Title
Scaling up multi-agent patrolling in urban environments

Authors
Patil, Rohan
Langley, Alexander
Christensen, Henrik

Publication Date
2023-06-13

DOI
10.1117/12.2663719

Peer reviewed

Scaling up multi-agent patrolling in urban environments

Rohan Patil, Alexander Langley, and Henrik Christensen

UC San Diego, 9500 Gilman Dr., La Jolla, California

ABSTRACT

Patrolling is generally classified as either regular or adversarial patrolling. Regular patrolling aims to periodically visit important locations so that the duration between visits to locations is minimized. Regular patrolling, however, is typically deterministic. In the presence of an adversary which is able to observe the patrollers' behavior before deciding on a plan for intrusion, deterministic patrolling allows an intruder to invade a location when it knows that patrollers will be elsewhere. Adversarial patrolling addresses this problem by using stochastic strategies that resist the intruder's ability to learn and predict the patrollers' actions.

Keywords: Patrolling, multi-robot, adversarial, decentralized

1. INTRODUCTION

Multi-robot systems have gained interest in the security domain because of their ability to perform tasks which are difficult for a single robot.[1,2] In particular, patrolling is a task which benefits from the cooperation of multiple robots because of the large area which needs to be continuously surveilled. Broadly speaking, we can classify patrolling algorithms as either regular or adversarial.[3] In regular patrolling, the goal is for a team of robots to collectively optimize a performance metric, such as the time between visits to important locations.[4] The focus of this work, however, will be adversarial patrolling, where patrollers must factor the intruder's intent and possible strategies into their own decision-making.[5]

There is previous work on large-scale patrolling which considers the problem of patrolling a large area with a limited number of patrollers.[6,7] However, the problem of calculating the expected gain for an additional patroller is underexplored. It is important to calculate the expected gain for an additional patroller because it allows us to determine the expected returns on resources required to deploy another agent.

We present a patrolling algorithm that is robust and linear in the number of patrollers. The algorithm is robust because it can handle the case when the number of patrollers is reduced or increased. There can be multiple reasons for the number of patrollers to be reduced. For example, a patroller might fail, or a patroller might be reallocated to another task. It is important to be able to handle such cases because it allows us to quickly recalculate a strategy for the reduced number of patrollers. The robustness of the algorithm is further enhanced by the fact that it is linear in the number of patrollers and that a change in number of patrollers does not require re-calculation of strategy for all patrollers. This is important because it allows us to quickly calculate a strategy for many patrollers.

Our work addresses the problem of computing the expected detection rate for a patroller. The expected detection rate is important because it allows us to determine the expected returns on resources required to deploy another agent. We show that we can explicitly find the Markov transition matrix for the patrollers and use it to compute the expected detection rate for a particular intruder strategy. The construction of the Markov transition matrix is done in a way which allows us to quickly compute the expected detection rate for many patrollers. Moreover, the update of the transition matrix is linear in the number of patrollers, which allows us to quickly calculate the expected detection rate for a different number of patrollers.

The construction of the algorithm results in a partially observed Markov decision process (POMDP) from the perspective of the intruder. It is known that finding an optimal solution for a POMDP is intractable.[8,9] The patroller strategies leverage this fact, making it hard to find an optimal intruder strategy. We experiment with various intruder strategies under different scenarios and show that the intruder even with complete knowledge about the patroller strategies and intensive computation power finds it difficult to achieve a low detection rate.

Further author information: (Send correspondence to Rohan Patil)
Rohan Patil: E-mail: rpatil@ucsd.edu

2. RELATED WORKS

As previously stated, patrolling algorithms are broadly classified as either regular or adversarial. In regular patrolling, on one hand, a team of robots formulates its strategy to optimize a metric such as the time between visits to each location of interest.[3,10] Such strategies are generally deterministic,[10–12] and for this reason can be brittle in the face of a strategic intruder who can learn about the patrollers' behaviors. On the other hand, adversarial patrolling methods explicitly model the potential intruder, and seek to optimize a lower bound on the chance of detecting an intruder anywhere in the environment.[3] Adversarial patrolling algorithms generally produce stochastic patrolling behaviors that are more resistant to exploitation by an intruder.[6,13,14] Another feature of adversarial patrolling is that the intruder is able to observe the patrollers' strategies to choose the best possible attack.[5,15]

Game-theoretic methods are a popular choice for modeling the adversarial patrolling scenario, where it is assumed that the patrollers know the intruder's possible set of strategies and its utility function.[5,6,16,17] In particular, zero-sum Stackelberg games model scenarios in which patrollers must commit to a patrolling strategy that is resistant to attacks even when the intruder is able to learn their strategies perfectly, and have enjoyed wide adoption in adversarial patrolling.[5,16–18] In this work, we also assume that the intruder has access to patrollers' strategies and is able to optimize its attack against them. We expand on previous approaches, however, in two ways. First, we explicitly account for the intruder's learning process,[15] while maintaining the assumption of complete intruder knowledge. Second, rather than jointly optimizing all patrollers' strategies at once, the patrollers compute their strategies one after another. Closely resembling other iterative strategy optimization frameworks such level-k thinking,[1,19,20] each patroller considers what the intruder observes about previous patrollers' behaviors when formulating its strategy. We refer the reader to[3] for a comprehensive survey of both regular and adversarial patrolling.

Defining the intruder's strategy set is a crucial design choice when formulating an adversarial patrolling problem, and determines the way the intruders move throughout the environment and initiate attacks. In several prior works, the intruder is assumed to be able to appear instantaneously at any location in the environment,[6,7,15,16] which we shall refer to for the remainder of this work as the *heavy hitters problem*. Other prior work[21] considers mobile intruders whose strategy sets consist of paths through the environment, but only consider stationary sensors rather than mobile patrollers, for covering an area (*coverage problem*). Another variety of problem is where the aim is to control a perimeter and deny entrance to intruders.[18,22] In this case, the motion of the patroller is constrained to be along the perimeter. In contrast to these prior works, we consider the fully general case of both patrollers and intruders moving on arbitrarily connected graphs.

3. BACKGROUND AND FRAMEWORK

In the previous section, we discussed the various settings in which patrolling algorithms are used, and the challenges faced in designing them. Here, we will describe the problem formulation and the framework that we use to design patrolling algorithms.

3.1 Problem Statement and Formulation

The previous section discussed the heavy hitters problem, linear defense problem (perimeter patrol) and coverage problem. All these settings address the issue of patrolling an environment to maximize the probability of detecting an intruder. However, these problem settings in general consider neither limited intruder mobility, nor that certain locations are more likely to be targeted by the intruder, which is an important factor in designing patrolling algorithms. The added limitations weaken the intruder, however, it is not necessary that previous works benefit from it. Hence, we consider the real-world scenario where the intruder is limited by the environment and design an algorithm that tries to benefit from the limitations.

In our setup, we select potential points of attack and try to patrol them. The intruder is a given a set of entry points to enter the environment to reach the points of interest (objectives). It is assumed that the patrollers know these objective points, but not the entry points. The idea is that in real life scenarios, some intruders may have access via some entry points, while others may have access via other entry points. Another assumption is that the environment is most of the time large enough that it might be impossible to have complete coverage of the environment.

These assumptions create certain additional expectations from the patrolling algorithm.

- The patrolling algorithm should be able to detect the intruder as soon as possible.
- The algorithm should prioritize that the intruder can not reach the objective points over faster detection.
- Computation required to determine patroller movements during execution should be low.

In addition, when a human designs a patrolling strategy, we consider various factors like areas of which have a higher chance of being targeted and strategic points arising due to topography, which if compromised might give intruders an advantage. Ideally, we want patrolling algorithms to consider all these factors along with the above-mentioned requirements. However, encoding all this information obtained from human understanding presents a challenge.

We model our environment as a strongly connected directed planar graph $G(V, E)$, where the vertices V represent the patrol regions, and the edges E represent the connectivity, similar to some previous works.[4,23,24] Moreover, as it is a planar graph, we have $|E| = O(|V|)$. We assume that the patrollers can move from one vertex to another only if there is an edge between them. The intruder can enter the environment from a subset of vertices $\mathcal{E} \subset V$. The intruder and patroller interaction is modelled as a game. The patrollers are given a set of objective points $\mathcal{O} \subset V$ to patrol. At the start of the game, the intruder is given an objective point $o \in \mathcal{O}$. The patrollers are not aware of the entry points $\mathcal{E}$ and the intruder's objective point o. When the game starts, the patrollers start from some positions and the intruder starts from its preferred entry point. The game ends either when the intruder comes in the visibility of a patroller or the intruder reaches the objective point, with the first case resulting in a patroller win and the other in a loss.

3.2 Importance and Utility Matrix

Previous works[6,16] have used the concept of utility matrix to model strategy finding as a polymatrix game. We use a similar concept to model the patrolling problem, however, we use a utility matrix which is a function of the relative "importance" of the nodes. We introduce the concept of importance to model the strategic points in the environment. Importance is a real scalar which gives the relative loss to the patrollers if the intruder were to occupy the particular node. As such, it is never beneficial for the patrollers if the intruder is occupying any of the nodes. Hence, the value of importance of a node is always non-negative. Note that defining importance from the perspective of intruder's gain allows us to model both strategic points and points of interest to the intruder. For example, if a point is important for the patrollers from a strategic viewpoint, then by giving a high importance value will promote the patrollers to visit the partcular place for often. In contrast, points which need to be surveiled but do not necessarily have strategic importance will also receive a high importance value, and the relative difference between the importance values for a strategic point and say an objective point instructs the patrollers which is to be given a higher preference.

We use these importance values to construct a utility matrix for the patrollers. Utility matrix is a $|V| \times |V|$ matrix which gives the utility of a patroller if it is at a node u and the intruder is a node v. We choose a simple function for the construction of the utility matrix.

In this section, we describe how we can define a notion of importance for the nodes in G. In particular, this allows us to construct a utility matrix

For this work, we define the importance of a node v, represented by $\mathcal{I}(v)$ as

$$\mathcal{I}(v) = \max_{o \in \mathcal{O}} \ \alpha \exp\left(-\gamma d(o, v)\right) \tag{1}$$

where α and γ are positive hyperparameters, and $d(o, v)$ is the length of the shortest path connecting o and v inclusive. We call α the scaling factor or scale and γ the exponent scale.

The entry for the utility matrix corresponding to nodes u, v is given by

$$U_{(u,v)} = (\alpha - \mathcal{I}(v) + 1) \times \exp(-\gamma d(u, v)) \tag{2}$$

3.3 Patrol Group Modelling and Strategy Formalism

A patrol group of size n consists of patrollers $p_1, p_2, \ldots, p_n$. The strategy construction of patrollers follows a "Multi-leader single follower" Stackelberg model. The i^{th} patroller p_i acts as a follower of $p_1, p_2, \ldots, p_{i-1}$ and a leader of $p_{i+1}, p_{i+2}, \ldots, p_n$. This means that the patrollers sequentially decide their strategies based on the strategies of the patrollers that have already decided their strategies and prioritize points that are not already covered by the patrollers that have already decided their strategies.

We represent the strategy of the patroller as a vector $\sigma_i \in \Delta^{|V|}$*. The strategy of the patroller is a probability distribution over the vertices of the graph. The patroller samples a vertex from this distribution and moves to that vertex from its current position using the shortest path. The patroller repeats this process until the game ends, when the intruder is either detected or reaches its objective. Naturally, the probability distribution of the patroller p_i being at a vertex v at any given point in time is different from σ_i. Let ρ_i be the probability distribution of the patroller p_i being at a vertex v at any given point in time which we call as the prior distribution of p_i.

LEMMA 3.1. *Suppose that a patroller has a strategy σ. Then, the probability that it selects to move along the shortest path from vertex u to vertex v is given by $\sigma(u)\sigma(v)$.*

Proof. The patroller can move from u to v only if it is at vertex u and the vertex v is sampled from the strategy σ. The patroller can be at vertex u and is supposed to sample a vertex from σ only if the vertex u was previously sampled from σ. Hence, the probability that the patroller moves from u to v is given by $\sigma(u)\sigma(v)$.

□

LEMMA 3.2. *Given a patroller with a strategy σ, we have:*

$$P(v|a \to b) = \frac{\mathbb{I}(v \in a \to b)}{|a \to b|}$$

where $a \to b$ is the ordered set of vertices in the shortest path from a to b.

Proof. It is easy to see that $P(v|a \to b)$ will be 0 if v is not in the shortest path from a to b. In a shortest path, the vertices in the path cannot be repeated. Hence, the probability of being at a vertex v in the shortest path from a to b is same as the probability of being at any other vertex w in the shortest path from b to a. Hence, $P(v|a \to b) \propto \mathbb{I}(v \in a \to b)$. Hence, $P(v|a \to b) = \frac{\mathbb{I}(v \in a \to b)}{|a \to b|}$.

□

We can define a transition matrix $T_{a \to b}$ which moves the patroller from vertex a to vertex b along the shortest path from a to b. The transition matrix is defined as:

$$T_{v,u} = \mathcal{I}((u, v) \in a \to b)$$

where $(u, v) \in G(E)$ and $(u, v) \in a \to b$ means that edge (u, v) is in the shortest path from a to b.

THEOREM 3.3. *Given a patroller with a strategy σ for a strongly connected graph $G(V, E)$ which has a prior ρ, there exists a static transition matrix $T \in \mathbb{R}^{|V| \times |V|}$ such that:*

$$\forall_{v \in \Delta^{|V|}} \forall_{\epsilon > 0} \exists_{k \in \mathbb{N}} ||T^k v - \rho|| < \epsilon$$

Furthermore,

$$\rho_u = \sum_{v \in V} \left\{ \frac{\sigma(a)\sigma(b)}{|a \to b|} T_{a \to b} \right\}_{v,u}$$

*Δ^n is the unit n-simplex

Proof. If we show that there is a stochastic matrix T such that $T_{v,u} > 0$ only if $(u,v) \in E$ and $T_{v,u} = 0$ otherwise, and $T_{v,u}$ is equal to the probability of the patroller moving along the edge (u,v), then we have shown that T is a transition matrix for the patroller. As it is a strongly connected graph, T has a stationary distribution ρ which is the prior of the patroller.

Let us represent the probability of the patroller moving from vertex u to vertex v in a single step as $P(v|u)$. By definition, $(u,v) \notin E \implies P(v|u) = 0$. Hence, $P(v|u) = 0 \iff T_{v,u} = 0$. Moreover, $P(v|u)$ is equal to the probability of the patroller moving along the edge (u,v) in a single step. Hence, $P(v|u) = T_{v,u}$. In addition, let $P(u) = \rho_u$ be the probability of the patroller being at vertex u at any given point in time.

$$\begin{aligned}
P(v|u) &= \sum_{a\in V, b\in V} P(v, a \to b|u) \\
&= \sum_{a\in V, b\in V} P(v|a \to b, u) P(a \to b|u) \\
&= \sum_{a\in V, b\in V} \mathcal{I}((u,v) \in a \to b) P(a \to b|u) \\
&= \sum_{a\in V, b\in V} \mathcal{I}((u,v) \in a \to b) \frac{P(u|a \to b) P(a \to b)}{P(u)} \\
\implies P(v|u)P(u) &= \sum_{a\in V, b\in V} \mathcal{I}((u,v) \in a \to b) P(u|a \to b) P(a \to b)
\end{aligned}$$

Using lemmas 3.2 and 3.1, we have:

$$P(v|u)P(u) = \sum_{a\in V, b\in V} \frac{\sigma(a)\sigma(b)}{|a \to b|} \mathcal{I}((u,v) \in a \to b)$$

Construct a matrix M such that $M_{v,u} = P(v|u)P(u)$ for all $v, u \in V$. We have:

$$M = \sum_{a\in V, b\in V} \frac{\sigma(a)\sigma(b)}{|a \to b|} T_{a \to b}$$

It is easy to see that $\rho_u = P(u) = \sum_{v\in V} M_{v,u}$ for all $u \in V$.

Hence, we have:

$$P(v|u) = \frac{1}{\rho_u} \sum_{a\in V, b\in V} \frac{\sigma(a)\sigma(b)}{|a \to b|} \mathcal{I}((u,v) \in a \to b) = \frac{M_{v,u}}{\rho_u} \tag{3}$$

Hence, we have $T_{v,u} = \frac{M_{v,u}}{\rho_u}$ which is the transition matrix for the patroller.

□

From the above theorem, we have that the movements of the patroller can be modelled as a Markov chain. The transition matrix T is a function of the strategy, σ and the prior is the stationary distribution of the Markov chain. Furthermore, the theorem proof gives an explicit construction of the transition matrix T as well as the prior ρ.

Before discussing the construction of the patrol group, we define the notion of visibility matrix. Let $G(V,E)$ be a graph. The visibility matrix $\mathcal{V}$ is a matrix such that $\mathcal{V}_{v,u} = 1$ if u can see v and $\mathcal{V}_{v,u} = 0$ otherwise. The vector $p_i^{vis} = \mathcal{V}\rho_i$ gives the probability that a vertex v is visible to the patroller p_i. Note that p_i^{vis} is not a probability distribution over the vertices, as multiple nodes can be visible to the patroller at the same time.

Given a patrol group P_g, we define the visibility probability of a vertex v as the probability that the vertex is visible to at least one patroller in the patrol group. Let P_g^{vis} be the vector of visibility probabilities of the vertices in the graph.

THEOREM 3.4. *When a graph $G(V, E)$ is patrolled by the patrol group P_g with n patrollers with the visibility matrix $\mathcal{V}$:*

$$P_g^{vis} = 1 - \prod_{i=1}^{n}(1 - p_i^{vis}) = 1 - \prod_{i=1}^{n}(1 - \mathcal{V}\rho_i)$$

Proof. Let P_g^{vis} be the vector of visibility probabilities of the vertices in the graph. Let p_i^{vis} be the vector of visibility probabilities of the vertices in the graph for patroller p_i. Let $p_i^{vis} = \mathcal{V}\rho_i$.

Hence, we have that the probability that a vertex v is not visible to the patroller p_i is given by $1-(p_i^{vis})_v$. The probability that a vertex v is not visible to any patroller is given by $1-\prod_{i=1}^{n}(1-(p_i^{vis})_v)$. Hence, we have that the probability that a vertex v is visible to at least one patroller is given by $1-\prod_{i=1}^{n}(1-(p_i^{vis})_v) = 1-\prod_{i=1}^{n}(1-\mathcal{V}\rho_i)_v$. Hence, we have that $P_g^{vis} = 1 - \prod_{i=1}^{n}(1 - \mathcal{V}\rho_i)$.

□

To create a multi-leader single follower model, we use the patrol group vector of visibility probabilities P_g^{vis} when only i patrollers are present to construct the strategy of the $i+1$ patroller. In general, every patroller in the patrol group can have a different utility matrix. However, for simplicity, we assume that all patrollers have the same utility matrix U as defined in equation 2.

For the $i+1$ patroller, we have:

$$\sigma_{i+1} = \arg\max_{x\in\Delta^{|V|}} \min_{1\le i\le |V|} \left(U \circ \exp\left(-\lambda \frac{P_g^{vis}}{||P_g^{vis}||_\infty}\right)^T x \right)_i \tag{4}$$

where λ is a predetermined constant and $\circ$ represents element-wise multiplication.

In essence, we are promoting the $i+1$ patroller to patrol the vertices that are not visible to the other patrollers. The λ parameter is used to control the trade-off between the utility of the patroller and the visibility of the vertices. The larger the value of λ, the more the patroller will focus on the utility of the vertices. The smaller the value of λ, the more the patroller will focus on the visibility of the vertices. Hence, a large value of λ will create a higher chance for faster capture of the intruder. Therefore, λ is a hyperparameter that allows us to balance between assuring that the intruder is detected before reaching the goal and the desired speed of detection.

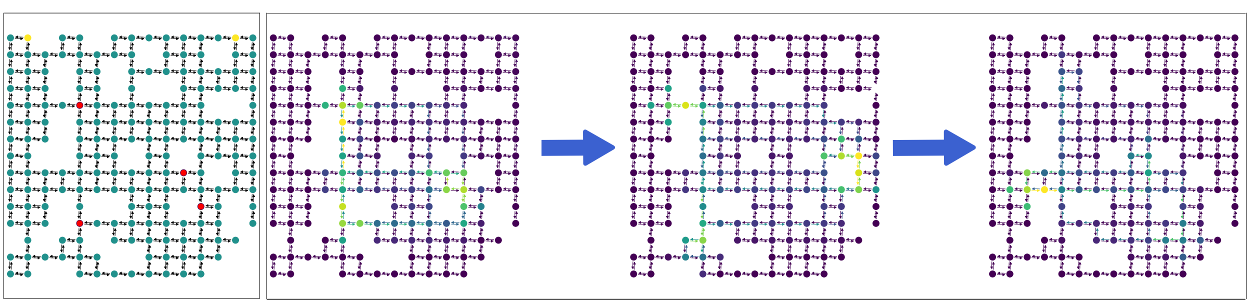

Figure 1. The graph on the left shows the objectives as red nodes and the entries as the yellow nodes. The remaining graphs from left to right show the visibility distribution for p_1, p_2, p_3 in the patrol group generated respectively. The higher the chance that the node is in visibility, the yellower the node. The edges are colored in a similar manner showing the transition probabilities.

Figure 1 shows the strategies learned by the patrollers in the multi-leader single follower model for a randomly generated graph and some selected objective points. The yellower the vertex, the more the patroller patrols the vertex. The patrollers patrol the vertices that are not visible to the other patrollers. The patrollers also patrol the vertices that are visible to the other patrollers but have a high utility. The patrollers patrol the vertices that are visible to the other patrollers but have a low utility less frequently. Figure 2 shows a schematic representation of how a patrol group strategy and prior calculation looks. There is supposed to be a connection between each ρ_i and σ_j where $i < j$ but, these edges are omitted for clarity.

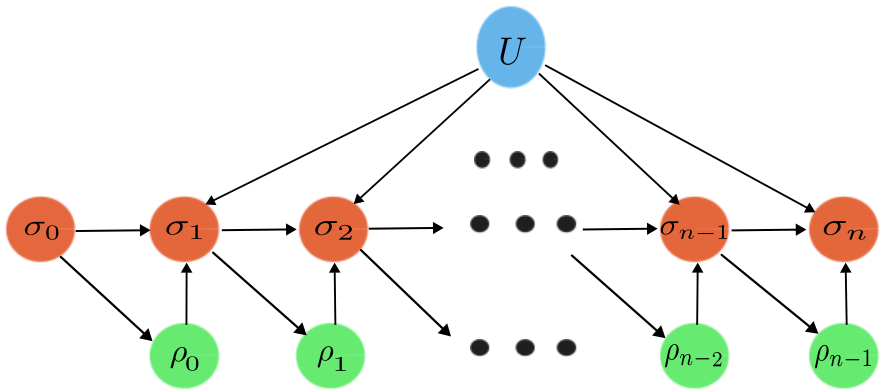

Figure 2. The figure schematically shows a dependency graph of values that depend on other values for the proposed patrol group. The utility matrix is common for all patrollers, represented by the blue circle. The σ's are the strategy of the patrollers and ρ's are the stationary distribution of the patrollers. The connections between ρ_i and σ_j for all $i < j$ are omitted as it only represents that instead of only using ρ_{i-1}, current P_g^{vis} will be used.

4. INTRUDER STRATEGIES AND EXPERIMENTAL SETUP

The intruder can use various strategies to attack the system. In this section, we describe the strategies that we consider in our experiments. We also describe the experimental setup that we use to evaluate the effectiveness of the proposed defense.

4.1 Intruder Strategies

The three strategies that we consider are as follows:

- **Simple**: The intruder takes the shortest path to the target node and then moves to the next node in the shortest path to the target node. This strategy takes very less computation. When multiple entries are present, the intruder chooses the entry which requires the least number of hops to reach the target node.
- **Greedy**: The intruder is provided access to the priors of the patrollers. The intruder then puts a cost on each vertex equal to the value of P_g^{vis} and chooses the path which has the least total cost.
- **Lookahead**: This strategy is similar to the greedy strategy. The intruder weighs the vertices according to patrol group visibility probabilities. After taking a step, the intruder uses the fact that it is not detected to redistribute the probability mass in priors of the patrollers to update its belief about the patroller positions. The patrol group visibility probabilities are then updated accordingly. The intruder then chooses the path which has the least total cost, takes a step and repeats the process until it reaches the target node.

Each strategy can result in different paths being taken by the intruder for the same target node. To evaluate the effectiveness of the path taken by the intruder, we use the expected detection rate as a metric. The expected detection rate is defined as the probability that the intruder is detected by the patrollers in a patrol group for a particular path.

Let $\mathbb{E}(P_g, (v_1, v_2, \ldots, v_n))$ denote the expected detection rate for a patroller group P_g and a path $(v_1, v_2, \ldots, v_n)$. The expected detection rate is defined as the sum of the probabilities of detection for each vertex in the path until the intruder is detected. The probability of detection at a vertex v_t is defined as the probability that the intruder is detected at v_t given that the intruder is not detected at any of the previous vertices.

THEOREM 4.1. *Let P_g be a patroller group with patrollers $p_1, p_2, \ldots, p_k$ and $v_1, v_2, \ldots, v_n$ be a path. Then, there is a recursive algorithm to calculate $\mathbb{E}(P_g, (v_1, v_2, \ldots, v_n))$.*

Proof. As the intruder moves from one vertex to another, information about the patroller positions is gained, as the intruder will move to the next vertex only if it is not detected. It implies that the belief of the intruder about the patroller positions will be updated.

Let ρ_i^t denote the belief of the intruder about the patroller p_i's position at the time t. By definition, $\rho_i^1 = \rho_i$ for all patrollers p_i.

We want to evaluate a path only until the intruder is not detected, and stop evaluating the path once the intruder is detected. If it was possible to evaluate only until the intruder is not detected, then we could have calculated the probability of detection itself. However, the intruder may have gotten detected somewhere in the path, and that creates the issue that we no longer know the probability of detection of for the rest of the path.

To overcome this issue, we can consider the expected detection rate as the sum of the probabilities of detection for each vertex in the path until the intruder is detected.

Let $X_1, X_2, \ldots, X_n$ be indicator random variables such that $X_t = 1$ if the intruder is detected at vertex v_t and $X_t = 0$ otherwise. Hence, we have,

$$\mathbb{E}(P_g, (v_1, v_2, \ldots, v_n)) = \sum_{t=1}^{n} \mathbb{E}(X_t)$$

However, we do not know the values of X_t for all t. We can only calculate the values of X_t for t such that the intruder is not detected at v_t. So we only do the summation until the expectation is less than unity. Furthermore, we add the constraint that only one of the X_i's can be 1 for all i such that $\sum_{j=1}^{i} \mathbb{E}X_j \leq 1$[†].

Let V^t represent the set of vertices visible to the patrollers at time t.

$$\implies P(X_t = 1) = P(v_t \in V^t) \prod_{j=1}^{t-1} (1 - P(v_j \in V^j))$$

Let $P_g^{vis}(t)$ represent the vector of visibility probabilities of the vertices in the graph at the time t for the patroller group P_g. Then, we have that $P(v_t \in V^t) = {P_g^{vis}(t)}_{v_t}$.

Using theorem 3.4, we have that $P_g^{vis}(t) = 1 - \prod_{i=1}^{k}(1 - \mathcal{V}\rho_i^t)$, where $\mathcal{V}$ is the visibility matrix of the graph.

Ideally ρ_i^t is a function of the history of the intruder being visible or not to the patroller p_i for all the previous time steps without the constraint that only one of the X_i's can be 1. However, as we are considering only one of the X_i's to be 1, we can calculate ρ_i^t under the assumption that for all previous time steps, the intruder was not in visibility. Hence, we have the following:

$$\rho_i^t = \Omega_{v_t}(T_i \rho_i^{t-1})$$

where T_i is the transition matrix of the patroller p_i and $\Omega_{v_t}(\rho_i^{t-1})$ is a function that returns the belief of the intruder about the patroller p_i's position at vertex v_t. The function $\Omega_{v_t}(w)$ is defined as follows:

[†] All the issues arise because there might be places where the intruder can be detected with probability 1 multiple times along the path. So, if we say that X'_t are 1 if intruder is detected and otherwise 0, then $P(X_t = 1) = P(X'_t = 1, X'_{t-1} = 0, \ldots, X'_1 = 0)$

$$(\Omega_{v_t}(w))_i = \begin{cases} 0 & \text{if } i \in \mathbb{V}(v_t) \\ \dfrac{w_i}{1 - \sum_{j \in \mathbb{V}(v_t)} w_j} & \text{otherwise} \end{cases}$$

where $\mathbb{V}(v_t)$ is the set of vertices from which v_t is visible.

Let $\Omega^n_{(v_1,v_2,\ldots,v_n),T}(w) = \Omega_{v_n}(T\Omega_{v_{n-1}}(T \ldots (T\Omega_{v_1}(w))))$. This implies that $\rho^t_i = \Omega^n_{(v_1,v_2,\ldots,v_t),T_i}(\rho_i)$. As the above equation is a recursive equation, it is possible to recursively calculate ρ^t_i.

Let $\Gamma_t = \prod_{j=1}^t \prod_{i=1}^k (1 - \mathcal{V}\rho^j_i)_{v_j}$. This implies that $\Gamma_t = (\Gamma_{t-1}) \prod_{i=1}^k (1 - \mathcal{V}\rho^t_i)_{v_t}$.

$$P(X_t = 1) = \Gamma_{t-1} \left(1 - \prod_{i=1}^k (1 - \mathcal{V}\rho^t_i) \right)_{v_t}$$

As it is possible to calculate Γ_t recursively, we can calculate the expected detection rate for each vertex in the path. We can also calculate the expected detection rate for the path as a whole using a recursive algorithm.

□

4.2 Experimental Setup

To experiment the effectiveness of the proposed algorithm, we generate random graphs with different number of vertices and edges. For each graph, we generate different configurations of objective points and entry points. We observe the effectiveness of the proposed algorithm for different number of patrollers with different visibility. The experiments assume that the visibility of both the patrollers and the intruder is the same. Furthermore, the number of objective points is always higher than the number of patrollers and that the entry points are always on the boundary of the graph. We add another constraint that the objectives need to be at least 6 hops away from the boundary of the graph.

For the purpose of comparison, we also consider the random patroller strategy and a modified version of the algorithm proposed by Langley, *et al.*(2022)[6] which we call the Nash algorithm. In random patroller strategy, the patrollers do a random walk on the graph. In the Nash algorithm, all patrollers have the same strategy and rather than selecting edges, it selects vertices. The vertices are weighed according to net probability of transiting to the vertex.

We use a base 18×18 graph for all the experiments. For each graph, we generate 3 different configurations of entries, with 2 entries in each configuration. The number of objectives is varied between 6 and 9 inclusive. For each entry configuration and objective count, 3 configurations of objectives are generated. The number of patrollers is varied between 2 and 5. The experiments are repeated 100 times for each configuration.

We use a total of 20 graphs for an experiment for a given visibility and hole probability. Hole probability represents the chance of dropping a vertex from the graph. We use a hole probability of $0.15, 0.25, 0.35$ for the experiments. The graphs are generated using the `networkx`[25] library in python. The visibility of the patrollers and the intruder is either 1 or 2.

5. OBSERVATIONS AND RESULTS

Figure 3 and figure 4 show the results when the visibility is 1 and, 2 respectively. We observe that the proposed algorithm outperforms the random patroller strategy and the modified version of the algorithm proposed in[6] in all cases. The dashed lines in the figures represent the averaged expected detection rate calculated, where the exact value is clipped to 1 if it is greater than 1 before averaging. Hence, the dashed lines are the clipped expected detection rates. The solid lines represent the experimental detection rate or experimental accuracy.

One of the reasons why we outperform the Nash equilibrium-based strategy, as the patrollers do not coordinate. Other reasons include that it does not encode the importance of the detection of the intruder, which acts as an explicit bias in our algorithm while determining patroller strategy. The graphs indicate that even

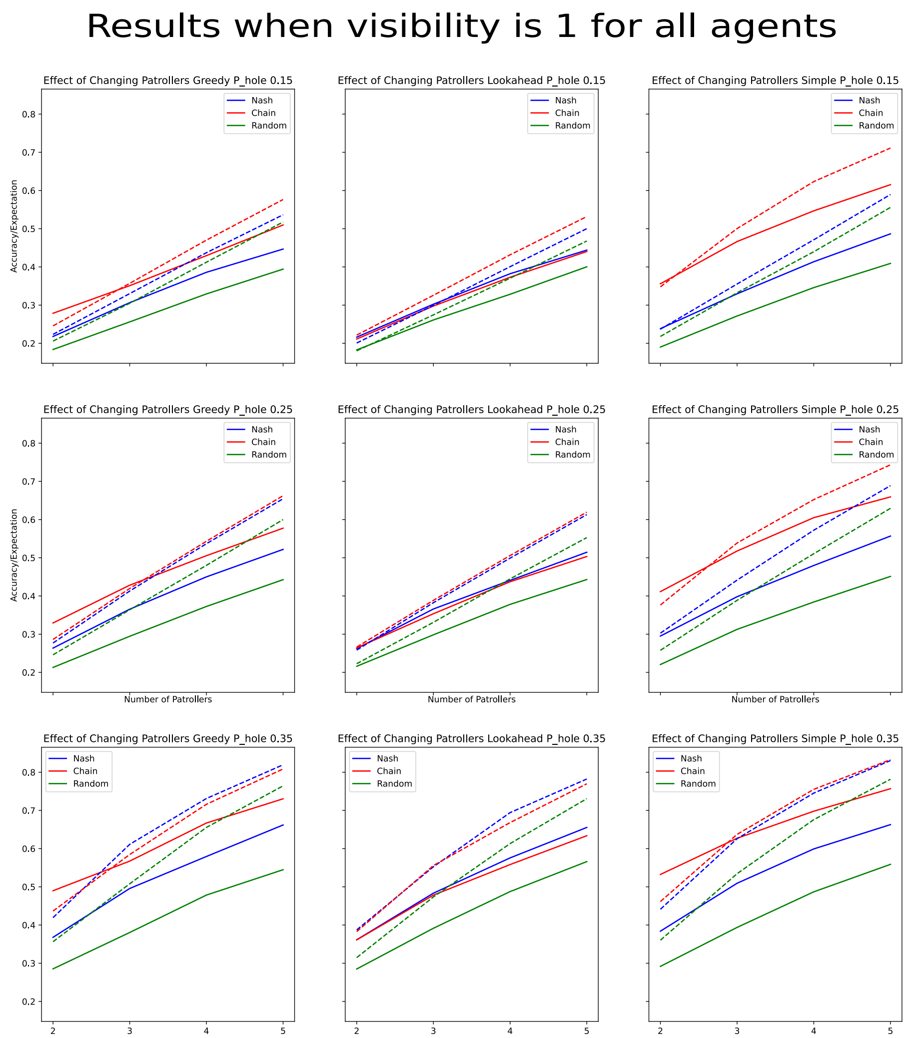

Figure 3. Shows the experimental accuracy with solid lines and the calculated averaged expected detection rate with dashed lines. The red color represents the proposed algorithm, the blue color represents the random patroller strategy and the green color represents the modified version of the Nash strategy[6]

when multiple patrollers use the same Nash equilibrium-based strategy, the detection rate is competitive with the proposed algorithm when the intruder uses sophisticated techniques like Look ahead strategy.

However, in cases where the intruder uses a simple strategy, like the shortest path, there is a large difference in accuracies between the proposed algorithm and the Nash equilibrium-based strategy. This shows that having the importance information is important for the patrollers to detect the intruder.

The clipped expected detection rate closely follows the experimental accuracy. As it is an expectation, it is always greater than or equal to the experimental accuracy. We also see the effect of increased visibility on the expected detection rate and the experimental accuracy. As expected, the accuracy increases with the visibility because the patrollers can cover more area with less movement.

6. CONCLUSIONS AND FUTURE WORK

We propose an algorithm that is scalable to many patrollers and is robust to change in number of patrollers. We also propose a method to encode human knowledge of the environment in the form of importance, which is used to determine the patroller strategy. The proposed algorithm can benefit more than other algorithms

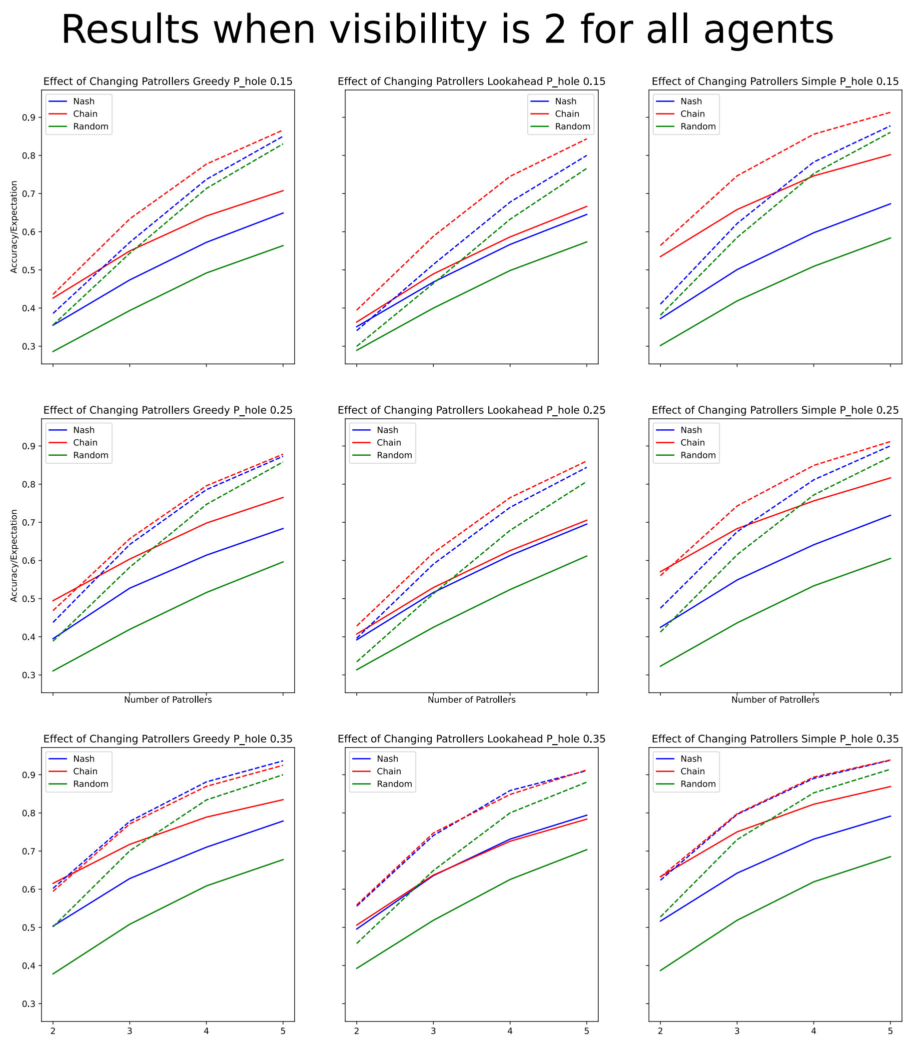

Figure 4. Same as figure 3 but when the visibility of the patrollers and the intruder is 2.

from the limitations that are faced by intruders in real life scenarios. We also show that the proposed algorithm can outperform the random patroller strategy and the modified version of the algorithm proposed in[6] in all the studied cases. The proposed definition of expected detection rate can be calculated in polynomial time and serves as a good measure to grasp the benefits and losses when a patroller is added or removed respectively.

Our future work includes exploring strategies which allow communication between patrollers during the execution of the patrolling algorithm while preserving the scalability, robustness and decentralized execution of the patrollers. Furthermore, in such a scenario, we can also explore the possibility of using the patrollers to detect the intruder and then use the patrollers to coordinate to capture the intruder. Some other directions are the case of multiple coordinating intruders, dynamically changing utility functions and patrolling in a dynamic environment.

REFERENCES

[1] Guan, Y., Maity, D., Kroninger, C. M., and Tsiotras, P., "Bounded-Rational Pursuit-Evasion games," in [2021 American Control Conference (ACC)], 3216–3221 (May 2021).

[2] Aydano, M., Almeida, A., Ramalho, G., Zucker, J., and Drogoul, A., "Multi-agent movement coordination in patrolling," in [Proceedings from the 1st Workshop on Agents in Computer Games. Edmonton, Canada. Retrieved January], **28**, 2010 (2002).

[3] Huang, L., Zhou, M., Hao, K., and Hou, E., "A survey of multi-robot regular and adversarial patrolling," IEEE/CAA Journal of Automatica Sinica **6**(4), 894–903 (2019).
[4] Portugal, D. and Rocha, R., "Msp algorithm: multi-robot patrolling based on territory allocation using balanced graph partitioning," in [Proceedings of the 2010 ACM symposium on applied computing], 1271–1276 (2010).
[5] Vorobeychik, Y., An, B., and Tambe, M., "Adversarial patrolling games," in [2012 AAAI Spring Symposium Series], (2012).
[6] Langley, A., Dhiman, V., and Christensen, H., "Heterogeneous multi-robot adversarial patrolling using polymatrix games," in [Advances in Automation, Mechanical and Design Engineering: SAMDE 2021], 13–27, Springer (2022).
[7] Alam, T., Rahman, M. M., Carrillo, P., Bobadilla, L., and Rapp, B., "Stochastic Multi-Robot patrolling with limited visibility," J. Intell. Rob. Syst. **97**, 411–429 (Feb. 2020).
[8] Madani, O., Hanks, S., and Condon, A., "On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems," in [AAAI/IAAI], 541–548 (1999).
[9] Papadimitriou, C. H. and Tsitsiklis, J. N., "The complexity of markov decision processes," Mathematics of operations research **12**(3), 441–450 (1987).
[10] Santana, H., Ramalho, G., Corruble, V., and Ratitch, B., "Multi-agent patrolling with reinforcement learning," in [Autonomous Agents and Multiagent Systems, International Joint Conference on], **4**, 1122–1129, jmvidal.cse.sc.edu (2004).
[11] Pham, H. X., La, H. M., Feil-Seifer, D., and Nefian, A., "Cooperative and distributed reinforcement learning of drones for field coverage," (Mar. 2018).
[12] Agmon, N., Fok, C.-L., Emaliah, Y., Stone, P., Julien, C., and Vishwanath, S., "On coordination in practical multi-robot patrol," in [2012 IEEE International Conference on Robotics and Automation], 650–656 (May 2012).
[13] Lin, E. S., Agmon, N., and Kraus, S., "Multi-robot adversarial patrolling: Handling sequential attacks," Artificial Intelligence **274**, 1–25 (2019).
[14] Xu, L., Perrault, A., Fang, F., Chen, H., and Tambe, M., "Robust reinforcement learning under minimax regret for green security," (June 2021).
[15] Alvarenga, C. D., Basilico, N., and Carpin, S., "Multirobot patrolling against adaptive opponents with limited information," in [2020 IEEE International Conference on Robotics and Automation (ICRA)], 2486–2492, IEEE (2020).
[16] Jiang, A. X., Yin, Z., Zhang, C., Tambe, M., and Kraus, S., "Game-theoretic randomization for security patrolling with dynamic execution uncertainty.," in [AAMAS], 207–214 (2013).
[17] Vorobeychik, Y. and Singh, S., "Computing stackelberg equilibria in discounted stochastic games," AAAI **26** (July 2012).
[18] Lei, X., Hu, X., Wang, G., and Luo, H., "A multi-UAV deployment method for border patrolling based on stackelberg game," Journal of Systems Engineering and Electronics **34**, 99–116 (Feb. 2023).
[19] Crawford, V. P., Costa-Gomes, M. A., and Iriberri, N., "Structural models of nonequilibrium strategic thinking: Theory, evidence, and applications," J. Econ. Lit. **51**, 5–62 (Mar. 2013).
[20] Dai, Z., Chen, Y., Low, B. K. H., Jaillet, P., and others, "R2-B2: Recursive reasoning-based bayesian optimization for no-regret learning in games," on Machine Learning (2020).
[21] Shishika, D., Macharet, D. G., Sadler, B. M., and Kumar, V., "Game theoretic formation design for probabilistic barrier coverage," in [2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)], 11703–11709, IEEE (2020).
[22] Agmon, N., Kaminka, G. A., and Kraus, S., "Multi-robot adversarial patrolling: facing a full-knowledge opponent," Journal of Artificial Intelligence Research **42**, 887–916 (2011).
[23] Alpern, S., Morton, A., and Papadaki, K., "Patrolling games," Operations research **59**(5), 1246–1257 (2011).
[24] De Nittis, G., Marchesi, A., and Gatti, N., "Computing the strategy to commit to in polymatrix games," in [Proceedings of the AAAI Conference on Artificial Intelligence], **32**(1) (2018).
[25] Hagberg, A. A., Schult, D. A., and Swart, P. J., "Exploring network structure, dynamics, and function using networkx," in [Proceedings of the 7th Python in Science Conference], Varoquaux, G., Vaught, T., and Millman, J., eds., 11 – 15 (2008).