Online Multi-Robot Coverage Path Planning in Dynamic Environments Through Pheromone-Based Reinforcement Learning

Kale Champagnie, Boli Chen, *Senior Member, IEEE*, Farshad Arvin, *Senior Member, IEEE*, and Junyan Hu, *Senior Member, IEEE*

***Abstract*— Two promising approaches to coverage path planning are reward-based and pheromone-based methods. Reward-based methods allow heuristics to be learned automatically, often yielding a superior performance to hand-crafted rules. On the other hand, pheromone-based methods leverage stimgergy to achieve superior generalization and adaptation in unknown or nonstationary environments. To obtain the best of both worlds, we introduce Greedy Entropy Maximization (GEM), a hybrid approach that aims to maximize the entropy of a pheromone deposited by a swarm of homogeneous ant-like agents. We begin by establishing a sharp upper-bound on achievable entropy and show that this corresponds to optimal dynamic coverage path planning. Next, we demonstrate that GEM closely approaches this upper-bound despite depriving agents of typical necessities such as memory and explicit communication. Finally, we show that GEM can be executed asynchronously in constant-time through distillation into a shallow neural network, making our approach highly scalable.**

I. Introduction

Coverage Path Planning (CPP) is the task of directing one or more mobile agents such that they collectively explore the entirety of a given area [1]. An example is illustrated in Fig. 1, where four robots are coordinated to cover an area with obstacles collaboratively. CPP has numerous applications across domains such as search and rescue [2], exploration of hazardous environments [3], robot-insect interaction [4], aerial surveillance [5], [6], and autonomous driving [7].

CPP approaches can be categorized along various axes, such as whether they operate in stationary or nonstationary environments, or whether they make use of discrete or continuous action-spaces. This work concerns two classes of approach known as *reward-based* and *pheromone-based* respectively. In reward-based approaches, task-specific desiderata are represented by a scalar-valued reward function that is subsequently maximized using reinforcement learning or other mathematical optimization strategies [8]. A major advantage of reward-based approaches is that they enable decision heuristics to be inferred automatically through reinforcement learning rather than crafted by hand, often leading to superior performance [9]. Orthogonally, pheromone-based approaches take inspiration from natural swarms such as ant colonies, wherein successful coverage path planning emerges from localized interactions between simplistic agents [10], [11]. Additionally, pheromone-based approaches typically make *very few* assumptions about the environment dynamics, paving the way for superior generalization and adaptation to unknown environments [12].

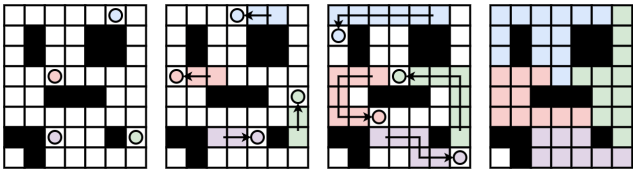

Fig. 1. Cooperative coverage of four robots in a cluttered environment.

While both reward- and pheromone-based approaches have respective merits, to the best of our knowledge, very few works aim to combine them. In this work, we affirm that combining these approaches through *learned pheromone-based policies* is highly effective and obtains the best of both worlds. Our proposed method, known as Greedy Entropy Maximization (GEM), is based on the observation that maximizing the entropy of pheromone deposited within an environment is equivalent to optimal dynamic coverage path planning. We find that with only local pheromone information, a swarm of homogeneous ant-like agents can rapidly increase the entropy, leading to highly efficient coverage. Most notably, we find that GEM closely approaches the upper-bound on entropy maximization performance, bringing it within proximity to an optimal policy. Our training approach is two-fold. Firstly, we pretrain GEM policies through behaviour cloning to approximate the behavior of an expert hand-crafted policy known as Argmin. Secondly, we fine-tune pretrained policies on an entropy maximization objective to learn further refinements. The resulting policies empirically demonstrate strong generalization across environment configurations that were not observed during training.

In summary, our contributions are as follows:

1) We introduce entropy maximization as a generalized dynamic coverage path planning objective suitable for learning pheromone-based policies.
2) We develop GEM policies that leverage a *pretrain then fine-tune* training strategy to obtain strong performance.
3) We conduct empirical investigations into GEM's performance under diverse environment configurations and

This work was supported by EU H2020-FET-OPEN RoboRoyale project [grant number 964492].

K. Champagnie is with the Department of Computer Science, University College London, London, UK. (e-mail: kale.champagnie.20@ucl.ac.uk)

B. Chen is with the Department of Electrical and Electronic Engineering, University College London, London, UK. (e-mail: boli.chen@ucl.ac.uk)

F. Arvin and J. Hu are with the Department of Computer Science, Durham University, Durham, UK. (e-mail: {farshad.arvin, junyan.hu}@durham.ac.uk)

demonstrate evidence of strong generalization.

II. Related Work

Prior work has established reward-based and pheromone-based coverage path planning approaches as highly successful. One line of work instigated by Predator-Prey Coverage Path Planning (PPCPP) [13] introduces a multi-faceted reward function based on the concepts of foraging and risk-of-predation in predator-prey relationships. The proposed reward function encourages agents to explore uncovered regions (foraging) while maximizing their distance to a set of virtual predator points. The predators cause agents to collectively avoid revisiting already covered regions. Subsequent approaches building on PPCPP include Dec-PPCPP [14] for decentralized execution and DH-CPP [15], where the method is extended to handle unbounded surface area environments. An alternative line of work is based on incremental expansion of agent regions. For instance, [16] train a decentralized multi-agent policy with a reward that strongly penalizes overlap between individual agent regions. Once coverage regions have been established, traditional spanning-tree methods [17] are used to obtain explicit trajectories.

Regarding pheromone-based approaches, a seminal work is Stigmergic Coverage (StiCo) [18]. In this scheme, a team of ant-like agents traverse continuous circular paths that encompass their individual coverage regions. Each agent deposits pheromone along the circumference of its region while also aiming to avoid traversing pheromone deposited by others. Collectively, this causes an initially dense packing of agents to disperse and cover the whole area of interest. Following this, BeePCo [19] takes inspiration from the dynamics between queen and worker bees to develop a complementary method to StiCo. Finally, HybaCo [20] introduces a hybrid ant-and-bee approach, effectively composing aspects of StiCo and BeePCo to compensate for their individual limitations. A key limitation of StiCo is that it does not specify how agents should traverse the interior of their individual coverage regions. While it provides an efficient decomposition of the target area, it does not provide the *explicit coverage trajectories* necessary in many applications [21]. Motivated by bio-inspired learning techniques [22], we aim to address this limitation in our work by utilizing the advantages of both sides.

III. Entropy Maximization

In this section, we introduce entropy maximization as a generalized coverage path planning objective and provide a corresponding reward function.

The goal of entropy maximization is to maximize the entropy of a pheromone deposited by a swarm of ant-like agents within an environment. Entropy in this case refers to the Shannon entropy of the normalized pheromone distribution. Since we allow the deposited pheromone to decay over time, regions that do not receive frequent updates will cause the pheromone distribution to become more concentrated and take a lower entropy value. Redundant visitation and unnecessary overlap between agent regions also induce a similar effect. Indeed, we find that many conventional coverage path planning desiderata are naturally optimized for by maximizing entropy. To this end, we consider entropy maximization a promising objective for training pheromone-based policies.

A. Partially Observable Markov Decision Process

In entropy maximization, we formulate the environment as a Partially Observable Markov Decision Process (POMDP) (S, A, P_a, R_a), where S is the state-space, A is the action-space, $P_a(\mathbf{s}_{t-1} \to \mathbf{s}_t)$ is a dynamics model, and $R_a(\mathbf{s}_{t-1} \to \mathbf{s}_t)$ is a reward model. A policy π is a deterministic or stochastic function that maps states to actions, i.e., for every state $\mathbf{s}_t \in S$, it produces a corresponding action $a_t \in A$.

Concretely, each state $\mathbf{s}_t$ holds a distribution of pheromone over the points of a lattice $G = (V, E)$. Each point $v \in V$ represents a location that an agent may exclusively occupy. Each edge $e \in V$ represents a path that an agent may travel along in order to move between points. Hence, the action space for n agents is the set of all n-sized subsets of E. Under this formulation, obstacles and prohibited actions can be implemented by removing subsets of edges in E.

While we assume P_a is unknown due to partial observability, we explicitly define the reward model R_a as follows. Firstly, whenever an agent moves, we assume that it deposits one unit of pheromone to the point it previously occupied. Then, representing the pheromone distribution as a vector, we have that

$$\mathbf{s}_t = (1 - \alpha)\mathbf{s}_{t-1} + \mathbf{o}_{t-1}, \tag{1}$$

where $\alpha \in (0, 1)$ is the *pheromone decay rate*, and $\mathbf{o}_{t-1}$ is a one-hot vector containing 1s at the indices of previously occupied points. For brevity, we denote the *normalized* pheromone distribution by

$$\mathbf{u}_t = \frac{\mathbf{s}_t - \min \mathbf{s}_t}{\max \mathbf{s}_t - \min \mathbf{s}_t}. \tag{2}$$

Under this formulation, we define the entropy maximization reward as follows:

$$R_{EM}(\mathbf{s}_{t-1} \to \mathbf{s}_t) = \mathbb{H}[\mathbf{u}_t] - \mathbb{H}[\mathbf{u}_{t-1}], \tag{3}$$

where $\mathbb{H}[\cdot]$ is the Shannon entropy measured in nats. Intuitively, this is the difference in entropy between successive time steps. The reward is positive if the entropy increases and negative otherwise. Accordingly, the entropy maximization objective can be formulated as the expected cumulative reward obtained by rolling out a given policy π. This is given by,

$$V^{\pi} = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^t R_{\pi(\mathbf{s}_{t-1})}(\mathbf{s}_{t-1} \to \mathbf{s}_t)\right], \tag{4}$$

where γ is the discount rate.

Interestingly, we find that low-value policies exhibit conventionally undesirable traits and thus cause low entropy, while high-value policies will lead to a better coverage performance, as shown in Fig. 2.

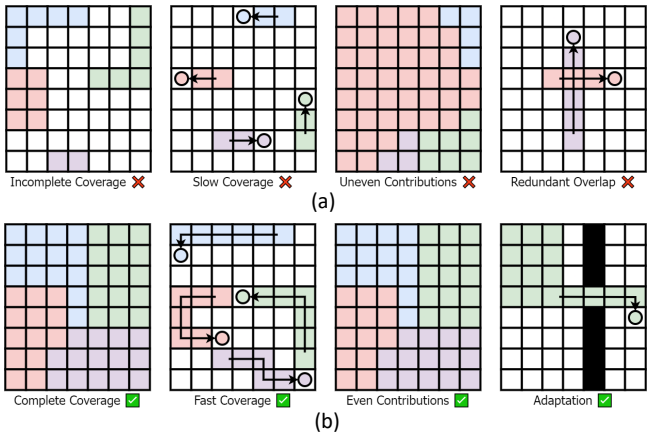

Fig. 2. Behavioural characteristics of (a) low-value and (b) high-value policies under entropy maximization.

B. Entropy Maximization Upper-bound

An advantage of formulating coverage path planning in terms of entropy maximization is that it enables us to derive a sharp upper bound on coverage path planning performance. With such a bound, we can make quantitative statements about the extent to which candidate policies are optimal.

According to our reward model, the reward accrued at each time step is the *change in entropy* of the pheromone distribution.

Theorem 1: Let π be a policy with $n \geq 1$ participating agents. Then, the expected cumulative reward at time t is *at most* $\log(nt)$.

Intuitively, a reward of $\log(nt)$ is obtained if, at every time step including t, the policy uses n agents to visit n *fresh points* while maintaining a perfectly uniform distribution over all visited points. The extent to which any policy can achieve this reward depends on the pheromone decay rate α. For low decay rates, any policy that avoids revisiting points for t steps will achieve an expected cumulative reward of approximately $\log(nt)$. However, for high decay rates, even a policy that avoids revisitation may struggle to maintain such high entropy. In this sense, the decay rate regulates the relative *difficulty* of the CPP task. In our experiments, we consider decay rates of $\alpha = 0.01$ (standard) and $\alpha = 0.1$ (hard). Ideally, we ought to use an upper bound that is dependent on the chosen decay rate. Unfortunately, this is highly nontrivial as the pheromone distribution at time t depends intricately on the particular sequence of actions taken in prior states. Nonetheless, the present upper bound is sufficient to make quantitative claims about the optimality of a given policy.

To motivate our use of this bound, we provide the following analysis.

Let π be a policy with $n \geq 1$ participating agents. Let $G = (V, E)$ be the environment lattice. Let $O_t \in V$ denote the subset of points that have ever been visited at time t.

Since the pheromone distribution is uniformly zero over unvisited points, the entropy at time t is only variant to the pheromone distribution over visited points. The maximum entropy distribution is a uniform distribution over O_t. Then,

$$\mathbb{H}[\mathbf{u}_t] \leq \log |O_t|. \tag{5}$$

With agents n, the maximum number of points visited at time t is nt. Then,

$$\mathbb{H}[\mathbf{u}_t] \leq \log(nt). \tag{6}$$

The maximum possible reward at time t occurs when the entropy jumps from zero to $\max \mathbb{H}[\mathbf{u}_t]$. Therefore, it is upper-bounded by $\log(nt)$ and thus the proof is completed.

IV. Greedy Entropy Maximization

Greedy Entropy Maximization (GEM) is a hybrid reward- and pheromone-based method for coverage path planning. Concretely, GEM aims to learn a pheromone-based policy $\pi(\cdot; \theta)$ that determines each agent's next move based only on adjacent pheromone levels. We use a stochastic policy function such that it returns a *distribution* over possible actions $\{\text{NORTH}, \text{EAST}, \text{SOUTH}, \text{WEST}\}$. A complete coverage path plan can then be simulated by rolling-out $\pi(\cdot; \theta)$.

A. Asynchronous Execution

An important practical consideration is the *time complexity* associated with the execution of $\pi(\cdot; \theta)$ for n agents. In theory, constant time complexity can be achieved if the policy is executed asynchronously by all agents in parallel. This implies that agents must move simultaneously *without waiting for each other*. In practice, this presents numerous difficulties such as race conditions that lead to collisions with agents or dynamic obstacles. To overcome this problem, we utilize a *pretrain then fine-tune* training strategy that adds a collision avoidance term to the regular entropy maximization reward function. This is given by

$$R_a(\mathbf{s}_{t-1} \to \mathbf{s}_t) = R_{\text{EM}}(\mathbf{s}_{t-1} \to \mathbf{s}_t) + \beta R_{\text{CA}}(\mathbf{s}_{t-1} \to \mathbf{s}_t), \tag{7}$$

where $R_{\text{CA}}(\cdot)$ returns a value of -1 on colliding actions and 0 otherwise. We set β to a high value (i.e. 100) in order to strongly penalize collisions.

B. Student Policy Architecture

We utilize a lightweight multilayer perceptron (MLP) to parameterize the policy function $\pi(\cdot; \theta)$ (shown in Fig. 3). We observe only minor gains in performance from utilizing a more sophisticated convolutional architecture, presumably due to the small spatial dimension of each agent's local field of view. However, we suspect that policies requiring inter-agent communication could motivate more expressive models such as attention-based transformer networks [23].

C. Argmin Pretraining

Our most successful GEM policies were pretrained through behavior cloning to approximate the behaviour of an expert policy known as *Argmin* (illustrated in Fig. 4). Argmin is exceptionally simple, yet highly effective. In short, it instructs every agent to move to the whichever adjacent point has the least pheromone deposited there. Despite its

synchronous execution model, we find that combining behavior cloning with collision avoidance is sufficient to distill Argmin into an asynchronous neural network policy.

After pretraining, we further fine-tune GEM policies on the entropy maximization and collision avoidance rewards given in equation (7). While Argmin is highly effective early in coverage, it frequently exhibits slow convergence once a large amount of pheromone has been deposited. Subsequent fine-tuning appears to learn *adjustments* that compensate for such failure modes.

In both pretraining and fine-tuning, we use multi-agent reinforcement learning with Proximal Policy Optimization (PPO) [24]. We found PPO to be substantially more effective than alternative algorithms such as deep Q-learning.

D. Argmin Pseudocode

Let O_t be the set of locations occupied by agents at time t, and let $\mathbf{o}_t$ be a one-hot representation of O_t, i.e., $\mathbf{o}_t[i] = 1 \iff i \in O_t$. Let $\text{adj}(i)$ denote the set of locations adjacent to location i. Finally, as before, let $\mathbf{s}_t$ denote the unnormalized pheromone distribution at time t. In this case, Argmin is executed as specified by Algorithm 1.

Algorithm 1 Argmin Policy Execution

for $t \in \{0, \ldots, t_{\text{max}}\}$ **do**
  for $i \in O_t$ **do**
    $O^i_{t+1} = \text{argmin}_{j \in \text{adj}(O^i_t)} \mathbf{s}_t[j]$
  end for
  $\mathbf{s}_{t+1} = (1 - \alpha)\mathbf{s}_t + \mathbf{o}_t$
end for

V. Results and Evaluation

In this section, we present results that empirically validate GEM's performance in diverse simulated environments. Our methodology revolves around varying key factors such as obstacle density, obstacle stochasticity, number of agents, and map size, while observing changes to performance.

A. Performance Metrics

To distill the performance into a single scalar value, we measure the *mean difference* between the upper bound of the entropy and the empirical entropy achieved by GEM over the course of a run. Since entropy is logarithmic, we prefer to measure the *mean difference in perplexity*. We refer to this metric as the "P-score", given by

$$\mathcal{P} = 1 - \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\exp(R^*_t) - \exp(R_t)}{\exp(R^*_t)}\right), \qquad (8)$$

where R^*_t denotes the reward obtained by an optimal policy at time t. This is just the entropy upper-bound $\log(nt)$. An optimal policy achieves a P-score of 1.0 since at all times, it matches the upper-bound perplexity.

We plot the cumulative reward of GEM, GEM* and the entropy maximization upper-bound averaged over 1000 runs. GEM* depicts the hypothetical reward that would be achieved if the pheromone was distributed *perfectly uniformly* over all points visited by GEM.

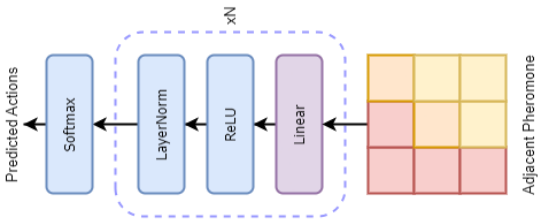

Fig. 3. Student Policy Architecture.

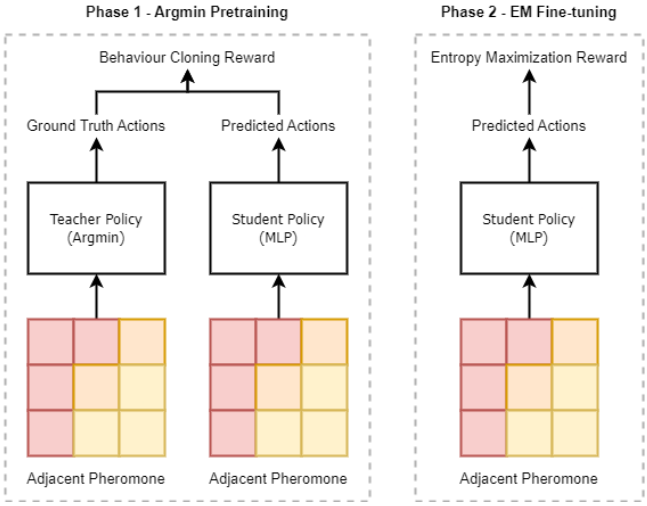

Fig. 4. Our *pretrain then fine-tune* training strategy.

Aside from $\mathcal{P}$, we use the following notation for other variables of interest. We denote by $D \in [0, 1)$, the obstacle density, i.e., the probability that a randomly selected point in the environment lattice is occupied by an obstacle. We denote by $S \in [0, 1)$, the obstacle stochasticity, i.e. the probability that a particular obstacle will move at time t. In our experiments, we simulate movement through Brownian motion. We denote by $n \in \mathbb{N}$, the number of agents participating in coverage. We denote by $A \in \mathbb{N}$, the size of the map (e.g. 16 for a 16x16 map). Finally, we denote by N, the number of times a particular experiment was run. Generally, we use $N = 1000$.

B. Obstacle Density

How do GEM policies be trained in low obstacle density environments (below 1%) perform in higher density settings? Remarkably, we find that policies trained without any obstacles can nonetheless achieve high coverage in their presence (e.g., Fig. 5). This holds for densities medium to high densities (5% to 20% respectively). At obstacle densities higher than 20% (extreme), a substantial proportion of points are entirely inaccessible (i.e. blocked by obstacles on all sides). In this case, no CPP policy may succeed in covering them. Moreover, many agents experience entrapment (i.e., confinement to small portions of the map), leading to substantially reduced coverage capacity. Under these circumstances, GEM

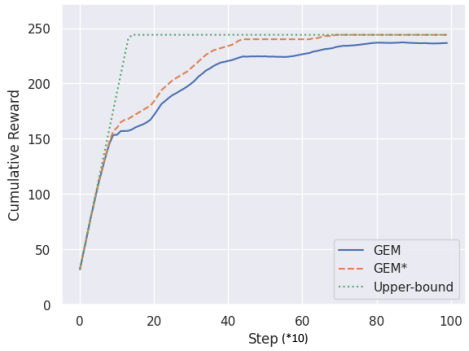

Fig. 5. Cumulative reward for GEM vs the upper-bound. $A = 16, n = 16, N = 1000, \mathcal{P} = 0.90$.

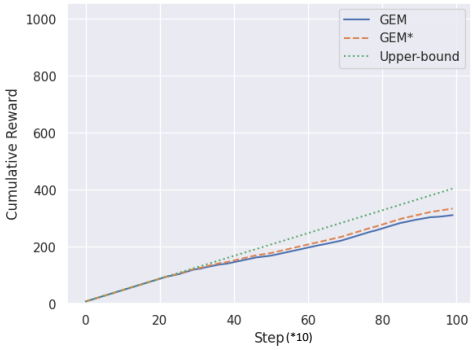

Fig. 6. Cumulative reward for GEM vs the upper-bound. $A = 32, n = 4, N = 1000, \mathcal{P} = 0.87$.

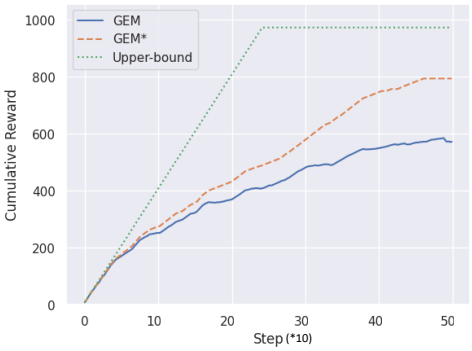

Fig. 7. Cumulative reward for GEM vs the upper-bound (500 steps). $A = 32, n = 4, N = 1000, \mathcal{P} = 0.61$.

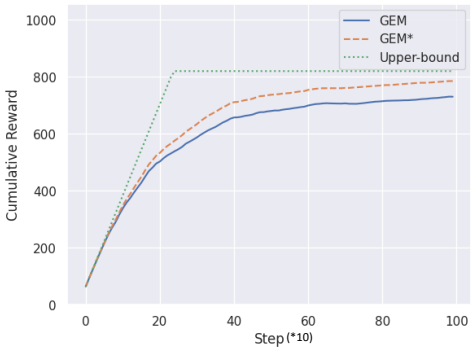

Fig. 8. Cumulative reward for GEM vs the upper-bound. $A = 32, n = 32, N = 1000, \mathcal{P} = 0.84$.

performs poorly. Nonetheless, we observe in GEM, a striking ability to *retain* high coverage once such extreme conditions have subsided. Table I compares P-score against obstacle density with $\alpha = 0.01, A = 16, n = 16, S = 0$ and $N = 1000$.

C. Obstacle Stochasticity

Obstacle stochasticity represents the rate at which obstacles move. In our experiments, we simulate each obstacle's movement through Brownian motion. The stochasticity explicitly corresponds to the probability that an obstacle will change location (using a Brownian motion) at time t. Future work may consider more sophisticated dynamics models. In general, we find that increasing stochasticity does not substantially degrade coverage performance in high obstacle density environments. On the contrary, in extreme obstacle density environments, increased stochasticity reduces the risk of long-term entrapment or inaccessible points, thereby improving coverage. We note that this effect may not be seen in alternative obstacle dynamics models. Table II compares P-score against obstacle stochasticity with $\alpha = 0.01, A = 16, n = 16, D = 0.2$, and $N = 1000$.

D. Map Size

Although we conduct training on 16x16 maps, we find that GEM policies remain effective in larger settings given sufficient agents (namely $n = A$), as illustrated in Fig. 8. We may attribute this to the fact that GEM policies only consider a small field of view around each agent. In this case, we may expect only a marginal change in how these localized features are distributed on small versus large maps. Table III compares P-score against map size with $\alpha = 0.01, n = A, D = 0.05, S = 0$ and $N = 1000$. Similarly, Fig. 5 and Fig. 8 demonstrate generalization across multiple map sizes (i.e. 16 and 32).

E. Number of Agents

Regarding the number of participating agents, we observe that GEM generally permits the use of only $n = A$ agents to achieve high coverage (i.e. $\mathcal{P} > 0.80$), with a pheromone decay rate of $\alpha = 0.99$. In other words, we may use n agents to cover n^2 points efficiently. With substantially fewer agents (i.e. $n < \sqrt{A}$), we find that GEM cannot match the upper-bound at this decay rate except during the first few steps of coverage as illustrated in Fig. 7. This can be attributed to *premature revisitation*, wherein already visited points with low pheromone are more likely to be observed than very distant unvisited points, causing revisitation. A simple remedy is to reduce the pheromone decay rate such that agents are less likely to encounter visited points that have exceptionally low pheromone. However, we find that this approach has diminishing returns when the ratio between n and A is below 0.1%. Table IV compares P-score against number of agents with $\alpha = 0.01, A = 16, D = 0.05, S = 0$ and $N = 1000$.

TABLE I

OBSTACLE DENSITY VS P-SCORE.

Obstacle Density D	P-score $\mathcal{P}$
0.05	0.88
0.10	0.86
0.20	0.85
0.50	0.23

TABLE II

OBSTACLE STOCHASTICITY VS P-SCORE.

Obstacle Stochasticity S	P-score $\mathcal{P}$
0.05	0.85
0.10	0.86
0.30	0.84
0.50	0.81

TABLE III

MAP SIZE VS P-SCORE.

Map size A	P-score $\mathcal{P}$
16	0.88
32	0.87
64	0.86

TABLE IV

NUMBER OF AGENTS VS P-SCORE.

Number of agents n	P-score $\mathcal{P}$
4	0.84
16	0.88
32	0.89

VI. CONCLUSION

In this work, we introduce GEM, a novel hybrid reward- and pheromone-based approach to coverage path planning that incorporates the advantages of both paradigms. We establish an upper-bound on entropy maximization performance and find empirically that GEM closely approaches it in diverse environments. Moreover, our experimental results demonstrate several instances of generalization to unseen environment configurations, which we attribute to the localized nature of pheromone-based communication.

REFERENCES

[1] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad, "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms," *IEEE Access*, vol. 9, pp. 119 310–119 342, 2021.

[2] F. Rekabi-Bana, J. Hu, T. Krajník, and F. Arvin, "Unified robust path planning and optimal trajectory generation for efficient 3D area coverage of quadrotor UAVs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 2492–2507, 2024.

[3] A. Henshall and S. Karaman, "Generalized multiagent reinforcement learning for coverage path planning in unknown, dynamic, and hazardous environments," in *AIAA SCITECH 2024 Forum*, 2024, p. 2762.

[4] F. Rekabi-Bana, M. Stefanec, J. Ulrich *et al.*, "Mechatronic design for multi robots-insect swarms interactions," in *2023 IEEE International Conference on Mechatronics*, 2023, pp. 1–6.

[5] K. Wu, J. Hu, Z. Li, Z. Ding, and F. Arvin, "Distributed collision-free bearing coordination of multi-UAV systems with actuator faults and time delays," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[6] J. Chen, F. Ling, Y. Zhang, T. You, Y. Liu, and X. Du, "Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system," *Swarm and Evolutionary Computation*, vol. 69, p. 101005, 2022.

[7] S. Xie, J. Hu, Z. Ding, and F. Arvin, "Cooperative adaptive cruise control for connected autonomous vehicles using spring damping energy model," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 2974–2987, 2023.

[8] R. Shivgan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *IEEE International Conference on High Performance Switching and Routing*, 2020, pp. 1–6.

[9] B. Ai, M. Jia, H. Xu, J. Xu, Z. Wen, B. Li, and D. Zhang, "Coverage path planning for maritime search and rescue using reinforcement learning," *Ocean Engineering*, vol. 241, p. 110098, 2021.

[10] X. Cheng, R. Jiang, H. Sang, G. Li, and B. He, "Trace pheromone-based energy-efficient uav dynamic coverage using deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, 2024.

[11] V. P. Tran, M. A. Garratt, K. Kasmarik, and S. G. Anavatti, "Dynamic frontier-led swarming: Multi-robot repeated coverage in dynamic environments," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 3, pp. 646–661, 2023.

[12] M. Hassan and D. Liu, "Ppcpp: A predator–prey-based approach to adaptive coverage path planning," *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 284–301, 2020.

[13] M. Hassan, D. Mustafic, and D. Liu, "Dec-ppcpp: A decentralized predator–prey-based approach to adaptive coverage path planning amid moving obstacles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 11 732–11 739.

[14] J. Zhang, P. Zu, K. Liu, and M. Zhou, "A herd-foraging-based approach to adaptive coverage path planning in dual environments," *IEEE Transactions on Cybernetics*, 2023.

[15] Y. Liu, J. Hu, and W. Dong, "Decentralized coverage path planning with reinforcement learning and dual guidance," *arXiv preprint arXiv:2210.07514*, 2022.

[16] J. Lu, B. Zeng, J. Tang, and T. L. Lam, "Tmstc*: A turn-minimizing algorithm for multi-robot coverage path planning," *arXiv preprint arXiv:2212.02231*, 2022.

[17] A. Jonnarth, J. Zhao, and M. Felsberg, "Learning coverage paths in unknown environments with reinforcement learning," 2023.

[18] I. Caliskanelli, B. Broecker, and K. Tuyls, "Multi-robot coverage: A bee pheromone signalling approach," in *First International Symposium on Artificial Life and Intelligent Agents*. Springer, 2015, pp. 124–140.

[19] B. Broecker, I. Caliskanelli, K. Tuyls, E. I. Sklar, and D. Hennes, "Hybrid insect-inspired multi-robot coverage in complex environments," in *Towards Autonomous Robotic Systems*, 2015, pp. 56–68.

[20] X. Huang, M. Sun, H. Zhou, and S. Liu, "A multi-robot coverage path planning algorithm for the environment with multiple land cover types," *IEEE Access*, vol. 8, pp. 198 101–198 117, 2020.

[21] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "Uav coverage path planning under varying power constraints using deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 1444–1449.

[22] S. Na, T. Rouček, J. Ulrich, J. Pikman, T. s Krajník, B. Lennox, and F. Arvin, "Federated reinforcement learning for collective navigation of robotic swarms," *IEEE Transactions on cognitive and developmental systems*, 2023.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.