege*, the outcome of the game will be *no-attack*. If the attacker plays *attack-when*(j, h), it can either result in *successful-attack-j* if the defender does not visit j while it was being attacked or *capture* otherwise.

Payoffs: Both the player's payoffs are given in the following table. The attacker's payoff is the value gained from attacking or the penalty incurred if it is captured. Setting $\rho = 0$ represents a defender that does not give priority to *capture* over *no-attack* and only wants to stop a successful attack.

outcome	attacker's payoff	defender's payoff
no-attack	0	0
successful-attack-j	ϕ_j	$-\phi_j$
capture	$-\psi$	ρ

Strategies: The defender's strategy $\sigma_d(h)$ gives the probability with which the defender will move to a vertex that is a neighbor of the last visited vertex in $h \in H$. Since the defender cannot observe the actions of the attacker, its strategy is not a function of the attacker's actions. The attacker's strategy is a function from the patrolling history to the possible actions *obs*, *attack-when*(j, h) or *renege*.

Remark 1. *We assume that T is much greater than the time the attacker waits to attack after deciding where to attack. This models scenarios where the attacker can commit significant resources/time to planning an attack relative to the time scale on which the defender traverses the graph. In this case the information gained while waiting will be negligible compared to the information gained while learning.*

[1]We use the uniform length l for simplicity of presentation. The length of attack can be different for each vertex and the results of the paper will hold with a slight change in payoff equations.

III. Solution Approach

For the scenario where the attacker has perfect knowledge of defender's strategy beforehand, the authors in [3] show that the defender's strategy will be Markovian with some finite memory length. However, they argue this problem is in general intractable and thus formulate their algorithms for Markov chains with memory length one. Therefore, we also restrict defender's strategy to Markov chains with memory one. Moreover, since the defender receives no information about the attacker during the game unless the attacker is captured (at which point the game ends), we only consider stationary Markov chains. Thus, we represent the defender's strategy as the transition matrix $P = [p_{ij}]$ of a Markov chain, where p_{ij} gives the probability of the patrolling agent going to vertex j when it is in i. In the rest of the paper, optimal strategy for the defender will refer to the optimal Markov chain.

A. Attacker's Strategy

Assuming that the players are rational, the following proposition simplifies the strategy of the attacker.

Proposition 2. *Restricting the defender's strategy to be a time homogeneous Markov chain, an optimal strategy for the attacker is to play the action obs until time T and then to either play attack-when(j, i) for some $i, j \in V$ or renege.*

Proof. Actions *attack-when*(j, h_1) and *attack-when*(j, h_2) for $h_1, h_2 \in H$ such that the last vertex in h_1 and h_2 is the same vertex i, will give the attacker the same expected payoff, since the defender's strategy only depends on the current vertex occupied by the defender. Given an observation sequence o_t of duration $t \le T$, consider the attacker's optimal action given by $a(o_t)$. Suppose that the expected payoff for the attacker is maximized for some o_{t^*} where $t^* < T$. Then the attacker can still receive the same payoff by playing the action $a(o_t^*)$ after observing o_T because there is no cost to play *obs* and the expected payoff for action *attack-when*(j, i) is independent of the time the action is played. $\square$

Given that the defender is using the transition matrix P as its strategy, the attacker's expected payoff for the action *attack-when*(j, i) can be calculated as follows [3], [11]:

$$F_t(i,j) = \begin{cases} p_{ij}\mathbf{1}_1(w_{ij}), & t = 1 \\ \sum_{h \neq j} p_{ih} F_{t-w_{ih}}(h,j) + p_{ij}\mathbf{1}_t(w_{ij}), & t \ge 2 \end{cases}$$

$$s_{ij} = 1 - \sum_{t=1}^{l} F_t(i,j) \tag{1}$$

$$u_{ij} = \phi_j s_{ij} - \psi(1 - s_{ij}). \tag{2}$$

$F_t(i, j) = 0$ for non positive values of t. In the above set of equations, $F_t(i, j)$ represents the probability of the patroller visiting vertex j for the first time from vertex i in exactly t time steps, and s_{ij} gives the probability of the attack being successful when the attacker plays *attack-when*(j, i). The defender's expected payoff for this action will be

$$x_{ij} = -\phi_j s_{ij} + \rho(1 - s_{ij}). \tag{3}$$

The attacker, not knowing P, cannot calculate its expected payoffs. Therefore, it uses the observations to learn the defender's strategy and uses the learned strategy to estimate its expected payoffs. Let $\hat{P}$ denote the estimated Markov chain transition matrix after time T with the covariance $\texttt{Cov}(\hat{P})$. If u_{ij} was a linear function of P, then the attacker could use $\hat{u}_{ij}$ (obtained by using $\hat{p}_{ij}$ instead of p_{ij} in above expressions) to estimate the expected payoff. Since, u_{ij} is not linear, maximizing $\hat{u}_{ij}$ may not be optimal and the attacker will need to consider higher moments of the estimate. The defender needs the optimal attacker response to optimize its own strategy. Since the Fisher information matrix [20] can easily provide $\texttt{Cov}(\hat{P})$, we approximate the attacker's response by only considering $\texttt{Cov}(\hat{P})$ and ignore the higher moments. We leave the true optimization for a future work. The uncertainty in the estimated strategy can then be propagated to evaluate $\texttt{var}(\hat{u}_{ij})$. Note that $\texttt{Cov}(\hat{P})$ and $\texttt{var}(\hat{u}_{ij})$ are functions of T. Given this data, the problem is to decide whether to play *renege* or *attack-when*(j, i) for some i, j. The efficient frontier for this risk-reward trade-off can be represented as a special case of Markowitz portfolio theory [21] where the attacker has to pick only one action instead of a portfolio. For a parameter $\Lambda \ge 0$, let $(i^*, j^*) = \arg\max_{(i,j)}\{\hat{u}_{ij} - \Lambda \texttt{var}(\hat{u}_{ij})\}$. Then the attacker will play

$$\begin{matrix} \textit{attack-when}(j^*, i^*) & \text{if} \quad \hat{u}_{i^*j^*} - \Lambda \texttt{var}(\hat{u}_{i^*j^*}) > 0 \\ \textit{renege} & \text{otherwise.} \end{matrix} \tag{4}$$

Here Λ models how risk averse the attacker is, e.g. a large value of Λ models a risk averse intruder who will attack a vertex only if it is confident enough in its estimate of the expected payoff. We will assume that the defender does not know the value of Λ but has a probability distribution $g(\lambda)$ over the possible values of Λ.

B. Defender's Strategy

The defender's strategy should be hard to attack (minimize the worst case attacker's payoff) as well as hard to learn (make attackers renege due to high payoff variance). The following proposition formalizes the defender's payoff to design such a strategy.

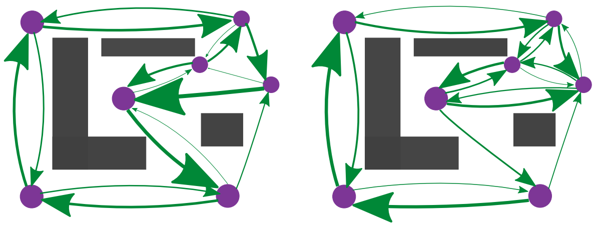

Fig. 1. The 'hard to attack' (left) and 'hard to learn and attack' (right) strategies in an environment with obstacles. The area of a vertex represents its reward. The thickness of an edge represents the probability of traversing that edge. Notice that the traversals in the strategy on the right are less deterministic as compared to that on the left.

Proposition 3. *If the defender's strategy is given by P and the attacker's strategy is defined by* (4) *and Proposition 2,* *then the expected payoff for the defender is given by*

$$\left(1-\sum_t \int_{\lambda=\max_{i,j}\frac{\hat{u}_{ij}}{var(\hat{u}_{ij})}}^{\infty} g(\lambda)d\lambda q(t)\right)x_{i^*j^*}. \quad (5)$$

Proof. Using (4), the attacker will renege if T and Λ are drawn such that $\max_{i,j}\{\hat{u}_{ij}-\Lambda\mathtt{var}(\hat{u}_{ij})\}\leq 0$. This implies $\Lambda \geq \hat{u}_{ij}/\mathtt{var}(\hat{u}_{ij}), \forall i,j$ and hence, the part inside the parentheses in (5) is the probability that the attacker will play *attack-when*(j^*,i^*) instead of *renege*. The defender's payoff is zero in case of no attack and, using Equation (3), $x_{i^*j^*}$ if the attacker plays *attack-when*(j^*,i^*). □

Figure 1 gives an example of a hard to attack strategy that maximizes $x_{\tilde{i},\tilde{j}}$ where $\tilde{i},\tilde{j}=\arg\max_{i,j}\{u_{ij}\}$ versus a hard to learn and attack strategy that maximizes an approximation of (5). Ideally, the defender should find P that maximizes its expected payoff (5), but the values of $\hat{u}_{ij}$, $\mathtt{var}(\hat{u}_{ij})$ and $x_{i^*j^*}$ depend on a realized path that the attacker observed to calculate these quantities. Hence, maximizing the said expression as it is, is not possible for the defender. We deal with this issue as follows.

Lower Bound on Expected Payoff: Although the defender does not know i^*, j^*, it can lower bound its expected payoff using $\min_{i,j}\{x_{ij}\}$ instead of $x_{i^*j^*}$ in Expression (5). This is equivalent to being prepared for the worst case scenario in case the intruder attacks. Note that if learning was not involved, and intruder knew the patrolling strategy P, then the attacker would maximize its own payoff resulting in the defender maximizing $x_{\tilde{i},\tilde{j}}$ where $\tilde{i},\tilde{j}=\arg\max_{i,j}\{u_{ij}\}$.

Attacker's Estimate: The attacker's estimate of the expected payoff $\hat{u}_{ij}$ is not known to the defender. However, since it knows P, it can calculate the actual value of the attacker's expected payoff u_{ij}. If the attacker is using a consistent estimator, the estimate $\hat{u}_{ij}$ converges to u_{ij}.

Variance in Estimate: Assuming the attacker is using an efficient estimator, the covariance in the attacker's estimate of P is given by the inverse of the Fisher information matrix of the Markov chain represented by $I_N(P)$, where N is the number of transitions observed. The average time taken during one transition of the Markov chain on a weighted graph is given by $\pi(P\circ W)\mathbf{e}$ where $\mathbf{e}$ is a column of ones [10] . Hence, for large N and T,

$$N\approx\frac{T}{\pi(P\circ W)\mathbf{e}}. \quad (6)$$

Therefore, theoretically, the defender can propagate the covariance using the expressions from [22] to find $\mathtt{var}(\hat{u}_{ij})$ as a function of T . However, the expression for u_{ij} includes iterative multiplications and summations of correlated variables and in practice this propagation is computationally expensive. Since the defender has access to P, it can use Monte Carlo simulations to estimate these variances. We will denote the variances calculated by the defender by $\mathtt{var}(u_{ij})$ although u_{ij} is not a random variable for the defender.

Now, we are in a position to write the optimization problem to be solved by the defender. Let

$$f(P)=-\left(1-\sum_t \int_{\lambda=\max_{i,j}\frac{u_{ij}}{\mathtt{var}(u_{ij})}}^{\infty} g(\lambda)d\lambda q(t)\right)\min_{i,j}x_{ij}, \quad (7)$$

then the optimization problem can be written as follows.

$$\begin{aligned} \underset{P}{\mathtt{minimize}} \quad & f(P) \\ \mathtt{subject\ to} \quad & \\ & \sum_j p_{ij}=1 \quad \text{for all } j \\ & 0\leq p_{ij}\leq 1 \quad \text{for all } i,j \\ & p_{ij}=0 \quad \text{for } (i,j)\notin E. \\ & P \quad \text{is irreducible} \end{aligned} \quad (8)$$

Note that we will relax the last constraint in our optimization and instead we verify that the final solution satisfies this constraint. This works well in practice because a Markov chain with more than one communicating class means that the chain cannot go from a state to all other states making $s_{ij}=1$ for all j for some value of i.

IV. Computing the Patrolling Strategy

The function $f(P)$ is a non convex function in general. This can be easily observed by calculating the Hessian of s_{ij} for some i,j. Moreover, the function is also non smooth in general because it picks the minimum element among x_{ij}. We now characterize the hardness of the problem.

Proposition 4. *Unless $\mathcal{P}=\mathcal{NP}$, for any $k\geq 1$, there does not exist a polynomial time k-factor approximation algorithm for Problem* (8).

Proof. We use a reduction from the instance $G=(V,E)$ of the HAMILTONIAN CYCLE to an instance of our problem with zero optimal value. Pass the graph G to the approximation version of Problem (8) with $w_{ij}=1,\forall\{i,j\}\in E$, $\phi=\mathbf{1}$, $\psi=\rho=0$ and $l=|V|$. Choose any distributions for $g(\lambda)$ and $q(t)$. A Hamiltonian cycle in G can be represented using a transition matrix P and it will ensure that the payoff of the attacker and the defender is zero which is optimal. If no Hamiltonian cycle exists in G, the defender cannot visit all the vertices within l time and the expected payoff for the defender will not be zero. □

A. Numerical Optimization

We will first show that our objective function is locally Lipschitz under mild assumptions on $g(\lambda)$ and $q(t)$.

Proposition 5. *If the variance $\mathtt{var}(u_{ij})$ is calculated by propagating the inverse of $I_N(P)$, and if the distributions $g(\lambda)$ and $q(t)$ for $t\in[T_{min},T_{max}]$ are bounded, then the objective function $f(P)$ is locally Lipschitz at each point P.*

Proof. The functions x_{ij} and u_{ij} are continuously differentiable with respect to $p_{xy},\forall x,y$, and hence, $\min x_{ij}$ is locally Lipschitz. The product of two bounded and locally Lipschitz functions is locally Lipschitz. The part inside the parentheses in Equation (7) is a probability and hence is bounded, and

Algorithm 1 PATROLLINGSTRATEGY

```
1: Pick a starting transition matrix $P_o$ and $\gamma_o$
2: for $k = 0, 1, 2, \ldots$ do
3:     if $\gamma_k < \gamma_t$ then
4:         stop
5:     $flag \leftarrow 0$
6:     for $d \in \mathcal{D}_k$ do
7:         $Q \leftarrow P_k + \gamma_k d$
8:         if $Q$ not feasible then
9:             $Q \leftarrow$ PROJECT($Q$)
10:        if $f(Q) < f(P_k) - \rho(\gamma_k)$ then
11:            $P_{k+1} \leftarrow Q$
12:            $\gamma_{k+1} \leftarrow \phi_k \gamma_k$
13:            $flag \leftarrow 1$
14:            break
15:    if $flag = 0$ then
16:        $\gamma_{k+1} \leftarrow \theta_k \gamma_k$
```

x_{ij} is bounded if ϕ and ρ are bounded. So proving that the probability of the attacker not reneging is locally Lipschitz will complete the proof. Using the definition of Fisher information matrix for a Markov chain along with the fact that π_i is a differentiable function of P [23], $u_{ij}/\mathtt{var}(\hat{u}_{ij})$ is differentiable[2] and $\max u_{ij}/\mathtt{var}(\hat{u}_{ij})$ is Lipschitz. It can be easily shown that $\int_{y(x)}^{c} g(\lambda)d\lambda$ and $\sum_{t=T_{\min}}^{T_{\max}} z(x,t)q(t)$ are locally Lipschitz for bounded $g(\lambda)$ and $q(t)$ if $y(x)$ and $z(x,t)$ are locally Lipschitz. $\square$

Given that the objective function is non-convex, non-smooth and locally Lipschitz, and the constraint set is convex, Mesh Adaptive Direct Search method (MADS) can be used to find a Clarke Stationary point [24]. Thus, we can apply standard gradient-free optimization techniques [25] to compute a patrolling strategy. Moreover, we also exploit some structure of the problem to improve the performance.

At iteration k of Algorithm 1, a set of search directions $\mathcal{D}_k$ is chosen and the function is evaluated at a given step length γ_k along these directions. As soon as a better point is found, it is chosen as the current point and the method proceeds to the next iteration. The step length is decreased or increased depending on whether a better point is found. The sufficient decrease function $\rho(\gamma_k)$ is $M\gamma_k^{3/2}$ where M is a constant, as suggested by [25] for the algorithm to converge.

Direction set: For each row i of the transition matrix, the constraints define a simplex in $\mathbb{R}^{|\mathcal{N}(i)|}$ dimensions. So, the set $\mathcal{D}_k$ includes $|\mathcal{N}(i)| - 1$ orthonormal directions on the simplex for each row i. In our implementation, we also include some random directions in $\mathcal{D}_k$.

Projection onto the feasible set: The function PROJECT in line 9 of Algorithm 1 projects the matrix Q onto the convex set defined by the first three constraints of Problem 8. Since each row i of matrix P lies in a $|\mathcal{N}(i)|$ dimensional simplex, we can use the algorithm from [26] to project each row of Q onto the desired simplex. This method employs simple vector operations and is therefore more efficient as compared to projection using convex optimization.

[2]If $\mathtt{var}(\hat{u}_{ij}) = 0$ for $u_{ij} > 0$, $f(P) = -\min x_{ij}$ is locally Lipschitz.

V. SIMULATIONS

In this section we demonstrate that when intruders have limited time to learn, solving for the patrolling strategies that are hard to learn along with being hard to attack can be more useful than optimizing for the worst case.

Strategy comparison: We conducted several experiments with n vertices placed in a square plane including obstacles. Each vertex was connected to a random number of its nearest neighbors and the euclidean distances between the vertices were rounded off to get integer edge weights. This construction of the graph is similar to the Probabilistic Roadmaps that are used for robotic path planning. The problem data for two of these instances is as follows.

Instance $1 (n = 7)$	Instance $2 (n = 10)$
$l = 8$	$l = 10$
$\psi = \rho = 5$	$\psi = \rho = 20$
$\phi = [10, 5, 10, 5, 10, 10, 5]$	$\phi_i \sim \mathcal{U}[20, 50]$
$T \sim \mathcal{U}[90, 110]$	$T \sim \mathcal{U}[200, 1000]$
$\Lambda \sim \mathcal{U}[0, 10]$	$\Lambda \sim \mathcal{U}[0, 10]$

We compare attacker's average payoffs $v_{\mathtt{lim}}$ and v_∞ for the 'hard to learn and attack' strategy $P_{\mathtt{lim}}$ (obtained by minimizing $f(P)$) and 'hard to attack' strategy P_∞ (obtained by minimizing $-x_{\tilde{i},\tilde{j}}$ where $\tilde{i}, \tilde{j} = \arg\max_{i,j}\{u_{ij}\}$). Since $\rho = \psi$, it is a zero sum game and the defender's payoff is symmetric. These average payoffs are calculated by simulating 100 attacks for each of these strategies and then taking the average of the payoff acquired by each of these attacks. The parameters T and Λ for these attackers were drawn randomly from their respective distributions.

Figure 1 shows the graph for Instance 1 along with both the strategies. For $P_{\mathtt{lim}}$, 67% attackers renege and $v_{\mathtt{lim}} = 1.96$, as compared to 1% reneging attackers and $v_\infty = 3.88$ for P_∞. If the intruder had perfect knowledge, the defender payoffs would be $v_\infty = 4.25$ and $v_{\mathtt{lim}} = 6.79$. It is to be expected since the optimization for P_∞ assumes the model in which the attackers will not renege due to lack of knowledge about the patrolling policy.

For Instance 2, we classify the attackers into different types based on the drawn parameters and show the average payoffs in Table I. For example, the attackers with Λ between 6.6 and 10 are risk averse attackers. Similarly the attackers with t from 200 to 466 are given less time to learn. Note that the strategy was calculated using $T \sim \mathcal{U}[200, 1000]$ and $\Lambda \sim \mathcal{U}[0, 10]$, and the attackers were classified during simulation to show the effect of T and Λ on attacker's payoffs.

For the $P_{\mathtt{lim}}$ patrolling strategy, 33 out of 100 attackers reneged. Some entries in the table for this strategy are zero because all of the attackers of that category reneged when this patrolling strategy was used. Note that for P_∞, corresponding entries are non zero, meaning that attackers decided to attack. In fact, none of the 100 attackers reneged for P_∞. The fraction of reneging attackers agrees with our

TABLE I

THE COMPARISON OF THE ATTACKER PAYOFFS FOR THE PATROLLING STRATEGIES. THE PAYOFFS ARE ROUNDED TO THE NEAREST INTEGER.

Λ	$v_{\texttt{lim}}$	v_{∞}	$v_{\texttt{lim}}$	v_{∞}	$v_{\texttt{lim}}$	v_{∞}	
$6.6 - 10$	0	28	0	33	38	32	
$3.3 - 6.6$	0	31	37	17	37	27	
$0 - 3.3$	32	21	37	29	37	46	
	$200 - 466$		$467 - 733$		$734 - 1000$		T

TABLE II

COMPARISON OF MATLAB'S *pattersearch* FUNCTION WITH ALGORITHM 1.

	patternsearch		Algorithm 1	
n	$-x_{\tilde{i},\tilde{j}}$	runtime	$-x_{\tilde{i},\tilde{j}}$	runtime
10	44.7	108	32.1	26
30	46.4	600	38.9	600
50	47.8	600	44.5	600

expression of probability of reneging used in (7). It can also be observed from the table that the risk averse attackers that are given less time to learn are more likely to renege. The average attacker payoff for $P_{\texttt{lim}}$ is 25 whereas it is 29 for P_{∞}. Hence, using the strategy for the attackers with limited learning time resulted in a 13.7% decrease of the attacker payoff. Also note that the starting point of the optimizations, which was a randomly generated Markov chain resulted in an average attacker payoff of 36.

Algorithm comparison: To evaluate the performance of Algorithm 1, we compare it with the MATLAB patternsearch function that is an adaptive mesh based direct search method [25]. For this experiment random graphs of different sizes were generated as described above in an obstacle free environment and the problem variables were generated as follows.

$$\begin{aligned} l &= 3/4(\text{MST length of graph}) \\ \phi &\sim \mathcal{U}[20, 50] \\ \psi &= \rho = 20 \end{aligned}$$

For the purpose of this comparison, $-x_{\tilde{i},\tilde{j}}$ was minimized, i.e., it was assumed that the attacker already knows defender's patrolling strategy. The algorithms were timed out at 10 minutes. The attacker's expected payoff $-x_{\tilde{i},\tilde{j}}$ for different sizes of the graph is reported in Table II, which shows that Algorithm 1 performs better in terms of final objective value and runtime.

VI. CONCLUSIONS AND FUTURE WORK

We considered a patrolling game where the attacker learns the patrolling strategy for a limited amount of time before deciding to attack or reneging. We showed that in the cases where the attackers have limited time, designing strategies that are hard to learn along with being hard to attack can be useful. The calculation of the variance in the estimates of the attacker is expensive and we would like to investigate methods to approximate this variance efficiently. Moreover, inclusion of observation costs in the model is an interesting direction for future work.

REFERENCES

[1] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport," in *AAMAS*, 2008, pp. 125–132.

[2] S. Alpern, A. Morton, and K. Papadaki, "Patrolling games," *Operations Research*, vol. 59, no. 5, pp. 1246–1257, 2011.

[3] N. Basilico, N. Gatti, and F. Amigoni, "Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder," *Artificial Intelligence*, vol. 184, pp. 78–123, 2012.

[4] D. Korzhyk, V. Conitzer, and R. Parr, "Solving Stackelberg games with uncertain observability," in *AAMAS*, 2011, pp. 1013–1020.

[5] K. Srivastava, D. M. Stipanović, and M. W. Spong, "On a stochastic robotic surveillance problem," in *IEEE Conference on Decision and Control and Chinese Control Conference*, 2009, pp. 8567–8574.

[6] J. Grace and J. Baillieul, "Stochastic strategies for autonomous robotic surveillance," in *IEEE Conference on Decision and Control and European Control Conference*, 2005, pp. 2200–2205.

[7] N. Agmon, G. A. Kaminka, and S. Kraus, "Multi-robot adversarial patrolling: facing a full-knowledge opponent," *Journal of Artificial Intelligence Research*, vol. 42, pp. 887–916, 2011.

[8] T. Sak, J. Wainer, and S. K. Goldenstein, "Probabilistic multiagent patrolling," in *Brazilian Symposium on Artificial Intelligence*. Springer, 2008, pp. 124–133.

[9] K. Y. Lin, M. P. Atkinson, T. H. Chung, and K. D. Glazebrook, "A graph patrol problem with random attack times," *Operations Research*, vol. 61, no. 3, pp. 694–710, 2013.

[10] R. Patel, P. Agharkar, and F. Bullo, "Robotic surveillance and Markov chains with minimal weighted Kemeny constant," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3156–3167, Dec 2015.

[11] A. B. Asghar and S. L. Smith, "Stochastic patrolling in adversarial settings," in *American Control Conference*, 2016, pp. 6435–6440.

[12] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez, and S. Kraus, "An efficient heuristic approach for security against multiple adversaries," in *AAMAS*, 2007, p. 181.

[13] M. Jain, D. Korzhyk, O. Vaněk, V. Conitzer, M. Pěchouček, and M. Tambe, "A double oracle algorithm for zero-sum security games on graphs," in *AAMAS*, 2011, pp. 327–334.

[14] Z. Wang, Y. Yin, and B. An, "Computing optimal monitoring strategy for detecting terrorist plots." in *AAAI*, 2016, pp. 637–643.

[15] T. Brázdil, P. Hliněný, A. Kučera, V. Řehák, and M. Abaffy, "Strategy synthesis in adversarial patrolling games," *arXiv preprint arXiv:1507.03407*, 2015.

[16] T. Alam, M. Edwards, L. Bobadilla, and D. Shell, "Distributed multi-robot area patrolling in adversarial environments," in *International Workshop on Robotic Sensor Networks*, Seattle, WA, USA, Apr. 2015.

[17] Y. Vorobeychik, B. An, M. Tambe, and S. P. Singh, "Computing solutions in infinite-horizon discounted adversarial patrolling games." in *ICAPS*, 2014.

[18] C. Kiekintveld, T. Islam, and V. Kreinovich, "Security games with interval uncertainty," in *AAMAS*, 2013, pp. 231–238.

[19] B. An, D. Kempe, C. Kiekintveld, E. Shieh, S. Singh, M. Tambe, and Y. Vorobeychik, "Security games with limited surveillance," in *AAAI Conference on Artificial Intelligence*, 2012, pp. 1241–1248.

[20] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[21] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[22] G. W. Bohrnstedt and A. S. Goldberger, "On the exact covariance of products of random variables," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1439–1442, 1969.

[23] P. J. Schweitzer, "Perturbation theory and finite Markov chains," *Journal of Applied Probability*, vol. 5, no. 2, pp. 401–413, 1968.

[24] C. Audet and J. E. Dennis Jr, "Mesh adaptive direct search algorithms for constrained optimization," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[26] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l 1-ball for learning in high dimensions," in *International Conference on Machine Learning*, 2008, pp. 272–279.