IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

EM-Patroller: Entropy Maximized Multi-Robot Patrolling with Steady State Distribution Approximation

Hongliang Guo, Qi Kang, Wei-Yun Yau, Marcelo H. Ang Jr. and Daniela Rus

***Abstract*—This paper investigates the multi-robot patrolling (MuRP) problem in a discrete environment with the objective of achieving uniform node coverage probability distribution by the robot team. Existing MuRP solutions for uniform node coverage either involve high computational complexity for the global optimal solution or rely on heuristics for approximate solutions without performance guarantees. To bridge the gap, we propose an efficient iterative algorithm, namely Entropy Maximized Patroller (EM-Patroller), with the per-iteration performance improvement guarantee and polynomial computational complexity. We reformulate the MuRP problem as an "unnormalized" joint steady state distribution entropy maximization problem and use multi-layer perceptron (MLP) to model the relationship between each robot's patrolling strategy and the individual steady state distribution. We derive a multi-agent model-based policy gradient method to update the robots' patrolling strategies towards the optimum. Complexity analysis indicates the polynomial computational complexity of EM-Patroller, and we show that EM-Patroller has additional benefits of accommodating user-defined joint steady state distributions and incorporating other objectives such as entropy maximization of individual steady state distribution. We compare EM-Patroller with state-of-the-art MuRP algorithms in various canonical MuRP environments and deploy it to a real multi-robot system for patrolling in a self-constructed indoor environment.**

***Index Terms*—un-normalized joint steady state distribution, multi-robot patrolling, multi-agent model-based policy gradient.**

I. Introduction

MULTI-ROBOT patrolling (MuRP) aims at protecting a physical environment by deploying multiple robots to persistently travel around it and perform local observations for security purposes. MuRP has application potentials in various scenarios, such as surveillance and vigilance for regional security, hazardous environment monitoring, patrolling and disinfecting a COVID-19 infected area. Many of the aforementioned tasks are mundane, dangerous and/or costly for human beings, and thus they serve as well suited use cases for multi-robot systems (MRSs).

To date, researchers have developed various algorithms as MuRP solutions, and a brief literature review is provided in Section II. Here, we wish to articulate that prevailing MuRP methodologies for the uniform node coverage problem can be roughly categorized into two groups, namely (1) optimization methods, which formulate MuRP as a (multi-agent) travelling salesman problem (TSP), and incur non-polynomial computational complexity algorithms for the global optimal solution; and (2) heuristic algorithms, which design various local heuristics/rules to foster efficient multi-robot collaboration. Optimization methods are able to deliver the optimal solution at the cost of high computational complexity. On the other hand, heuristic algorithms yield simple yet effective patrolling strategies but cannot offer any global performance guarantee.

This paper aims at bridging the research gap by proposing an efficient optimization method, which guarantees the per-iteration performance improvement and meanwhile possesses polynomial computational complexity. Specifically, we propose an Entropy Maximized Patroller (EM-Patroller), which formulates MuRP for uniform node coverage as an *unnormalized* joint steady state distribution entropy maximization problem[1], and employs multi-layer perceptron (MLP) to model the relationship between each robot's patrolling strategy and the individual steady state distribution. We iteratively update the robot team's patrolling strategies through multi-agent model-based policy gradient and show that EM-Patroller has polynomial computational complexity and guarantees to improve the multi-robot patrolling performance iteration by iteration. Additionally, EM-Patroller has the flexibility of catering to miscellaneous user-defined target joint steady state distributions, *e.g.*, selectively put emphasis on a certain area's coverage probability, as well as incorporating other objectives, *e.g.*, the *individual* steady state distribution entropy maximization, into the objective. We will verify empirically that incorporating the individual steady state distribution entropy as an auxiliary optimization objective enhances EM-Patroller's robustness performance against individual failures. We evaluate and compare EM-Patroller's performance with state of the arts in a range of canonical MuRP environments, and also demonstrate the deployment process of EM-Patroller to a real multi-robot system in self-constructed indoor environments.

The contributions of this paper can be summarized as follows: (1) EM-Patroller serves as a polynomial computational complexity algorithm with the per-iteration performance improvement guarantee; (2) EM-Patroller has the flexibility of catering to miscellaneous user-defined joint steady state distribution instead of confining itself to unnormalized joint steady state entropy maximization; and (3) EM-Patroller exhibits great robustness performance against individual robot failures when we incorporate the individual steady state distribution

Manuscript received: March 1, 2023; Revised: June 7, 2023; Accepted: July 4, 2023.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. [Note that the Editor is the Senior Editor who communicated the decision; this is not necessarily the same as the Editor-in-Chief.]

The research is supported by A*STAR with grant no. C221518004.

[1]H. Guo, Q. Kang and WY. Yau are College of Computer Science, Sichuan University (SCU), Chengdu, China. Corresponding author: Hongliang Guo.

[2]M. H. Ang is with National University of Singapore (NUS), Singapore.

[3]D. Rus is with Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA.

Digital Object Identifier (DOI): see top of this page.

[1]We will explain the rationale of selecting entropy maximization of the *unnormalized* joint steady state distribution as MuRP objective in Section III-B.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE I
BIRD'S-EYE-VIEW OF THE MURP LITERATURE

Environments			Robot Models		Objectives			Methodologies			Deployment
Continuous	[1]–[5]	2D Env. [4]–[13] [14]–[18] [19]–[24] [25]–[29]	Comm. Models	robot-to-robot (R2R) [1], [9], [28], [30]	Inherent	Minimize Idleness	Worst Idleness [1], [8]–[10], [31] [28], [32], [33]	Planning based	Offline	Global [10], [20], [24]	Real World Env. [2], [3] [7], [9], [16] [17], [32], [33]
Discrete	Grid World [8], [17], [19] [21], [24] [30], [32]–[34]			robot-to-environment (R2E) [14]–[16] [18]–[21], [35]			Mean Idleness [7], [14], [16] [21], [27], [30]			Partitioned [1]–[4] [9], [28] [7], [17], [32]	
				robot-to-center (R2C) [24], [31] [6], [34]			MFPT [17], [22], [29]				Simulation Env. [1], [31] [4]–[6], [8] [10]–[15] [18]–[30] [34], [35]
	Topology Graph [6], [7], [31] [9]–[11], [20] [14]–[16], [18] [26]–[28], [35]	3D Env. [1]–[3] [30], [32] [33]–[35]		Decentralized(No communication) [2]–[4], [10] [7], [17], [26], [32]		Approximate Frequency	Uniform Dist [11], [15], [18] [24], [27], [30]		Online [8], [34]		
							Prioritized Dist [3], [4], [20]	Learning based [14]–[16], [19], [21], [35]			
			Misc. Factors	Motion Constraint [33]		Maximize Unpredictability	Entropy of revisit [12]				
				Motion Cost [11]			Entropy Rate [13]				
				Limited Visibility [8], [17]	Intruder Oriented	Maximize Intruder Capture Probability [23], [25]		Heuristics [11], [18], [26], [27], [30], [31], [33]			
				Limited Endurance [2], [3]		Maximize Intruder Detection Count [26]					

entropy maximization into the optimization objective.

II. LITERATURE REVIEW

This section presents a brief literature review of multi-robot patrolling along the taxonomies of (1) patrolling environments, (2) robot models, (3) objectives, and (4) mainstream methodologies. Table I displays a bird's-eye-view of the MuRP literature and one is referred to [36] for comprehensive review.

1) Patrolling Environments: In MuRP, the patrolling environments can be continuous [1]–[5], or discrete. For the discrete environment, it is either represented by a grid world [8], [17], [19], [21], [24], [30], [32]–[34], or by a topological graph [6], [7], [9]–[11], [14]–[16], [18], [20], [26]–[28], [31], [35], which describes the topological relationship between different areas of the environment. Another categorization criterion is whether the patrolling environment is two-dimensional (2D) [4]–[13] or three-dimensional (3D) [1]–[3], [33]–[35].

2) Robot Models: First, the patrolling robots' communication models during online execution serve as one of the most crucial elements in fostering multi-robot collaboration for coordinated patrolling. We partition the robots' communication models into the following four categories, namely (1) robot-to-robot (R2R) communication [1], [9], [14], [28], [30]; (2) robot-to-environment (R2E) communication [14]–[16], [18]–[21], [35]; (3) robot-to-center (R2C) communication [6], [24], [31], [34] and (4) purely decentralized (no communication) [2]–[4], [7], [10], [17], [26], [32]. In the meanwhile, researchers in the MuRP domain have also considered miscellaneous factors of the robots, *e.g.*, motion characteristics [33], motion cost [11], limited visibility [8], [17], fuel or battery life constraints [2], [3], into the MuRP-related algorithm design process.

3) Patrolling Objectives: Researchers have proposed various objectives, and in this paper, we categorize these objectives into two groups: inherent MuRP objectives and intruder-oriented MuRP objectives. The inherent MuRP objectives focus on internal characteristics, such as idleness, node visitation frequency, and system unpredictability. Examples include minimizing maximal/worst idleness [1], [8]–[10], [28], [32], [33], minimizing mean idleness [7], [14], [16], [19], [21], [27], [30], and minimizing mean first-passage time (MFPT) [17], [22], [29]. Node visit frequency objectives aim to approach uniform coverage frequency [11], [15], [18], [24], [27], [30] or approximate prioritized node visit frequencies [3], [4], [20]. Unpredictability metrics include maximizing the entropy of return time (RT) [12] or the average entropy rate [13]. On the other hand, intruder-oriented MuRP objectives assess the performance of the system based on the intruder's behavior, such as maximizing capture probability within a given time budget [23], [25] or maximizing the number of intruders captured/detected within a specified time horizon [26].

4) Mainstream Methodologies: This paper characterizes the mainstream methodologies for MuRP problem into three main categories, namely planning-based methods [17], [32], learning-based methods [19], [21] and heuristics [27], [30]. (1) Planning-based methods typically formulate the MuRP problem into the mathematical optimization framework, and either incur off-the-shelf optimization solvers for offline planning solutions [1]–[4], [7], [10], [17], [20], [24], [28], [32] or receding horizon optimization tools [8], [34] for online replanning solutions. (2) Learning-based methods are deemed as recent emerging trends for MuRP solutions. They typically formulate the MuRP problem within the decentralized partially observable Markov decision process (Dec-POMDP) framework, and design the proper reward signal for each robot, so that the cumulative rewards represent (approximately) the MuRP system's overall objective. The majority of learning-based methods for MuRP target idleness-related objectives, and the reward signal is designed to reflect the instantaneous system-level idleness metric [19] or individual node-level idleness metric [14], [21]. (3) The third category of methods designs various local heuristics/rules for effective and coordinated MuRP solutions. Each robot just follows the sometimes randomized decision-making policy based on certain predefined local rules for patrolling services, and the system will exhibit emerging collective performance. In general, designing heuristics for multi-robot patrolling is an effective and robust strategy, and is easy to implement. However, the algorithms in this category cannot establish a clear relationship between the local heuristics/rules and the system-level performance metric. Therefore, it is very difficult, if not impossible, to twist the heuristics for a fresh new MuRP objective metric.

This paper targets the MuRP problem of reaching a uniform node coverage probability. Planning methods for the uniform coverage objective usually incur non-polynomial computational complexity algorithms for the ultimate optimal solution, on the other hand, most learning-based methods struggle to design an appropriate local reward signal whose accumulation resembles the system-level uniform node coverage requirement. Lastly, for heuristics, the related algorithms are usually targeting the system-level *idleness-based* metric instead of uniform coverage, and it is impossible for the heuristic-based algorithms to yield iteration-by-iteration performance improvement

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

guarantee, which is crucial in some critical MuRP application scenarios. Therefore, this paper aims at bridging the gap by proposing an efficient iterative optimization algorithm, which possesses both polynomial computational complexity and the per-iteration performance improvement guarantee.

III. Problem Formulation: Multi-Robot Patrolling for Uniform Node Coverage

In this section, we present the mathematical problem formulation of multi-robot patrolling for uniform node coverage. We introduce key concepts in the MuRP problem: individual steady state distribution, joint node coverage probability, joint node coverage frequency, and joint steady state distribution. We then formulate the MuRP problem for uniform coverage as an unnormalized joint steady state distribution entropy maximization problem, followed by an explanation of the rationale for objective selection. A list of major notations used throughout the paper can be found in Table II. Note that we define symbols when they are introduced in the main content for the first time. In this paper, bold symbols represent matrices or vectors, while non-bold symbols represent scalars or elements of a vector.

TABLE II
List of Major Notations Used in the Paper

Notations	Descriptions
N	number of patrolling robots
$\mathcal{G}(\mathcal{V},\mathcal{E})$	unit-cost graph with nodes $\mathcal{V}$ and edges $\mathcal{E}$
$\mathcal{V}$	set of nodes, $\lvert\mathcal{V}\rvert = n$
$\mathcal{E}$	set of edges, $\lvert\mathcal{E}\rvert = m$
i	robot index, $i \in \{1, 2, \ldots, N\}$
s	node index, $s \in \mathcal{V}$
$\boldsymbol{\mu}_i$	individual steady state distribution of robot i (a vector)
$\boldsymbol{\mu}$	joint node coverage probability (a vector)
$\mu(s)$	joint node coverage probability at node s
$\lambda(s)$	joint node coverage frequency at node s
π_i	robot i's decision-making policy
J	unnormalized entropy
$\boldsymbol{\theta_i}$	policy parameters of robot i (a vector)
$\boldsymbol{P}_{\boldsymbol{\theta}_i}$	(parameterized) transition probability matrix
$T_{\max}$	max. training epochs
h/d	number of hidden/input features in MLP
J_r	auxiliary object for robustness
J_s	auxiliary object for softness

A. MuRP Problem Setup

The multi-robot patrolling (MuRP) problem that we are considering in this paper is to coordinate a team of N mobile robots to persistently monitor a given discrete environment ($\mathcal{G}$) so that each node's *coverage probability* by the robot team is equal to each other, *i.e.*, uniform coverage.

The patrolling environment is modeled as an undirected and connected unit-cost graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ ($|\mathcal{V}| = n$) refers to the set of nodes and $\mathcal{E}$ ($|\mathcal{E}| = m$) refers to the set of edges. The term 'unit-cost' means that $\forall(s,s') \in \mathcal{E}$, the transition time from node s to node s' is one unit time step for any robot in the multi-robot system. The 'unit-cost' assumption of graph $\mathcal{G}$ for the MuRP problem in discrete environments is a common scheme in the MuRP domain, see [27], [30] as examples. Note that EM-Patroller takes the discretized environment as input and output the multi-robot patrolling strategy. In practice, when facing continuous environments, one may resort to Delauney triangulation [37] or occupancy grid map discretization to convert it into discrete ones. Meanwhile, we wish to point out that EM-Patroller does not deal with navigation deadlocks or spatial conflict resolutions that emerge when real robots move in constrained environments, *e.g.*, narrow corridors.

Definition 1 (Individual Steady State Distribution ($\boldsymbol{\mu}_i$))**.** The individual steady state distribution, denoted as $\boldsymbol{\mu}_i$ for robot i, is the stationary distribution of the Markov chain induced by robot i's decision-making policy π_i for Graph $\mathcal{G}$.

Note that, in the MuRP context, π_i denotes the (stochastic) decision-making policy of robot i, and it takes as input robot i's current residing node, *e.g.*, node s, and output the probability that an edge (*e.g.*, a) is chosen to execute, *i.e.*, $\mathbb{P}[a|s] = \pi_i(a|s)$. Therefore, the action space of the policy is the set of edges in Graph $\mathcal{G}$. $\mu_i(s)$ essentially refers to the probability that robot i resides in node s at any given time step, when the referred Markov chain reaches the stationary distribution. In practice, the individual steady state distribution ($\boldsymbol{\mu}_i$) are *independent* from each other when given the decision-making policy (π_i), as for a fixed Graph $\mathcal{G}$, π_i already dictates the individual steady-state distribution. To ensure a stable stationary distribution, the underlying graph $\mathcal{G}$ needs to be connected and have at least one spanning tree. With $\mu_i(s)$, we deliver the following two closely related but different concepts, namely the joint node coverage **probability** $\big(\mu(s)\big)$ and the joint node coverage **frequency** $\big(\lambda(s)\big)$.

Definition 2 (Joint Node Coverage Probability ($\boldsymbol{\mu}$))**.** The joint coverage probability of a node s, *i.e.*, $\mu(s)$, is defined as the probability that node s is visited by *any* robot at any given time step.

Definition 3 (Joint Node Coverage Frequency ($\boldsymbol{\lambda}$))**.** The joint coverage frequency of node s, *i.e.*, $\lambda(s)$, is defined as the average number of robots visiting s at any given time step.

Note that $\mu(s)$ is, in most cases, not equal to $\lambda(s)$. With the individual steady state distribution $\boldsymbol{\mu}_i$, and the assumption of independent decision making of each individual robot, one may calculate the node coverage probability for node s as:

$$\mu(s) = 1 - \prod_{i=1}^{N}(1 - \mu_i(s)), \tag{1}$$

and the node coverage frequency for node s as:

$$\lambda(s) = \sum_{i=1}^{N} \mu_i(s), \tag{2}$$

where N is the total number of patrolling robots. Note that one may think of $\lambda(s)$ as the average number of robots visiting node s at any given time step in the discrete-time Markov chain context.

Definition 4 ((Unnormalized) Joint Steady State Distribution ($\boldsymbol{\mu}$))**.** The joint steady state distribution, denoted as $\boldsymbol{\mu}$ for the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

MRS, is the **unnormalized** node coverage probability vector, with each element referring to the node's coverage probability.

In Definition 4, the term 'unnormalized' means that in most cases $\sum_{s\in\mathcal{V}}\mu(s)\neq 1$.

B. MuRP Problem Formulation

The uniform node coverage MuRP problem is formulated as the following unnormalized entropy maximization problem:

$$\underset{\pi_1,\ldots,\pi_N}{\text{maximize}} \quad J=\sum_{s\in\mathcal{V}}-\mu(s)\log\big(\mu(s)\big). \tag{3}$$

Before proceeding to the multi-agent model-based policy gradient theorem in Section IV, we first lay down the rationale of selecting entropy maximization of the unnormalized joint steady state distribution as the objective function for MuRP.

Firstly, why do we choose the node coverage **probability** $\big(\mu(s)\big)$ over the node coverage **frequency** $\big(\lambda(s)\big)$ as the core evaluation ingredient in the objective function? In most of the multi-robot patrolling scenarios, the key criterion is whether a node has been visited or not $\big(\mu(s)\big)$ within a certain time period, instead of the average visit times $\big(\lambda(s)\big)$ within that time frame. Imagine a scenario, which is a ring of 6 nodes, and we are given 6 robots starting at the same node. If all the robots are circling the ring with the same pace in the 'unit-cost' graph, we have $\forall s\in\mathcal{V}$, $\mu(s)=1/6$, in that the 6 robots 'conglomerate' together all the time, and the actual node coverage probability is $1/6$. However, $\lambda(s)=1$, which means that *on average*, each node is visited 1.0 times during a unit time frame. Apparently, for this simple use case, letting each robot 'stay' on a distinct node is the optimal solution, which results in $\mu(s)=1$. However, the solution still yields $\lambda(s)=1$, which is exactly the same as what we do to let the robots circle the ring with the same pace. Therefore, in the targeted MuRP problem, we use $\mu(s)$ instead of $\lambda(s)$ as the core evaluation ingredient.

Secondly, why do we use the rather complex (unnormalized) entropy of the joint steady state distribution to quantify the robot team's performance with the objective of uniform node coverage probability? A much more straightforward way of expressing the objective function is to minimize the variance of $\boldsymbol{\mu}$, *i.e.*, $\sum_s(\mu(s)-\bar{\mu}(s))^2$, where $\bar{\mu}(s)=\sum_s\mu(s)/n$, and n is the total number of nodes in $\mathcal{G}$. However, minimizing the variance of $\boldsymbol{\mu}$ will, sometimes, lead to a 'lazy' and meaningless solution. For example, we are given a discrete environment with n nodes, and N robots start at the robot depot, which does not count as a node. In this case, the robot team would select to stay at the depot, which results in $\mu(s)=0$ for all the nodes. It is the optimal solution, as the variance of $\boldsymbol{\mu}$ is zero. However, the solution is meaningless, in that the robot team does not surveil the environment at all. On the other hand, when we express the objective as the entropy maximization of the joint steady state distribution, it automatically drives the robots' strategies towards a uniform distribution, in that the entropy is 'peaked' at the uniform distribution. In the meanwhile, we choose the 'unnormalized' entropy expression, which tends to increase the the summation of $\mu(s)$ as well, in that when the summation of $\mu(s)$ increases, the 'unnormalized' entropy value increases accordingly. Therefore, in this paper, we formulate the mathematical optimization objective as the 'unnormalized' entropy maximization of the joint steady state distribution, as stated in Eq. (3).

IV. Methodology

This section presents EM-Patroller as an efficient solution to MuRP for uniform node coverage. We first introduce the multi-agent model-based policy gradient theorem, which serves as the core of EM-Patroller to update each robot's policy parameters, and then present EM-Patroller's pseudo code and analyze its polynomial computational complexity with the big $\mathcal{O}$ notation. The section ends with displaying three main variants of the EM-Patroller, namely (1) robust EM-Patroller, (2) variational EM-Patroller, and (3) soft EM-Patroller.

A. Multi-Agent Model-based Policy Gradient

In this subsection, we derive the gradient of the objective function in Eq. (3), with respect to the parameterized individual policies[2]. Before that, we first establish the relationship between $\boldsymbol{\mu}_i$ and $\pi_i(\boldsymbol{\theta}_i)$, where $\boldsymbol{\theta}_i\in\mathcal{R}^d$ refers to robot i's policy parameters.

Given a parameterized policy $\pi_i(\boldsymbol{\theta}_i)$, and the topological graph $\mathcal{G}$, one is able to calculate the state transition matrix of the policy-induced first-order Markov chain, and we represent the state transition matrix as $\boldsymbol{P}_{\boldsymbol{\theta}_i}$, which denotes the transition/evolution of the Markov chain, *i.e.*, $\boldsymbol{P}_{\boldsymbol{\theta}_i}$ contains the probabilities from any state s to any state s'. In this case, the individual steady state distribution $\boldsymbol{\mu}_i$ satisfies that:

$$\boldsymbol{\mu}_i^\top\boldsymbol{P}_{\boldsymbol{\theta}_i}=\boldsymbol{\mu}_i^\top. \tag{4}$$

Now, when given any $\boldsymbol{\theta}_i$, we are able to generate $\boldsymbol{P}_{\boldsymbol{\theta}_i}$, and then calculate $\boldsymbol{\mu}_i$ by solving Eq. (4) either analytically or iteratively. However, we need to calculate the gradient of $\boldsymbol{\mu}_i$ with respect to $\boldsymbol{\theta}_i$, *i.e.*, the Jacobian matrix $\partial\boldsymbol{\mu}_i/\partial\boldsymbol{\theta}_i$, which is one of the required inputs of the multi-agent model-based policy gradient.

In this paper, we treat the modeling process from $\boldsymbol{\theta}_i$ to $\boldsymbol{\mu}_i$ as a machine learning problem, and establish a multi-layer perceptron (MLP) which takes $\boldsymbol{\theta}_i$ as inputs and $\boldsymbol{\mu}_i$ as outputs, *i.e.*, $\boldsymbol{\mu}_i=\text{mlp}(\boldsymbol{\theta}_i)$. Since for any $\boldsymbol{\theta}_i$, we are able to calculate $\boldsymbol{\mu}_i$ with Eq. (4). It means that we can have as many training and testing samples as we need, and train the MLP to approximate the relationship between $\boldsymbol{\theta}_i$ and $\boldsymbol{\mu}_i$. With well-trained MLP, we can calculate the gradient of $\boldsymbol{\mu}_i$ with respect to $\boldsymbol{\theta}_i$, *i.e.*, $\partial\boldsymbol{\mu}_i/\partial\boldsymbol{\theta}_i$, with back-propagation.

With the available Jacobian matrix, *i.e.*, $\partial\boldsymbol{\mu}_i/\partial\boldsymbol{\theta}_i$, from the well-trained MLP, we present the multi-agent model-based policy gradient theorem as follows:

Theorem 1 (Multi-Agent Model-based Policy Gradient)**.** $\forall i\in\{1,2,\ldots,N\}$, we have:

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}_i}J=\sum_{s\in\mathcal{V}}(\log\mu(s)+1)(1-\mu(s))\boldsymbol{\nabla}_{\boldsymbol{\theta}_i}\log(1-\mu_i(s)).$$

The proof process of Theorem 1 is to make use of the chain rule of multivariate calculus, and the fact that

[2]Here, we use a *parameterized* policy instead of a tabular one, so that MLP can be employed to connect policy parameters with steady state distribution.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

$\forall j \neq i$, $\boldsymbol{\nabla}_{\boldsymbol{\theta}_i}\mu_j(s) = \mathbf{0}$. One may obtain the derivation process here: https://github.com/kevinkang1125/EM-Patroller/blob/main/Appendix-A.pdf.

B. Pseudo Code and Computational Complexity Analysis

With the multi-agent model-based policy gradient theorem, we deliver the training process of EM-Patroller in Algorithm 1. Note that in Algorithm 1, Line 5 actually can be replaced with the pre-trained MLP for calculating $\boldsymbol{\mu}_i$. However, MLP is, anyway, an approximated solution to the individual steady state distribution, we prefer to use it as few times as possible. Therefore, we use the canonical iterative solution scheme to calculate $\boldsymbol{\mu}_i$. Next, we use the big $\mathcal{O}$ notation to analyze the computational complexity of EM-Patroller's training process. Examining Algorithm 1, we find that the core computation happens between Line 2 and Line 12, which consists of three loops. For the first loop, Line 3 has $\mathcal{O}(1)$ complexity and Line 4 have the $\mathcal{O}(n)$ complexity. Line 5 involves solving Eq. (4), and has $\mathcal{O}(n^3)$ complexity regardless of an analytic or iterative solution strategy. In summary, the first loop has $\mathcal{O}(n^3N)$ complexity. Similar analysis can be applied to evaluate the computational complexity of the second loop, and we get $\mathcal{O}(n(N+Ndh))$, where h is the number of hidden nodes in MLP. The third loop has an $\mathcal{O}(N)$ computational complexity following the same analysis strategy. Summing up the complexity for the three loops and ignoring all terms except for the leading ones, we get EM-Patroller's computational complexity at $\mathcal{O}\big((n^3+n\times d\times h)\times N\times T_{\max}\big)$, which is polynomial with respect to number of robots (N) and scale of the graph.

Besides the polynomial computational complexity, EM-Patroller also possesses the characteristic of per-iteration performance improvement guarantee. The underlying rationale is that since the objective function (J)'s gradient has been derived with Theorem 1, when we set the learning rate α to be sufficiently small, the per-iteration update of individual policy parameters ($\boldsymbol{\theta}_i$) along J's gradient direction will increase the objective value. Furthermore, from Algorithm 1, we can see that there is no sampling process during the gradient calculation process, which means that the individual policy's parameter update is truly gradient ascent rather than *stochastic* gradient ascent. Due to the universal approximation ability of MLP, it is able to approximate the true relationship between $\boldsymbol{\mu}_i$ and $\boldsymbol{\theta}_i$ with arbitrary accuracy. Therefore, EM-Patroller possesses the per-iteration performance improvement guarantee.

Algorithm 1: Training Process of EM-Patroller

Input: Graph $\mathcal{G}$; number of robots N; pre-trained MLP for $\mathcal{G}$, *i.e.*, $\forall i \in \{1,2,\dots,N\}$, $\boldsymbol{\mu}_i = \text{mlp}(\boldsymbol{\theta}_i)$; max. training epoch: $T_{\max}$; learning rate α;

Output: Parameterized individual policy for each robot, *i.e.*, $\forall i \in \{1,2,\dots,N\}$, $\pi_i(\boldsymbol{\theta}_i)$;

Init: Randomly initialized policy parameters: $\forall i \in \{1,2,\dots,N\}$, $\boldsymbol{\theta}_i \in \mathcal{R}^d$; $t \leftarrow 0$;

1 **while** $t \leq T_{\max} - 1$ **do**
2 **foreach** $i \in \{1,2,\dots,N\}$ **do**
3 $\boldsymbol{\nabla}_{\boldsymbol{\theta}_i} J \leftarrow \mathbf{0}$;
4 Calculate $\boldsymbol{P}_{\boldsymbol{\theta}_i}$ based on $\mathcal{G}$;
5 Calculate and store the individual steady state distribution $\boldsymbol{\mu}_i$ by solving Eq. (4) either analytically or iteratively;
6 **foreach** $s \in \mathcal{V}$ **do**
7 Calculate $\mu(s)$ based on $\mu_i(s)$ through Eq. (1);
8 **foreach** $i \in \{1,2,\dots,N\}$ **do**
9 Calculate $\boldsymbol{\nabla}_{\boldsymbol{\theta}_i}\log(1-\mu_i(s))$;
10 $\boldsymbol{\nabla}_{\boldsymbol{\theta}_i} J \leftarrow \boldsymbol{\nabla}_{\boldsymbol{\theta}_i} J + (\log\mu(s)+1)(1-\mu(s))\boldsymbol{\nabla}_{\boldsymbol{\theta}_i}\log(1-\mu_i(s))$;
11 **foreach** $i \in \{1,2,\dots,N\}$ **do**
12 $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i + \alpha \cdot \boldsymbol{\nabla}_{\boldsymbol{\theta}_i} J$;
13 $t \leftarrow t+1$;
14 Final.

C. Variations of EM-Patroller

In this subsection, we briefly discuss three main variants of EM-Patroller, namely robust EM-Patroller, variational EM-Patroller and soft EM-Patroller.

1) Robust EM-Patroller: To increase the EM-Patroller's robustness against individual failures, *i.e.*, the individual robot malfunctions and completely quits the MuRP task, we augment EM-Patroller with an auxiliary objective, which targets maximizing the *individual* uniform coverage property. Specifically, we design $J_r = -\frac{1}{N}\sum_{i=1}^{N}\sum_{s\in\mathcal{V}}\mu_i(s)\log\mu_i(s)$ as the auxiliary objective, and define $J+\alpha_r J_r$, where $\alpha_r \geq 0$, as the augmented objective function, then we get the robust EM-Patroller. The rationale behind a robust EM-Patroller is that although some robots in the MRS malfunction and quit the team during execution, since each of the remaining robots is having an auxiliary uniform coverage objective, the remaining MRS will still exhibit the uniform node coverage property.

2) Variational EM-Patroller: Suppose that we have a specific multi-robot patrolling task, which has *prioritized* regions (nodes) to be covered with higher probabilities instead of the uniform node coverage. Defining the target joint steady state distribution as $\boldsymbol{\mu}'$, which is not necessarily a uniform distribution, we can formulate the new objective as minimizing the unnormalized Kullback–Leibler divergence (KL divergence) [38] from $\boldsymbol{\mu}$ to $\boldsymbol{\mu}'$, *i.e.*, $D_{\text{KL}}(\boldsymbol{\mu} \parallel \boldsymbol{\mu}')$. After derivation, we can define $\tilde{J} = -D_{\text{KL}}(\boldsymbol{\mu} \parallel \boldsymbol{\mu}') = -\sum_{s\in\mathcal{V}}\mu(s)\log(\mu(s)/\mu'(s))$ as the new objective function. Note that when we designate $\boldsymbol{\mu}'$ as the uniform distribution, variational EM-Patroller recovers to the canonical EM-Patroller.

3) Soft EM-Patroller: Another desired property for a MuRP algorithm is that it is 'unpredictable' from the observers'/intruders' perspective. We do not want the robots to regularly surveil the environment with a certain predictable pattern, in which case, the adversarial intruder will take advantage of the surveillance pattern, and attack the environment. In this paper, we gauge the unpredictability metric with the average expected entropy rate. Here, the entropy rate at node s for robot i refers to the entropy of robot i's policy at s. In this case, the auxiliary objective is defined

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

as $J_s = -\frac{1}{N}\sum_{i=1}^{N}\sum_{s\in\mathcal{V}}\left(\mu_i(s)\sum_a \pi_i(a|s)\log\pi_i(a|s)\right)$. We define $J + \alpha_s J_s$, where $\alpha_s \geq 0$, as the augmented objective function, and the resulting solution is named as a soft EM-Patroller.

V. Simulation Results and Analysis

In this section, we benchmark EM-Patroller's performance with the state of the arts in a range of canonical MuRP environments. For state of the arts, we select (1) multiple travelling salesman problem with spectral clustering (mTSP-SC) [32]; (2) DQN-Patroller [19]; (3) weighted node counting (w-NC) [27]; (4) PatrolGRAPH* [20] and (5) PatrolGRAPH^A [18]. Note that mTSP-SC, DQN-Patroller and w-NC serve as the most recent MuRP solutions along the categories of planning, learning and heuristics, respectively. We twist DQN-Patroller to fit to topological environments by enlarging the action space to contain all valid edges in Graph $\mathcal{G}$, instead of confining it to four actions in the original paper. Note that we assign negative infinity values to DQN-Patroller's state-action values, which correspond to an un-executable edge given the current residing node. In this way, the un-executable edges cannot be selected during execution.

The hyperparameters for EM-Patroller and state of the arts are configured as follows: (1) EM-Patroller[3]: $\alpha = 1\times10^{-4}$, $\alpha_r = 1.0$, $\alpha_s = 2.0$; (2) DQN-Patroller: $\alpha = 7.5\times10^{-4}$, $\gamma = 0.95$, $\epsilon = 0.93\times0.992^K$, where K is the number of episodes; (3) PatrolGRAPH*: $\sigma = 0.01$; (4) PatrolGRAPH^A: $K_1 = 1.0$, $K_2 = 10.0$, $\epsilon = 1\times10^{-8}$. Note that mTSP-SC and w-NC are free of hyperparameters. All algorithms are implemented in Python3.8 with publicly available source code[4], and all tests are executed on 8 core 16 threads 3.3 GHz CPU, 16GB RAM, NVIDIA RTX3080 and 64-bit Ubuntu system.

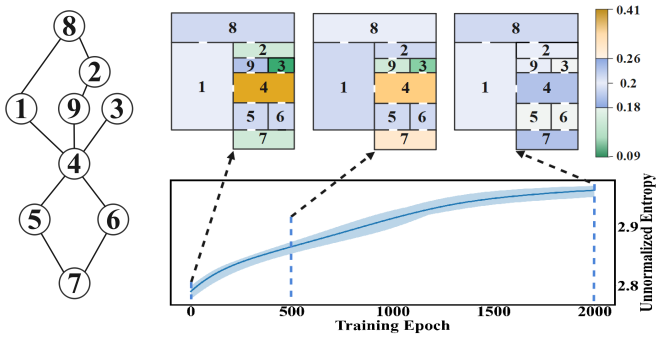
A. Performance Illustration in a Simple Environment

This subsection illustrates the per-iteration performance improvement characteristic of EM-Patroller in a simple environment, namely HOUSE, whose topology is shown in Fig. 1(a).

We configure two robots (both starts at node 4) with randomly initialized policy parameters, $\boldsymbol{\theta}_i$, and train EM-Patroller for 2000 epochs/iterations. The bottom half of Fig. 1(b) shows the evolution process of the joint steady state distribution's (unnormalized) entropy, which reflects its uniformity, and the top half of Fig. 1(b) visualizes the representative heat maps at Epoch 1, Epoch 500 and Epoch 2000, respectively. We can see that as the training process continues, the joint steady state distribution gradually approaches the uniform distribution.

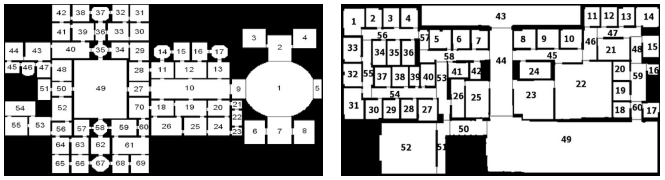
B. Performance Comparison with State of the Arts

This subsection compares EM-Patroller with state of the arts in terms of the unnormalized entropy of the joint steady state distribution, *i.e.*, the value of J in Eq. (3), in two canonical MuRP environments, namely MUSEUM and OFFICE, as visualized in Fig. 2. Note that MUSEUM and OFFICE are two canonical MuRP problem testing environments, represented by 'unit-cost' topological graphs.



(a) House Topo. (b) Entropy Evolution Process and Heat Maps

Fig. 1. Training process of EM-Patroller in 'HOUSE' Environment, with $N = 2$ robots. The shaded area in bottom half of Fig. 1(b) indicates the standard deviation for differently initialized policy parameters.



(a) MUSEUM (b) OFFICE

Fig. 2. Canonical MuRP test environments from [39], each room is associated with the corresponding node number. Left: MUSEUM; Right: OFFICE

We let the robots start at node 1 in MUSEUM and node 43 in OFFICE, and patrol the respective environment for uniform coverage. Fig. 3 shows the comparison results, where we can see that EM-Patroller achieves the best performance (gauged by the un-normalized entropy: J) in both environments for a variety of team sizes, *i.e.*, 5, 10 and 15. Note that EM-Patroller specifically maximizes the un-normalized entropy (J), while the selected state of the arts are targeting other patrolling objectives, *e.g.*, maximizing the nodes' minimal visitation frequency. It is thus unfair to make comparisons only with respect to J. Therefore, we also report the performance comparison with respect to the nodes' minimal visitation frequency in Table III. In the table, we can see that EM-Patroller achieves up-to-standard performance when compared with state of the art, *i.e.*, top three performance across the board. We conjecture that the underlying reason is maximizing the unnormalized entropy naturally disperses the robots evenly in the environment, which yields good frequency performance.

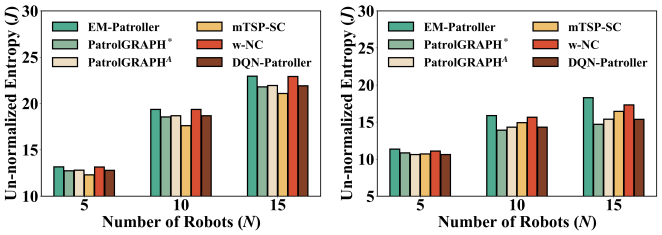
Additionally, we compare the computational efficiency between EM-Patroller and state of the arts in term of the training/planning time, and report the results in Table IV. In the table, we can see that when compared with planning-based methods, *i.e.*, PatrolGRAPH*, PatrolGRAPH^A, and learning-based methods, *i.e.*, DQN-Patroller, EM-Patroller needs much less training time. On the other hand, heuristic solutions, *i.e.*, w-NC, need virtually no training/planning time, and thus are very efficient in terms of computational complexity. However,

[3]Due to space limitations, we skip the ablation study results of EM-Patroller's hyper-parameters, *i.e.*, α, α_r, α_s, and directly display the final settings. Interested audiences are referred to the code repository for details.

[4]Code repository: https://github.com/kevinkang1125/EM-Patroller

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.



(a) MUSEUM Environment (b) OFFICE Environment

Fig. 3. Performance comparison with state of the arts in MUSEUM and OFFICE. Y-axis indicates the unnormalized entropy defined in Eq. (3) and the values are evaluated after running the corresponding algorithms for 2000 epochs

.

TABLE III
NODES' MINIMAL VISITATION FREQUENCY

	MUSEUM/OFFICE		
Algorithms	5 Robots	10 Robots	15 Robots
w-NC	**0.0440** 0.0295	**0.0890** 0.0615	**0.1275** 0.1130
EM-Patroller	**0.0369** 0.0314	**0.0658** 0.0677	**0.1157** 0.1142
PatrolGRAPH*	**0.0370** 0.0065	**0.0897** 0.0087	**0.1386** 0.0159
PatrolGRAPH^A	**0.0260** 0.0384	**0.0540** 0.0770	**0.0860** 0.1153
mTSP-SC	**0.0270** 0.0357	**0.0526** 0.0588	**0.0625** 0.0590
DQN-Patroller	**0.0258** 0.0306	**0.0526** 0.0736	**0.0789** 0.1108

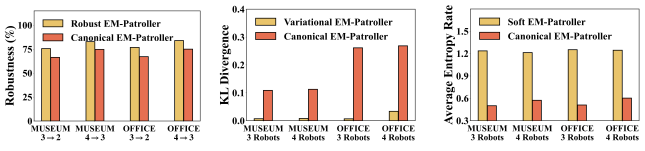
TABLE IV
TRAINING/PLANNING TIME COMPARISON

	MUSEUM/OFFICE		
Algorithms	5 Robots	10 Robots	15 Robots
w-NC	**0.7861s** 0.8011s	**1.3931s** 1.4629s	**1.9534s** 2.0925s
EM-Patroller	**2m05s** 1m47s	**4m51s** 4m25s	**8m11s** 7m10s
PatrolGRAPH*	**14m48s** 12m42s	**26m55s** 24m47s	**37m24s** 34m29s
PatrolGRAPH^A	**12m22s** 10m54s	**24m37s** 22m53s	**29m22s** 28m04s
mTSP-SC	**11.3834s** 10.3428s	**8.9652s** 7.7844s	**7.1589s** 6.1362s
DQN-Patroller	**3h09m08s** 3h15m27s	**6h40m43s** 6h19m06s	**9h47m32s** 8h54m51s

there exists no system-level performance guarantee for w-NC. Note that in Table IV, the EM-Patroller's training time does not include MLP's training time, in that we believe the training of MLP can be done independently from the multi-agent model-based policy gradient's iteration process. For the referred hardware configuration stated in the beginning of this section, and we collect $10,000$ data samples, and train for 2000 epochs (iterations), the training time of MLP for the MUSEUM and OFFICE environments are 5.42 minutes and 4.27 minutes, respectively.

C. Evaluation of EM-Patroller's Variations

This subsection evaluates the performance of robust EM-Patroller, variational EM-Patroller and soft EM-Patroller in MUSEUM and OFFICE. We gauge robustness as the percentage of the system's performance when one robot quits the team to the normal system's performance. Fig. 4(a) shows robustness comparisons between the canonical EM-Patroller and robust EM-Patroller. Fig. 4(b) compares the KL divergence values between variational EM-Patroller and canonical EM-Patroller in both environments, when we pick 4 prioritized nodes and double the corresponding importance weights. Fig. 4(c) compares the 'unpredictability' (gauged by the averaged entropy rate) between soft EM-Patroller and the canonical EM-Patroller with respect to different number of robots in both MuRP environments.

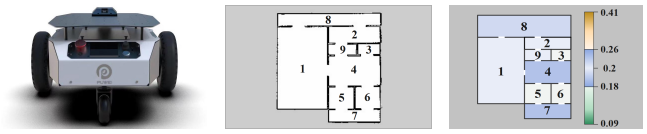


(a) Robust (b) Variational (c) Soft

Fig. 4. Performance evaluation of EM-Patroller's variations in terms of (1) robustness; (2) KL divergence; (3) un-predictability (average entropy rate).

VI. SYSTEM INTEGRATION AND EXPERIMENTAL RESULTS

This section deploys EM-Patroller to a real multi-robot system and demonstrates its functionality in a self-constructed indoor environment. The patrolling robots are DM3008 robots[5] as shown in Fig. 5(a). DM3008 is a differential drive robot, with an embedded single beam LiDAR (LDS-50C-2) for map construction and obstacle detection. The product offers the simultaneous localization and mapping (SLAM) functionality, as well as an autonomous navigation module, which navigates the robot within a pre-constructed map while avoiding obstacles.



(a) DM3008 Robot (b) Constructed Map (c) Final Heat Map

Fig. 5. The patrolling robot testbed, constructed map and the final heat map.

We integrate EM-Patroller, which functions as an intermediate sub-goal generator, to DM3008, and evaluate EM-Patroller's performance through visualizing the joint steady state distribution. The indoor environment mimics the 'HOUSE' environment, whose constructed map is shown in Fig. 5(b). We deploy two DM3008 robots for patrolling, and execute 2000 consecutive time steps. The heat map of the environment's unnormalized coverage probability are shown in Fig. 5(c), where we can see that EM-Patroller approaches, more or less, the uniform coverage of the environment. A demonstration video is uploaded with the main manuscript, and more videos are available in the github repository.

[5]More details about DM3008 are available from www.puweii.com.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

VII. Conclusion and Future Work

This paper proposes EM-Patroller for the uniform node coverage MuRP problem. EM-Patroller enjoys both the polynomial computational complexity and the per-iteration performance improvement guarantee. We compare EM-Patroller with state of the arts in two canonical MuRP environments, and also deploy it to real autonomous robot testbeds. In the future, we are keen on improving EM-Patroller to apply directly to continuous environments, in that many real-world scenarios are having space conflict resolutions and navigation deadlocks, which are not straightforward to be abstractly represented as a topological graph. In the meanwhile, we would like to deploy EM-Patroller with a larger number of robots in real world environments, and benchmark its performance in terms of the joint node coverage probability and algorithm's training time. Moreover, we would like to make use of the analytical solution to steady state distribution and derive the analytical derivative representation of the multi-agent model-based policy gradient, and thereby bypass the MLP's training process.

References

[1] J. J. Acevedo, B. Arrue, J. M. Diaz-Banez, I. Ventura, I. Maza, and A. Ollero, "Decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications," in *IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2013, pp. 565–574.

[2] D. Mitchell, M. Corah, N. Chakraborty, K. Sycara, and N. Michael, "Multi-robot long-term persistent coverage with fuel constrained robots," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1093–1099.

[3] V. Mersheeva and G. Friedrich, "Multi-UAV monitoring with priorities and limited energy resources," in *International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 25, 2015, pp. 347–355.

[4] V. Sea, A. Sugiyama, and T. Sugawara, "Frequency-based multi-agent patrolling model and its area partitioning solution method for balanced workload," in *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*. Springer, 2018, pp. 530–545.

[5] C. Banerjee, D. Datta, and A. Agarwal, "Chaotic patrol robot with frequency constraints," in *IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2015, pp. 340–344.

[6] S. Hoshino and K. Takahashi, "Dynamic partitioning strategies for multi-robot patrolling systems," *Journal of Robotics and Mechatronics (JRM)*, vol. 31, no. 4, pp. 535–545, 2019.

[7] Y. Hong, Y. Kyung, and S.-L. Kim, "A multi-robot cooperative patrolling algorithm with sharing multiple cycles," in *European Conference on Networks and Communications (EuCNC)*. IEEE, 2019, pp. 300–304.

[8] J. Scherer and B. Rinner, "Multi-robot persistent surveillance with connectivity constraints," *IEEE Access*, vol. 8, pp. 15 093–15 109, 2020.

[9] 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, 2012.

[10] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *IEEE International Conference on Intelligent Agent Technology (IAT)*. IEEE, 2004, pp. 302–308.

[11] M. Baglietto, G. Cannata, F. Capezio, and A. Sgorbissa, "Multi-robot uniform frequency coverage of significant locations in the environment," in *International Symposium on Distributed Autonomous Robotic Systems*. Springer, 2009, pp. 3–14.

[12] X. Duan, M. George, and F. Bullo, "Markov chains with maximum return time entropy for robotic surveillance," *IEEE Transactions on Automatic Control (TAC)*, vol. 65, no. 1, pp. 72–86, 2019.

[13] M. George, S. Jafarpour, and F. Bullo, "Markov chains with maximum entropy for robotic surveillance," *IEEE Transactions on Automatic Control (TAC)*, vol. 64, no. 4, pp. 1566–1580, 2018.

[14] H. Santana, G. Ramalho, V. Corruble, and B. Ratitch, "Multi-agent patrolling with reinforcement learning," in *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, vol. 4. IEEE Computer Society, 2004, pp. 1122–1129.

[15] T. Mao and L. Ray, "Frequency-based patrolling with heterogeneous agents and limited communication," *arXiv:1402.1757*, 2014.

[16] D. Portugal and R. P. Rocha, "Cooperative multi-robot patrol with Bayesian learning," *Autonomous Robots (AR)*, vol. 40, no. 5, pp. 929–953, 2016.

[17] T. Alam, M. M. Rahman, P. Carrillo, L. Bobadilla, and B. Rapp, "Stochastic multi-robot patrolling with limited visibility," *Journal of Intelligent & Robotic Systems (JINT)*, vol. 97, no. 2, pp. 411–429, 2020.

[18] G. Cannata and A. Sgorbissa, "A distributed, real-time approach to multi robot uniform frequency coverage," in *International Symposium on Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 19–32.

[19] M. Jana, L. Vachhani, and A. Sinha, "A deep reinforcement learning approach for multi-agent mobile robot patrolling," *International Journal of Intelligent Robotics and Applications (IJIRA)*, pp. 1–22, 2022.

[20] G. Cannata and A. Sgorbissa, "A minimalist algorithm for multirobot continuous coverage," *IEEE Transactions on Robotics (T-RO)*, vol. 27, no. 2, pp. 297–312, 2011.

[21] S. Y. Luis, D. G. Reina, and S. L. T. Marín, "A multiagent deep reinforcement learning approach for path planning in autonomous surface vehicles: The Ypacaraí lake patrolling case," *IEEE Access*, vol. 9, pp. 17 084–17 099, 2021.

[22] P. Agharkar, R. Patel, and F. Bullo, "Robotic surveillance and Markov chains with minimal first passage time," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2014, pp. 6603–6608.

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

[24] Y. Elmaliach, N. Agmon, and G. A. Kaminka, "Multi-robot area patrol under frequency constraints," *Annals of Mathematics and Artificial Intelligence*, vol. 57, no. 3, pp. 293–320, 2009.

[25] N. Basilico and S. Carpin, "Balancing unpredictability and coverage in adversarial patrolling settings," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2018, pp. 762–777.

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

[27] P. A. Sampaio and K. F. d. S. da Silva, "Decentralized strategies based on node marks for multi-robot patrolling on weighted graphs," in *Latin American Robotics Symposium (LARS)*. IEEE, 2019, pp. 317–322.

[28] F. Pasqualetti, J. W. Durham, and F. Bullo, "Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms," *IEEE Transactions on Robotics (T-RO)*, vol. 28, no. 5, pp. 1181–1188, 2012.

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

[30] K. S. Kappel, T. M. Cabreira, J. L. Marins, L. B. de Brisolara, and P. R. Ferreira, "Strategies for patrolling missions with multiple UAVs," *Journal of Intelligent & Robotic Systems (JINT)*, vol. 99, no. 3, pp. 499–515, 2020.

[31] A. Machado, A. Almeida, G. Ramalho, J.-D. Zucker, and A. Drogoul, "Multi-agent movement coordination in patrolling," in *International Conference on Computer and Game*, 2002, pp. 155–170.

[32] L. Collins, P. Ghassemi, E. T. Esfahani, D. Doermann, K. Dantu, and S. Chowdhury, "Scalable coverage path planning of multi-robot teams for monitoring non-convex areas," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7393–7399.

[33] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Transactions on Control Systems Technology (TCST)*, vol. 20, no. 5, pp. 1236–1251, 2011.

[34] N. Rezazadeh and S. S. Kia, "A sub-modular receding horizon approach to persistent monitoring for a group of mobile agents over an urban area," in *IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS*, vol. 52, no. 20. Elsevier, 2019, pp. 217–222.

[35] X. Zhou, W. Wang, T. Wang, Y. Lei, and F. Zhong, "Bayesian reinforcement learning for multi-robot decentralized patrolling in uncertain environments," *IEEE Transactions on Vehicular Technology (T-VT)*, vol. 68, no. 12, pp. 11 691–11 703, 2019.

[36] N. Basilico, "Recent trends in robotic patrolling," *Current Robotics Reports*, pp. 1–12, 2022.

[37] L. Guibas, D. Knuth, and M. Sharir, "Randomized incremental construction of Delaunay and Voronoi diagrams," *Algorithmica*, vol. 7, no. 1-6, pp. 381–413, 1992.

[38] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79 – 86, 1951.

[39] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *International Journal of Robotics Research (IJRR)*, vol. 28, no. 2, pp. 201–219, 2009.