An Empirical Method for Benchmarking Multi-Robot Patrol Strategies in Adversarial Environments

James C. Ward and Edmund R. Hunt
University of Bristol

In the field of multi-robot patrolling, graph-based environment models are a popular approach for designing and testing distributed multi-robot patrol strategies. These strategies are typically optimized for regular visiting of the vertices of the patrol graphs. However, analysis of these strategies against potential attackers is limited. We present an empirical, simulation-based method to assess performance of multi-agent patrol strategies against potential adversaries by estimating the probability of simulated attackers succeeding against the patrol agents. We show that this approach can provide new insights into performance that would not be found in standard non-adversarial analysis.

I. Introduction

PREVIOUS work in the field of robot patrol has described "adversarial patrolling", in which an adversary or attacker is attempting to avoid detection by the patrol agents. Other work has also provided distributed strategies for robot teams to employ to monitor areas efficiently. However, little work has been carried out to assess how these strategies would perform in adversarial settings.

In this work, we propose a method of benchmarking and comparison for multi-robot patrol strategies in adversarial environments, by estimating the performance of an adversary against the patrol agents. In Section 2, we discuss the background and related work. In Section 3, we define the problem we wish to tackle. In Section 4, we describe our method and the work carried out to validate our approach. In Section 5, we present an initial comparison of three patrol strategies. In Section 6 we conclude this work.

II. Background

A. Multi-robot patrol

Multi-robot patrol has seen a growing body of research over the past two decades [1] due to its applicability to environmental monitoring and security. The core idea is that a team of mobile autonomous agents should monitor an environment, by repeated observation of points of interest. Within this broad problem definition there have been many approaches: one popular one, and the one considered in this work, is to model the environment as a weighted undirected graph [2]. In such a graph the vertices can represent either points of interest or locations chosen based on the sensor ranges of the agents to fill the environment. The edge weights represent distance or travel time between these points. For such a model, a common criterion used to both optimize and assess performance of a patrol strategy is vertex idleness, defined as the time since a specified vertex was last visited by a patrol agent [2]. Designing an optimal patrol strategy to minimize idlenesses over time on such a patrol graph has been shown to be NP-complete in both the multi-agent case [3] and the single agent case [4]. Consequently, work with such a model is typically concerned with designing distributed strategies for cooperative teams of agents [5], [6], [7], or for adversarial settings [8], [9], [10].

B. Adversarial patrol

The multi-robot patrol problem can be complicated by the addition of an adversary: an external agent, seeking to bypass the patrollers to gain temporary access to the environment. Adversarial patrolling has had significant examination in the context of fence patrolling [11], [12], [13], but is more complicated in less constrained environments. In the case of a patrol graph, the goal of an adversary is often modeled as spending a certain amount of time at a given vertex without being visited by a patrol agent. Fully deterministic patrol strategies are considered inappropriate in adversarial settings [12], because an adversary with full knowledge of the patrol strategy could select times to attack vertices that would guarantee no visit by a patrol agent for some amount of time. Consequently a range of stochastic approaches are taken: one popular approach is to build a Markov chain on the patrol graph [14] to minimize idleness and maximize probability of detecting an adversary [8], [10], but such approaches are typically only designed for single patrol agents. Game theoretical approaches have also been examined [15], as the case of a single patroller and adversary can be tackled as a Bayesian Stackelberg Game [16]. Game theoretical approaches which assume imperfect knowledge have also been examined [17], but also typically only consider a single patrol agent. Adversarial strategies involving multiple patrol agents are less-well examined, as they are impractical to attempt to solve directly, and are the focus of this work.

C. Previous benchmarking work

Some previous work exists comparing of distributed multi-robot patrol strategies against different criteria in a non-adversarial setting. This has been carried out with a simulator [18] built for multi-robot patrolling, and has assessed several patrol strategies designed to minimize idleness on the basis of performance and scalability [5].

D. Patrol simulation

ROS Patrolling Sim[1] is a framework for simulation of multi-robot patrol systems [18]. It allows for easy simulation of patrol teams using a range of provided strategies, and supports addition of new strategies. The patrolling sim is built on Stage[2], allowing for full simulation of the patrol agents including lidar and odometry. The simulator includes several maps and logs large amounts of data from the simulation, including visit times to all vertices.

III. Problem Definition

We model the environment as an undirected, weighted graph of vertices representing points in the environment and edges representing routes between vertices, with edge weights correspond to travel time between vertices: the patrol graph. The idleness of a vertex is defined as the time since it was last visited by a patrol agent, and we define the vulnerability of a vertex as the time until it will next be visited by a patrol agent.

The patrol system is defined as the combination of the patrol graph and all patrol agents. The observable state of the patrol system is defined as the poses and velocities of the patrol agents, and the idlenesses of the vertices of the patrol graph.

The problem is then to assess the performance of patrol strategies within a patrol system against a range of adversary models, including intelligent adversaries capable of predicting vulnerabilities of vertices from the observable state of the patrol system. Full-knowledge adversaries, which know the details of the patrol strategies being used, are outside the scope of this work. The goal of an adversary is to make a single attack at a target vertex, which succeeds if the vertex is then not visited by a patrol agent for a specified amount of time. If the attack fails, the adversary has failed.

A. Adversary models

Asghar and Smith [10] have previously proposed four adversary models. The first two models informed the design of our "random" and "simple" attack time selection methods, but as the remaining models are full knowledge adversaries they are not considered here. We consider the adversary model as consisting of separate methods for target vertex selection and attack time selection.

The target selection methods are as follows:

1) Probabilistic target selection: The target vertex is selected randomly from a given distribution on the patrol graph
2) Opportunistic target selection: The target vertex is selected to maximise expected probability of a successful attack

The attack time selection methods are as follows:

1) Random attack time selection: The target vertex is attacked at a random time
2) Simple attack time selection: The target vertex is attacked as soon as a patrol agent leaves
3) Intelligent attack time selection: The target vertex is attacked at a time selected to maximize expected probability of a successful attack

Each adversary modl must also include an attack duration τ, the time it must spend undetected at the target to succeed, and a time window T in which an attack is to be made.

[1] http://wiki.ros.org/patrolling_sim
[2] http://playerstage.sourceforge.net/index.php?src=stage

IV. Benchmarking Method

To assess patrol strategies in a way that allows for direct comparison between different strategies, we estimate the probability of success of an adversary against the patrol strategy in a given patrol system. Varying the patrol strategy but holding all other properties of the system constant then allows for comparison of performance.

The following approaches calculate estimators of adversary success probability, using empirical (simulated) data of the behavior of the patrol system.

A. Random attack time selection

An estimate of probability of a successful attack at a given vertex with a time window T and attack duration τ is given by

$$\frac{1}{T}\sum_{t=0}^{T}(1 \text{ if } \tau \leq V(t), 0 \text{ otherwise}) \tag{1}$$

Where $V(t)$ is the vulnerability of the target vertex at time t.

B. Simple attack time selection

An estimate of probability of a successful attack at a given vertex with a time window T and attack duration τ is given by:

$$\frac{1}{n}\sum_{i=0}^{n}(1 \text{ if } \tau \leq V(t_i), 0 \text{ otherwise}) \tag{2}$$

Where n is the number of visits by patrol agents to the vertex in the time window, and $V(t_i)$ is the vulnerability of the vertex at the time of the i^{th} patrol agent visit.

C. Intelligent attack time selection

The goal of an adversary using this method is to select its time of attack to maximize the expected probability of success in the time window T. We use the observable state of the patrol system to predict the probability of successful attack at the target vertex for a given τ. From the observable state we generate three predictors :

1) Distance metric m_d: sum of $1/d$ over all patrol agents, where d is the length of the shortest path between the agent and the target vertex
2) Velocity metric m_v: sum of $\dot{d}/d$ over all patrol agents, where d is the length of the shortest path between the agent and the target vertex
3) Instantaneous idleness i at the target vertex

These three predictors then, in principle, can be fit to an equation:

$$p_e(\text{success}|\text{state}) = p_e(\tau \leq V(t)) = f(m_d(t), m_v(t), i(t)) \tag{3}$$

To calculate the expected probability of attack duration being less than or equal to instantaneous vulnerability, and therefore an attack succeeding, for a given observable state

If we discretize into timesteps of one second, the probability of a successful attack within the time window T is given by:

$$\begin{aligned} p(\text{success}) &= p(s) + (1-p(a))p(s) + ... + (1-p(a))^{T-1}p(s) \\ &= p(s)\sum_{i=1}^{T}(1-p(a))^{i-1} \\ &= \frac{p(s)}{p(a)}(1-(1-p(a)^T) \end{aligned} \tag{4}$$

Where $p(s)$ is the probability of a successful attack in a single timestep, and $p(a)$ is the probability of any attack in a single timestep. We maximize this by selecting a set S states such that:

$$p(a) = p(\text{state} \in S) \tag{5}$$

And:

$$p(\text{s}) = \sum_{i \in S} p(\text{state}_i)p(\text{success}|\text{state}_i) \tag{6}$$

Where "state" refers to the combination of m_d, m_v, i rather than the entire observable state.

Having followed this process, we are left with a set S of states in which the adversary should attack to maximize its expected probability of success, and the probability that the adversary will succeed given S.

D. Target selection

If the expected probability of adversary success is calculated for each vertex in the patrol graph, the final expected success probability is given by applying the adversary target selection method over these probabilities. In the case of probabilistic target selection, the final expected probability can be obtained from an appropriately weighted mean over all vertices. In the case of opportunistic target selection, the final expected probability is the maximum value over all vertices. In the analysis carried out in this work, we consider probabilistic target selection with the mean expected probability over all vertices as the final probability of attack success.

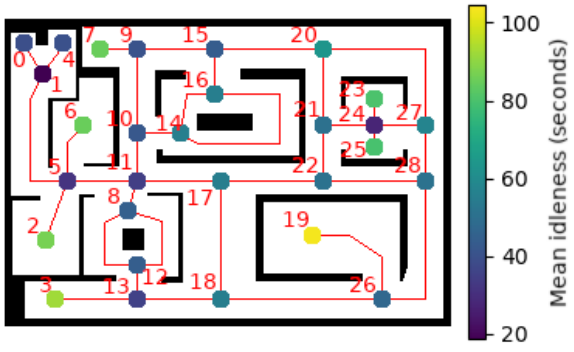

Fig. 1. The patrol graph and metric map, with vertex color corresponding to idleness using Greedy Bayesian Strategy (GBS). Based on an image provided in ROS Patrolling Sim

E. Testing and validation of method

This method was tested on data collected in ROS Patrolling Sim. A team of six agents was simulated in the map shown in Fig. 1, for six hours at a time. Five such simulations were carried out for two strategies: GBS and SEBS [5], both distributed strategies aiming to minimize idleness over the patrol graph. For all adversary models, we selected attack duration $\tau = 50, 100$, and 200 seconds, and attack time window $T = 1000$ seconds. We then trained an intelligent adversary on each of the five datasets for each patrol strategy, and tested each of these adversaries on the four remaining datasets. The simple and random adversary models were tested on all five datasets for each patrol strategy.

For our intelligent adversary model to be useful, we require: (1) the attack strategies produced can, depending on the patrol strategy, outperform non-intelligent adversary models against test datasets; and (2) the estimates of success probability are consistent across different training and test datasets for the same patrol strategy and patrol system. If a patrol strategy is sufficiently unpredictable, condition (1) may not be met for that specific strategy.

The full results from this test are shown in tables I and II. For both GBS and SEBS, intelligent adversaries show a clear increase in mean success probability over simpler adversary models, satisfying our first requirement. In the one exception, SEBS for $\tau = 200$ s, no adversary model could perform well enough against SEBS to give meaningful differences between them. Our second requirement, that the estimates of success probability should be similar across different training and test datasets, is also shown to be satisfied by our results, in particular, the coefficients of variation c_v.

V. Initial Strategy Comparison

To demonstrate the value of our approach, we present a limited analysis of three high-performing distributed strategies from the literature: SEBS [5], DTA-Greedy, and DTAP [7]. These were selected based on the strategy comparison from [7], which shows these as the three highest performing strategies based on idleness metrics. Moving forwards, we plan to carry out a full examination of these strategies in addition to CBLS [19], a more recent and higher-performing variant of the same idea as SEBS. This full examination will include testing on several different maps and team sizes ranging from 1-12 agents in order to obtain a more thorough comparison of these strategies in adversarial settings.

In our initial analysis, using the same methodology as was discussed in Section IV-E, patrol teams of size four, six, and eight were simulated on the map shown in Figure 1 using SEBS, DTAP, and DTAG, with τ varying from 25 to 200 seconds. Figure 2 shows the key outcome of this test, showing adversary success probability and the most successful patrol strategy as patrol team size and adversary attack duration vary. Comparing this to table III, which shows mean and maximum idlenesses, reveals an interesting result: that conclusions on performance in adversarial settings cannot be drawn from idleness metrics alone, supporting the need for specific analysis in adversarial settings. This is shown by SEBS having shown

TABLE I
ADVERSARY PERFORMANCE (SUCCESS PROBABILITY) AND MEAN VERTEX IDLENESS WITH SEBS FOR $\tau = 50, 100, 200$S

SEBS	$\tau = 50s$			$\tau = 100s$			$\tau = 200s$		
	μ	σ	c_v	μ	σ	c_v	μ	σ	c_v
p(success|intelligent)	0.498	0.0106	0.00455	0.0824	0.00623	0.0757	0.000727	0.00129	1.77
p(success|simple)	0.415	0.00283	0.00681	0.0746	0.00101	0.0135	0.000414	0.000181	0.436
p(success|random)	0.190	0.00129	0.0681	0.0240	0.00106	0.0442	0.0000668	0.000917	1.37

TABLE II
ADVERSARY PERFORMANCE (SUCCESS PROBABILITY) AND MEAN VERTEX IDLENESS WITH GBS FOR $\tau = 50, 100, 200$S

GBS	$\tau = 50s$			$\tau = 100s$			$\tau = 200s$		
	μ	σ	c_v	μ	σ	c_v	μ	σ	c_v
p(success|intelligent)	0.598	0.0324	0.0543	0.196	0.0115	0.0588	0.0207	0.00784	0.379
p(success|simple)	0.471	0.00750	0.0159	0.173	0.00346	0.0200	0.00699	0.00159	0.227
p(success|random)	0.335	0.00569	0.0170	0.0901	0.00602	0.668	0.00500	0.00124	0.247

the lowest mean idleness in all cases and the lowest maximum idleness for 6 and 8 agents, but each of the three strategies giving the lowest probability of adversary success for some combination of parameters.

TABLE III
IDLENESS METRICS

	SEBS		DTAG		DTAP	
No. agents	μ	max	μ	max	μ	max
4	75.1	421	95.4	535	95.2	357
6	40.2	252	50.0	367	52.2	295
8	30.4	181	43.4	322	46.2	417

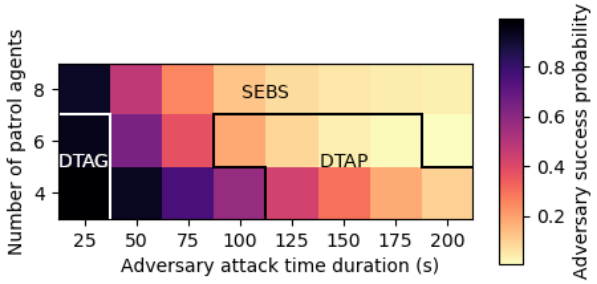

Fig. 2. Adversary success probability against τ and number of patrol agents, marked with most successful patrol strategy

VI. CONCLUSIONS

The results presented here indicate that our method can be useful in comparing multiple patrol strategies in adversarial settings. While the intelligent adversary strategies generated by this method are unlikely to be exactly optimal, due to both the dimensionality reduction introduced by our use of state metrics and the subsequent discretization of the state space, they still appear to be fit for purpose as a tool for comparison.

The proposed use of this method is to offer insight into the behavior and performance of multi-robot patrol strategies beyond measures of idleness. From the initial examination of SEBS, DTAG, and DTAP presented in this work, we have shown that comparative performance of patrol strategies in adversarial settings cannot be determined exclusively from idleness measures. In our test case, we have found that all three of SEBS, DTAG, and DTAP may be better choices depending on patrol team size and assumptions about adversary attack duration. We anticipate that the patrol map and graph will also inform patrol strategy selection, and this will be examined in future work.

REFERENCES

[1] N. Basilico, "Recent trends in robotic patrolling," *Current Robotics Reports*, vol. 3, no. 2, pp. 65–76, June 2022.

[2] A. Machado, G. Ramalho, J. Zucker, and A. Drogoul, "Multi-agent patrolling: An empirical analysis of alternative architectures," in *Proceedings of the 3rd International Conference on Multi-Agent Based Simulation II*, Bologna, Italy, July 2002, pp. 155–170.

[3] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 592–606, June 2012.

[4] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, September 2004, pp. 302–308.

[5] D. Portugal and R. P. Rocha, "Distributed multi-robot patrol: A scalable and fault-tolerant framework," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1572–1587, December 2013.

[6] C. Yan and T. Zhang, "Multi-robot patrol: A distributed algorithm based on expected idleness," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, November 2016.

[7] A. Farinelli, L. Iocchi, and D. Nardi, "Distributed on-line dynamic task assignment for multi-robot patrolling," *Autonomous Robots*, vol. 41, no. 6, pp. 1321–1345, August 2017.

[8] T. Alam, "Decentralized and nondeterministic multi-robot area patrolling in adversarial environments," *International Journal of Computer Applications*, vol. 156, no. 2, pp. 1–8, December 2016.

[9] A. B. Asghar, "Multi-robot path planning for persistent monitoring in stochastic and adversarial environments," Ph.D. dissertation, University of Waterloo, Waterloo, Canada, 2020.

[10] A. B. Asghar and S. S. L., "Stochastic patrolling in adversarial settings," in *Proceedings of the 2016 American Control Conference (ACC)*, Boston, USA, July 2019, pp. 6435–6440.

[11] N. Agmon and S. Kraus, "Multi-robot adversarial patrolling: Handling sequential attack," *Artifical Intelligence*, vol. 274, pp. 1–25, February 2019.

[12] N. Agmon, G. A. Kaminka, and S. Kraus, "Multi-robot adversarial patrolling: Facing a full-knowledge opponent," *Journal of Artificial Intelligence Research*, vol. 42, no. 1, pp. 887–916, September 2011.

[13] E. Sless, N. Agmon, and S. Kraus, "Multi-robot adversarial patrolling: Facing coordinated attacks," in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, Paris, France, May 2014, pp. 2093–1100.
[14] X. Duan and F. Bullo, "Markov chain-based stochastic strategies for robotic surveillance," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 243–264, May 2021.
[15] E. Hernández, J. del Cerro, and A. Barrientos, "Game theory models for multi-robot patrolling of infrastructures," *International Journal of Advanced Robotic Systems*, vol. 10, no. 3, March 2013.
[16] P. Paruchuri, J. P. Pearce, S. Kraus, and J. Marecki, "Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, May 2008, pp. 895–902.
[17] A. B. Asghar and S. L. Smith, "A patrolling game for adversaries with limited observation time," in *Proceedings of the 2018 IEEE Conference on Decision and Control (CDC)*, Miami Beach, USA, December 2018, pp. 3305–3310.
[18] D. Portugal, L. Iocchi, and A. Farinelli, "A ros-based framework for simulation and benchmarking of multi-robot patrolling algorithms," in *Robot Operating System (ROS)*. Springer, 2018, vol. 3, pp. 3–28.
[19] D. Portugal and R. R. P., "Cooperative multi-robot patrol with bayesian learning," *Autonomous Robots*, vol. 40, pp. 929–953, October 2015.