2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Decentralized Partially Observable Markov Decision Process for Dynamic Obstacle Avoidance and Complete Area Coverage using Multiple Reconfigurable Robots

J.J.J. Pey, S. M. Bhagya P. Samarakoon, M. A. Viraj J. Muthugala, and Mohan Rajesh Elara

Abstract— **Achieving complete area coverage in robotics is an essential aspect for applications such as cleaning and patrolling. While multi-agent frameworks have been implemented to address the challenge of complete coverage, the area coverage performances are hindered by physical constraints and dynamic obstacles that cause inaccessibility to certain areas of the environment. Reconfigurable robots have been adopted to mitigate this issue as the independent alteration of the morphologies during deployments enables overcoming tight spaces to access obstructed areas. Hence, this paper proposes a Multi-Agent Reinforcement Learning (MARL) framework leveraging the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) to enable a team of reconfigurable robots to achieve complete coverage under the presence of dynamic obstacles. The framework is modelled to allow the robots to coordinate and plan their motions effectively while using shape adaptability to access narrow spaces while avoiding dynamic obstacles. Experimental results demonstrated the framework's ability to be generalised even when scaled up to a different number of agents across larger environments.**

I. Introduction

Robots are essential in simplifying a plethora of essential daily applications such as cleaning [1], maintenance [2], [3], auditing [4], and patrolling [5] to improve productivity and efficiency. Within robotics, one major challenge is the Complete Coverage Planning (CCP) in unknown environments with dynamic obstacles where robots determine a path from an initial state to a final state to ensure thorough coverage of all areas in the environment [6]. In recent years, multi-robot systems have been adopted for CCP due to significant improvements in robustness, reduced operational time and energy as compared to single robots [7]. Single robots are also more vulnerable to malfunction and failure, which would render the deployment unsuccessful and unproductive [8].

In the current state-of-the-art, coverage and path planning within multi-agent settings are achieved through either classical or heuristic-based algorithmic approaches. Under the classical methods, [9], [10], and [11] documented the usage of modified cell decomposition and sampling-based approaches for complete coverage when robots have limited range or communication. While the approaches are effective for high-dimensional environments, the methods assume a full state of the environment prior to deployment and also struggle with the presence of dynamic obstacles.

In contrast, heuristic-based methods focus on evolutionary and human-inspired algorithms such as swarm intelligence, Reinforcement Learning (RL) and neural networks [12]. The work [13], explored implementing multi-robot exploration algorithms with Grey Wolf Optimisation for distributing search assignments while [14] further discussed the demonstration of a modified Bee Colony Optimisation algorithm maximising area coverage in a distributed manner, but both approaches do not always achieve complete coverage. Amongst RL approaches, [15] demonstrated an integrated RL and cell decomposition algorithm by creating the coverage path through recursively solving cells modelled as a Travelling Salesman Problem (TSP). [16] utilised an energy and time based RL approach with TSP while [17] presented a tabular temporal difference learning approach for online CCP. Additionally, [18] proposed a dynamic area coverage RL algorithm and a γ-information map to transform continuous dynamic coverage process into a γ point traversal process for no-hole coverage. However, these approaches are limited to single-agent settings, fixed starting positions, and also require a fully observable state.

From the prior literature, the research focuses on fixed morphology robots that do not enable flexible alteration of morphology during real-time deployments in narrow and tight spaces for complete coverage. These limitations hinder the performance of area coverage and productivity of deployment as not all areas can be accessed by the robots [19]. In contrast, reconfigurable robots have the ability to change their shape per the requirements [20]. The ability to change its shape according to environmental constraints allows the robots to access narrow spaces, thereby improving the coverage performance. Hence, reconfigurable robots have been introduced to overcome the limitations of fixed-shape robots in area coverage applications [21], [22]. Several research aspects on CCP have been explored in these reconfigurable robots, such as energy efficient area coverage [23] and risk-aware path planning [24]. Although reconfigurable robots can access narrow spaces and perform area coverage, it is difficult for a single robot to cover when the environment increases. Therefore, multiple robots should be present to cover the area. However, according to the literature, multi-robot area coverage using reconfigurable robots has not been studied.

This research is supported by the National Robotics Programme under its National Robotics Programme (NRP) BAU, Ermine III: Deployable Reconfigurable Robots, Award No. M22NBK0054, A*STAR under its "RIE2025 IAF-PP Advanced ROS2-native Platform Technologies for Cross sectorial Robotics Adoption (M21K1a0104)" programme, and also supported by SUTD Growth Plan (SGP) Grant, Grant Ref. No. PIE-SGP-DZ-2023-01.

The authors are with the Engineering Product Development Pillar, Singapore University of Technology and Design, 8 Somapah Rd, Singapore 487372.

Both area coverage and coverage time could be improved using multiple reconfigurable robots.

As such, in this paper, a Multi-Agent Reinforcement Learning (MARL) framework is proposed for reconfigurable robots to achieve CCP and also dynamic obstacle avoidance. Given that it would not be practical to consider the full state of the environment, the framework also considers partial observations and random initialisation of positions for each robot. This approach is a potential opportunity to enable reconfigurable robots to coordinate actions through real-time interactions with the environment. To the author's best knowledge, no work has been reported on a RL technique for CCP using multiple reconfigurable robots.

By taking the considerations discussed above, the general objective of the proposed methodology is summarised as follows:

1) A first-of-its-kind CCP and dynamic obstacle avoidance technique for self-organisation among multiple reconfigurable robots.
2) Formulation of a novel MARL framework using the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) using multiple shape-changing reconfigurable robots.
3) Training and validating the effectiveness of the proposed MARL framework in simulation environments and comparing performance with baselines.

The rest of the paper is structured into four sections. Section II details the formulation of the proposed MARL framework for CCP and dynamic obstacle avoidance using using a team of shape-changing reconfigurable robots called Smorphi. Section III documents the experimental results obtained from training simulation and validation, as well as comparison of coverage performance with baselines. Section IV concludes the paper and discusses future works of the research.

II. Problem Formulation

This section discusses the formulation of the CCP problem in multi-agent settings and the integration into the RL framework onboard reconfigurable robots. The structures of the observation and action spaces of each agent and the reward assignment for policy learning are also further detailed.

A. Smorphi: A Shape-Reconfigurable Robot

A set of Smorphi[1] reconfigurable robots is considered in this paper to formulate the proposed CCP. Fig. 1 documents the major components on a Smorphi robot, which consist of secondary modules (Modules 1, 3 and 4) controlled by a primary module (Module 2). The utilisation of mecanum wheels provide holonomic motion for each module during translation and shape-reconfiguration processes. The hinges between adjacent modules enable module rotation during shape change.

These features enable each Smorphi to be capable of toggling between O and I shapes, shown in Fig. 2. The robot will first assess if the respective reconfiguration workspace is free. If this condition is satisfied, the reconfiguration process, which elapses for approximately 4 seconds, is performed in which modules 3 and 4 are rotated 180° about the middle hinge while modules 1 and 2 remain fixed. Thus, this alteration of the physical morphology creates flexibility for the robots to navigate through different types of tight areas.

[1]www.wefaarobotics.com

B. Multi-Agent Reinforcement Learning (MARL)

In the domain of MARL, the objective is to enable a group of robots to learn behaviours for achieving goals through interactions with the environment. Unlike single-agent scenarios where the stationarity assumption is satisfied due to rewards and state transitions being fixed, multi-agent settings cause non-stationary expected rewards and state dynamics because all agents are concurrently interacting and learning from the environment. Hence, all agents must adapt their behaviours in accordance with the other agents' actions and changing policies as the environmental updates are dependent on the joint actions of all agents.

To achieve complete coverage while also avoiding collision with static and dynamic obstacles, the robots would need to ensure cooperation and collaboration to maximise the joint cumulative rewards. Given the hardware constraints resulting in limited perception, each robot only experiences partial observation and information of the local environment. Taking into account these considerations, the Decentralized Partially

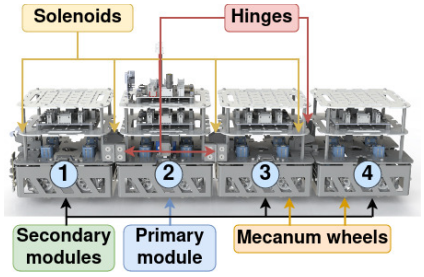
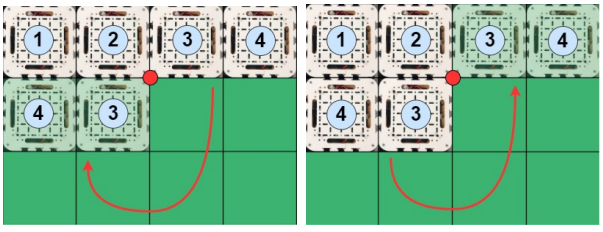


Fig. 1: Single Smorphi robot highlighted with the major components across the 4 modules and each module is allocated a unique number.



(a) (b)

Fig. 2: Smorphi reconfiguration workspace.(a): I to O shape and (b): O to I shape. Green cells represent the workspace that Smorphi will assess to enable successful reconfiguration into the respective shapes. Red dots represent the hinge in which modules 3 and 4 rotate about during shape change.

Observable Markov Decision Process (Dec-POMDP) is an appropriate framework to tackle the CCP problem in MARL [25].

An overview of the proposed Dec-POMDP framework is displayed in Fig. 3 which models a centralized training and decentralized execution approach [26], enabling stability for each agent's learning during training even when the policies of the other agents are changing. A local policy on each robot is trained to generate an action based on its own observations through interactions with the environment. A joint policy, which emerges from the information sharing, interactions and dependencies that the robots have with one another, takes in each robot's action to understand how the robots collectively behave. The calculated joint rewards are returned as an input to each robot's local policy and also updates the new state of the global environment for the next timestep.

A parameter tuple of (I, S, A_i, T, R, Ω_i, O, γ) is used to model the Dec-POMDP where:

- I is the finite set of N agents.
- S is the finite state space from all agents using the initial state distribution, b_0.
- A_i is the finite set of actions for each agent, i.
- A is the set of joint actions, where $A = \prod_i A_i$ and $\prod$ represents the Cartesian product operator.
- T is the state transition probability function, where $T{:}S \times A \times S \rightarrow [0, 1]$. This parameter displays the probability of transitioning from state $s \in S$ to $s' \in S'$ if the set of actions of $\vec{a} \in A$ is performed by the agents.
- Ω_i is a finite set of observations for each agent, i, where Ω is the set of joint observations and $\Omega = \prod_i \Omega_i$.
- O is the observation probability function, where $O{:}\Omega \times A \times S \rightarrow [0, 1]$. O indicates the probability of visualising the set of observations of $\vec{o} \in \Omega$ to obtain $s' \in S$ if the set of actions of $\vec{a} \in A$ is executed.
- R is the reward function, where $R{:}S \times A \rightarrow \mathbb{R}$. R represents the immediate joint reward achieved for using the set of actions $\vec{a}$ to be in state $s \in S$.
- γ is the discount factor which is usually equal to 1 in the finite-horizon case.

As discussed by [27], the joint policy is represented by a tuple of local policies from each robot. The history of each robot is explicitly represented to allow robots to access the histories of one another. The Action-Observation History (AOH) for each agent i, h_i^A, is the sequence of actions and observations taken and received at a specific timestep up until timestep t, given in (1).

$$h_i^A = (a_i^0, o_i^0, ..., a_i^t, o_i^t) \tag{1}$$

A deterministic policy of an agent i, π_i, is then a function of the history when the AOH is mapped to actions that contain all agent information. The value, $V(\pi)$, of the joint policy, π, given in (2), represents the expected cumulative reward obtained by the agents across all timesteps using the actions given by the policy until the horizon is achieved. The objective of the proposed framework would be to learn an optimal policy to maximise the total expected cumulative reward which reflects the quality of the coordinated behaviour in which the robots collectively select their actions.

$$V^{\pi}(s) = E[\sum_{t=0}^{h-1} \gamma^t R(\vec{a}_t, s_t) | s, \pi] \tag{2}$$

C. Policy Representation

1) Observation space: Each Smorphi robot can only access the spatial state of the environment through the lens of a partially observable discrete 2D grid world map centred around itself to provide information about the local vicinity. This information includes nearby static and external dynamic obstacles, covered and uncovered cells, and the position of the robots. If the robot is close to the boundaries of the environment, obstacles are added to the positions of the out-of-bounds areas. The partial observation map (6×6 Field Of View (FOV)) considers the practicality of real-world deployments as robots should not have a full state of the environment.

The only information given to each robot is its current location in global coordinates. However, each robot needs have access to information on the global area coverage which is usually outside the FOV. To this end, a map reconstruction is simultaneously conducted at each timestep where all robots contribute to a common map that logs information on explored areas while also identifying potential unexplored areas and previously explored regions that still consist of uncovered cells. To maintain explicit communication, all robots will also share their global positions with one another to enable the robots to learn to distinguish between robots and dynamic obstacles. Additionally, each robot will also have information on its current shape to learn to navigate through narrow spaces.

Specifically, the observation space for each robot i, summarised in Fig. 3, consists of a partial observation map around the robot, P_i, the current robot shape, CS_i, an array of the global positions of all agents, C, and the position of the nearest uncovered cell based on explored areas, UC_i.

2) Action space and reward assignment: Each robot is capable of discrete translation in the four cardinal directions and also shape-changing. If an action leads to an illegal move, the robot will maintain its current position for that timestep. This includes collision with obstacles or with other robots (multiple robots intend to move to the same position at the same time).

Unlike fixed morphology robots, the reward assignment for reconfigurable robots, collated in Table I, must account for the motions of each individual robot module and the shape-changing morphology of the robot. For each robot, the reward assignments, discussed by [28], are structured to consist of dense rewards, $R_{d,i}$, which are given frequently based on immediate actions and spare rewards, $R_{s,i}$, which are awarded according to the robots' collective progress upon achieving a milestone. The joint reward, R is then calculated

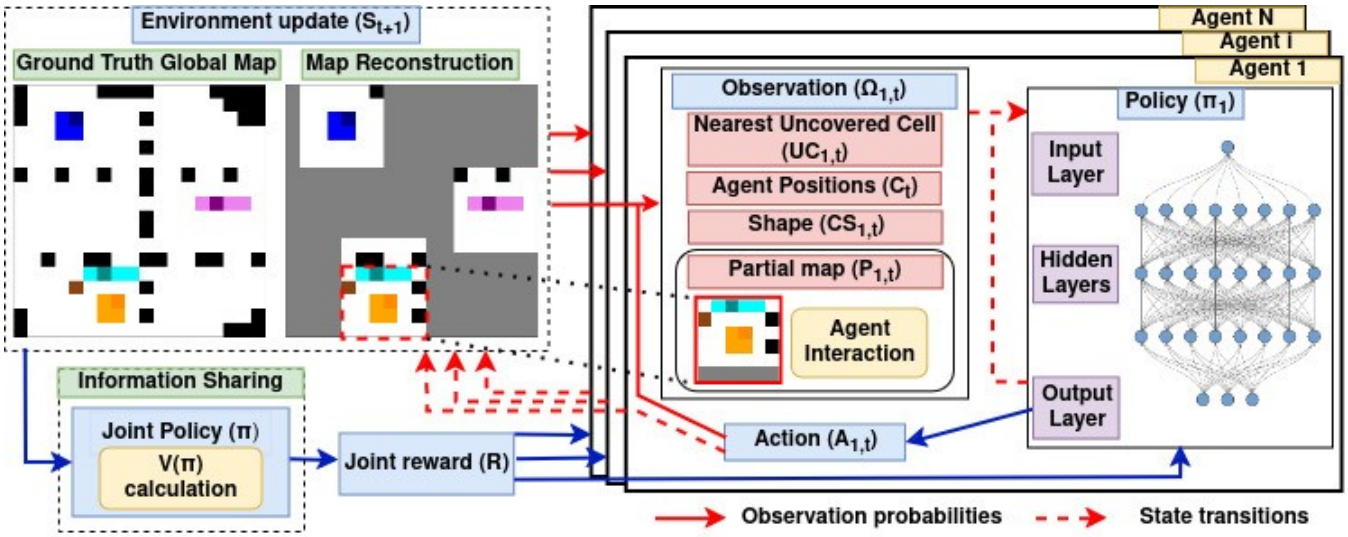


Fig. 3: Overview of policy training process using Dec-POMDP. In the maps, darker shades of coloured cells (Blue, Turquoise, Pink, and Orange) represent Smorphi's primary module and lighter shades represent secondary modules. Free space, static obstacles and dynamic obstacles are represented by white, black and brown respectively. The robots reconstruct a map based on combined partial observations and each robot's policy learns to output an action based on the observations. The joint policy trains the robots to iteratively select actions that collectively maximise the joint rewards.

as a sum of the individual rewards, R_i, obtained across all robots.

Positive dense rewards are given if the robot module visits uncovered cells and a bonus is given for uncovered cells at narrow spaces to incentivize the robots to shape change appropriately. Negative dense rewards are imposed when an action results in an illegal move or when a module visits an already covered cell again. Termination of an episode was avoided when performing an illegal move due to poor policy learning and exploration by the robot. The robot thus remains still for that timestep. To discourage collision with external dynamic obstacles, a penalty is also given to the robot if a dynamic obstacle is within the vicinity of the robot.

To guide the robots collectively towards complete area coverage, positive sparse rewards are awarded to each robot if a certain coverage percentage, CP, has been achieved. A higher reward is given to each coverage milestone closer to complete coverage as it would be more difficult to visit the remaining uncovered cells while also preventing deadlock. Simultaneously, to encourage the robots to complete the tasks in the shortest time, rewards are scaled (per N number of robots) and given based on the number of timesteps, t, taken to achieve complete coverage. In contrast, if complete coverage is not achieved after a certain maximum number of steps, t_s, a significant penalty is given to discourage deadlock.

TABLE I: Dense and sparse reward assignments for complete coverage using reconfigurable robots.

Type	Condition	Value
Dense	Uncovered cell	10
	Uncovered bonus cell	30
	Covered cell	-0.5
	Dynamic obstacle observed	-0.5
	Robot collision	-2
Sparse	CP = 25, 50, 60, 70, 80, 90	40
	CP = 95, 96, 97, 98, 99	80
	CP = 100	200
	t <200 + (50×(N-2))	200
	$t \geq$ 200 + (50×(N-2)) and t <400 + (50×(N-2))	150
	$t \geq$ 400 + (50×(N-2)) and t <600 + (50×(N-2))	100
	$t \geq$ 600 + (50×(N-2)) and t <800 + (50×(N-2))	50
	$t \geq$ 800 + (50×(N-2))	0
	$t > t_s$	-200

D. Multi-Agent Proximal Policy Optimisation (MAPPO)

Proposed by [29], PPO is a family of policy gradient methods that utilise the actor-critic architecture. The work [30] analysed the implementation of PPO for multi-agent scenarios, citing a strong baseline performance for cooperative MARL.

MAPPO extends the principles of PPO to multi-agent settings by incorporating parameter sharing among agents and utilising a centralised critic during training. The main component of PPO focuses on the Clipped Surrogate Objective Function, $J(\theta)$, shown in (3), which uses multiple epochs of stochastic gradient ascent to perform policy updates.

$$J(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad (3)$$

The ratio function, $r_t(\theta)$, displayed in (4), is the probability of taking the action a_t at state s_t in the current policy over the previous policy and represents the action probability using the current policy, π_θ, and old policy, $\pi_{\theta_{old}}$, respectively. A ratio function value more than 1 indicates that action a_t at state s_t is more likely to occur in π_θ than $\pi_{\theta_{old}}$.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{4}$$

The clipped component, $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)$, limits the objective function within the range $[1-\epsilon, 1+\epsilon]$ by penalising changes that lead to $r_t(\theta)$ away from 1. Clipping controls the policy updates to ensure that the change is not too large or too small which may lead to divergence. The unclipped component, $r_t(\theta)\hat{A}_t$, multiplies the probability ratios by the advantage and $J(\theta)$ will select the minimum value between the clipped and unclipped components. Hence, the usage of policy clipping and a centralized critic enables MAPPO to be advantageous in stability, scalability, and sample efficiency as opposed to other MARL algorithms.

E. Training Details

Training was conducted on teams of 2, 3, 4, and 5 Smorphi robots using the proposed framework. The number of agents required in an environment is determined such that the total initial coverage size by the robots maintains a percentage of approximately 5% of the square environment size. For example, a 13×13 environment, comprising 169 cells, requires 2 robots that cover an initial area of 8 cells. Similarly, a 20×20 environment, comprising 400 cells, requires 5 robots that cover an initial area of 20 cells, thus maintaining a consistent initial robot coverage to environment area percentage. Training was conducted on map sizes of 13, 16, 18, and 20. The initial positions of the robots are randomly generated at any free cell within the environment, which indicates that the robots could either be close or far apart from one another. Based on the scenario, the robots would learn how to coordinate their motions to ensure complete coverage is still achieved.

1) Environment Modelling: The maps used for the training are designed such that the positions of the obstacles ensure that only shape-change actions executed by the robots can enable them to achieve complete coverage and obstacle density is varied from 10% to 20%. Examples of the design of the different training maps are displayed in Fig. 4. External dynamic obstacles will be continuously moving within the environment to obstruct the paths of the robots, thereby encouraging robots to find an alternative path to avoid the obstacles. The dynamic obstacles are also capable of shifting out of the environment and reappearing at another section of the map.

2) Parameter Selection: The training was performed using Ray RLlib [31], modelled by PettingZoo [32], and supported by the TensorFlow framework. The training process was conducted on Ubuntu 20.04 using NVIDIA GeForce RTX 3080 GPU and Intel Core i7 CPU. The hyperparameters of the PPO algorithm used for training are summarised in Table II.

III. Results and Discussion

In this section, the experimental results of the training process and validation of the trained policies on unseen space-constrained environments are presented. Finally, the area coverage performance using the proposed framework is compared with fixed morphology robots baselines.

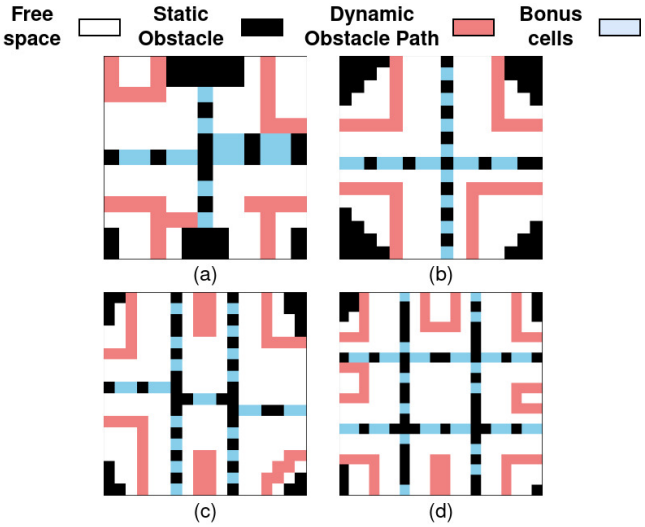


Fig. 4: Example of (a) 13, (b) 16, (c) 18, (d) 20 sized environments used for training. The obstacles segment the environment into different zones and require robots to shape change in order to achieve complete coverage.

TABLE II: Hyperparameters of the PPO algorithm used for training.

Hyperparameter	Value
Learning rate	1×10^{-5}
Training batch size	4096
Mini batch size	128
Lambda	0.95
Number of SGD iterations	30
Value loss coefficient	0.5
Entropy coefficient	0.01
Clip parameter	0.2
Number of workers	10

A. Training Results Analysis

Fig. 5 and Fig. 6 display the comparison of the mean reward and episode length across the experiments. From the results, the trend indicates that all experiment scenarios achieve full convergence as the mean rewards initially dip but increase and plateau at a stable reward value. Similarly, the mean episode lengths initially peak but decrease and plateau to a stable episode length.

The 13, 16, 18, 20-sized environments achieved an approximate convergence mean reward value of 3188, 4215, 5609, and 7413 at 2.06×10^6, 2.63×10^6, 3.78×10^6, and 9.68×10^6 trained timesteps respectively. Similarly, at these timesteps, the average episode lengths also converged to 162, 227, 258, and 229 respectively.

To cross-check the consistency of the results, the maximum reward obtained for the 13, 16, 18, 20-sized environment are 3308, 4433, 5865, and 7550 at the respective timesteps of 3.70×10^6, 3.66×10^6, 4.34×10^6, and 1.00×10^7. The minimum episode length achieved for the environments

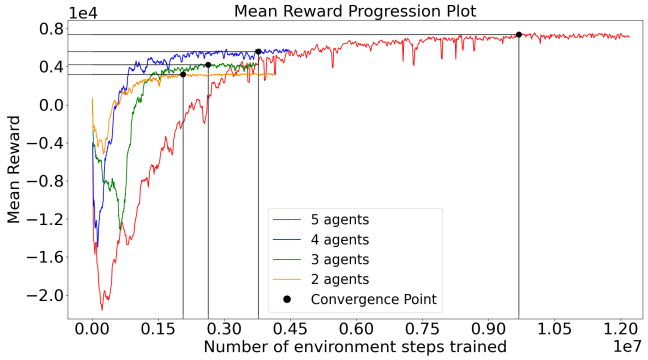


Fig. 5: Mean reward curves across the different training environments.

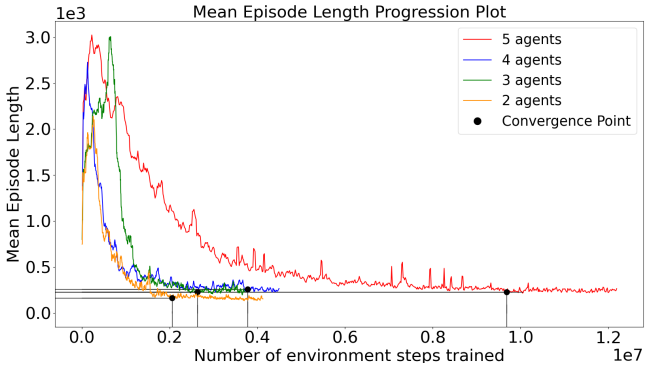


Fig. 6: Mean episode length curves across the different training environments.

were 134, 195, 227, and 216, corresponding to the same timesteps at the maximum reward, which is indicative of consistent policy learning.

It can be observed that the mean rewards increase as the environment size increases which is consistent with the larger spatial area and number of robots involved because more rewards can be accumulated. Additionally, with a larger number of agents and area, more timesteps are required to achieve convergence which could be potentially attributed to the increased number of possible states and variation in randomised starting positions, resulting in a more challenging policy learning process.

Nevertheless, the results have demonstrated the potential of the proposed framework to learn and achieve CCP in a team of reconfigurable robots, even when scaled across larger environments using an increased number of robots.

B. Policy Inference and Validation

To validate the generalisation capabilities of the trained policies, the policies are implemented onto unseen environments using the same environment setup configuration discussed in Section II-E. Three different unseen environments are tested for each map size and the results are collated in Table III. From the results, all trials achieved complete coverage, and the mean number of steps required across all different environment sizes was lower than the mean episode lengths documented in Section III-A. This is indicative that

TABLE III: Area coverage performance and steps required using the trained policies deployed across different test environments.

Map Size	Test Map 1		Test Map 2		Test Map 3		Mean Steps
	Steps Taken	Area (%)	Steps Taken	Area (%)	Steps Taken	Area (%)	
13×13	91	**100**	123	**100**	99	**100**	104
16×16	119	**100**	205	**100**	130	**100**	151
18×18	162	**100**	150	**100**	137	**100**	150
20×20	217	**100**	173	**100**	216	**100**	202

the trained policies are capable of generalising well even in newly deployed environments.

For each of the different map sizes, the waypoint trajectories of the robots from the best-performing complete coverage trial that used the least number of steps are displayed in Fig. 7. From the visualisation, it can be observed that the robots coordinate their motions and attempt to cover unvisited cells while avoiding visited cells, trajectories of other robots, and dynamic obstacles. For example, in Fig. 7(b), Smorphi2 extensively covers the bottom right section of the environment while Smorphi1 and 3 each cover extensively the respective top right and bottom left areas and covering the top left section together. Similarly, in Fig. 7(c), Smorphi2, 3 and 4 each extensively cover the left, right, and bottom sections of the map respectively. Simultaneously, Smorphi1 attempts to cover any remaining unvisited areas missed out by Smorphi2 before extensively covering the top area of the map. In Fig. 7(d), while Smorphi2, 3, and 5 cover areas far away from the initial positions, this can be attributed to the initial exploratory behaviour by the robot. Once a region of uncovered space is detected from the shared reconstructed map, each robot then covers the area more rigorously.

The training simulation results have shown that the proposed Dec-POMDP framework is capable of enabling coordination among multiple reconfigurable robots to achieve complete area coverage even in the presence of external dynamic obstacles. Additionally, the framework is also capable of scaling up towards larger environments using a larger reconfigurable robot team.

C. Baseline Comparisons

To further evaluate the effectiveness of the proposed MARL approach, the area coverage performance achieved by reconfigurable robots is compared with a team of fixed morphology robots using a new set of test maps. A fixed morphology robot group consisting of 1 I-shaped and 1 O-shaped robot is compared with 2 reconfigurable robots, and all robots are initialised at any random starting position.

From Table IV and Fig. 8, the fixed morphology robots do not achieve complete coverage in any of the environments. The inability to reconfigure hinders the effectiveness of area coverage as the robots are unable to pass through the tight spaces to access the other sections of the environment, resulting in an unproductive deployment. In contrast, the reconfigurable robots achieve complete coverage in all the scenarios. Similar to the analysis discussed in Section III-

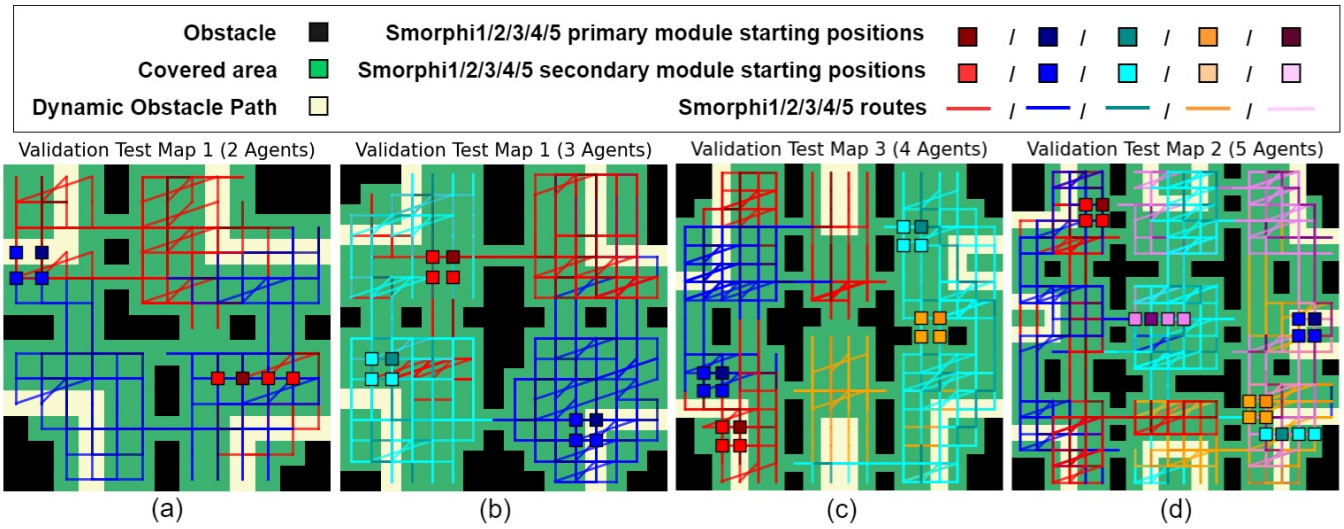


Fig. 7: Waypoint trajectories of the best complete area coverage performance by the robots across the different test map sizes (a) 13×13 (2 agents), (b) 16×16 (3 agents), (c) 18×18 (4 agents), and (d) 20×20 (5 agents).

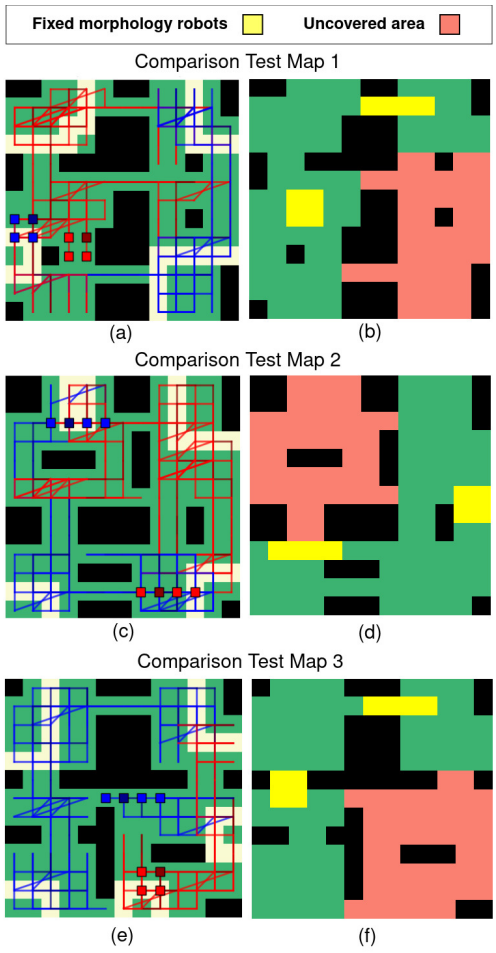


Fig. 8: Comparison of area coverage performance between reconfigurable robots that achieve complete coverage 8(a), 8(c), and 8(e), and fixed morphology robots 8(b), 8(d), and 8(f) that only achieve partial coverage across 3 test maps.

TABLE IV: Area coverage performance between reconfigurable robots and fixed morphology robots on different comparison maps.

Comparison Map	Area Coverage (%)	
	Reconfigurable (proposed)	Fixed
1	**100** (see Fig. 8(a))	64.57 (see Fig. 8(b))
2	**100** (see Fig. 8(c))	64.89 (see Fig. 8(d))
3	**100** (see Fig. 8(e))	63.64 (see Fig. 8(f))

B, from Fig. 8(a), 8(c), and 8(e), the reconfigurable robots coordinate their paths to each cover approximately half of the environment based on the positions of the other robots and identifying unexplored areas from the shared reconstructed map. Additionally, each robot also attempts to ensure covering areas where the other robot may have previously missed out to achieve complete coverage.

IV. Conclusion

This paper proposed a novel Decentralized Partially Observable Markov Decision Process framework for complete area coverage in the domain of reconfigurable robots. The framework considers the partial observations and positions of each reconfigurable robot to construct a shared map which all robots can use to coordinate their paths while also avoiding external dynamic obstacles. The framework trained policies for each robot in simulation across different sizes of environments while also being capable of scaling up to involve a larger number of robots. The policies are validated on a multitude of unseen environments and the robots still retain the capacity to achieve complete coverage. Further baseline comparison of coverage performance is conducted against fixed morphology robots in which the reconfigurable robots attained complete coverage while the fixed morphology robots were unable to obtain complete coverage across all test environments.

This work considers reconfiguration between two shapes,

and the proposed framework could be improved to consider multiple shape reconfigurations by increasing the action space. Exploration of this improvement is proposed for future work. Large environments with large numbers of robots would increase the time complexity of the training process. Composing larger environments into smaller ones would be advantageous to alleviate this issue. Therefore, combining map decomposition strategies with the proposed framework would be interesting for future work.

REFERENCES

[1] I. D. Wijegunawardana, M. A. V. J. Muthugala, S. M. B. P. Samarakoon, O. J. Hua, S. G. A. Padmanabha, and M. R. Elara, "Insights from autonomy trials of a self-reconfigurable floor-cleaning robot in a public food court," *Journal of Field Robotics*, vol. n/a, no. n/a. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22288

[2] X. Wang, M. Liu, C. Liu, L. Ling, and X. Zhang, "Data-driven and knowledge-based predictive maintenance method for industrial robots for the production stability of intelligent manufacturing," *Expert Systems with Applications*, vol. 234, p. 121136, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742301638X

[3] A. Ezhilarasu, J. J. Pey, M. A. V. J. Muthugala, M. Budig, and M. R. Elara, "Enhancing robot inclusivity in the built environment: A digital twin-assisted assessment of design guideline compliance," *Buildings*, vol. 14, no. 5, 2024. [Online]. Available: https://www.mdpi.com/2075-5309/14/5/1193

[4] J. J. J. Pey, A. P. Povendhan, T. Pathmakumar, and M. R. Elara, "Robot-aided microbial density estimation and mapping," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 2265–2272.

[5] M. S. K. Yeo, J. J. J. Pey, and M. R. Elara, "Passive auto-tactile heuristic (path) tiles: Novel robot-inclusive tactile paving hazard alert system," *Buildings*, vol. 13, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2075-5309/13/10/2504

[6] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901300167X

[7] 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.

[8] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "A survey on multi-robot coverage path planning for model reconstruction and mapping," *SN Applied Sciences*, vol. 1, no. 8, p. 847, Jul 2019. [Online]. Available: https://doi.org/10.1007/s42452-019-0872-y

[9] A. Janchiv, D. Batsaikhan, B. Kim, W. G. Lee, and S.-G. Lee, "Time-efficient and complete coverage path planning based on flow networks for multi-robots," *International Journal of Control, Automation and Systems*, vol. 11, no. 2, pp. 369–376, Apr 2013. [Online]. Available: https://doi.org/10.1007/s12555-011-0184-5

[10] P. Fazli, A. Davoodi, P. Pasquier, and A. K. Mackworth, "Complete and robust cooperative robot area coverage with limited range," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5577–5582.

[11] H. I. A. Perez-imaz, P. A. F. Rezeck, D. G. Macharet, and M. F. M. Campos, "Multi-robot 3d coverage path planning for first responders teams," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 1374–1379.

[12] 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.

[13] K. Albina and S. G. Lee, "Hybrid stochastic exploration using grey wolf optimizer and coordinated multi-robot exploration algorithms," *IEEE Access*, vol. 7, pp. 14 246–14 255, 2019.

[14] I. Caliskanelli, B. Broecker, and K. Tuyls, "Multi-robot coverage: A bee pheromone signalling approach," in *Artificial Life and Intelligent Agents*, C. J. Headleand, W. J. Teahan, and L. Ap Cenydd, Eds. Cham: Springer International Publishing, 2015, pp. 124–140.

[15] P. T. Kyaw, A. Paing, T. T. Thu, R. E. Mohan, A. Vu Le, and P. Veerajagadheswar, "Coverage path planning for decomposition reconfigurable grid-maps using deep reinforcement learning based travelling salesman problem," *IEEE Access*, vol. 8, pp. 225 945–225 956, 2020.

[16] A. V. Le, R. Parween, P. T. Kyaw, R. E. Mohan, T. H. Q. Minh, and C. S. C. S. Borusu, "Reinforcement learning-based energy-aware area coverage for reconfigurable hrombo tiling robot," *IEEE Access*, vol. 8, pp. 209 750–209 761, 2020.

[17] J. P. Carvalho and A. P. Aguiar, "A reinforcement learning based online coverage path planning algorithm," in *2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2023, pp. 81–86.

[18] J. Xiao, G. Wang, Y. Zhang, and L. Cheng, "A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning," *IEEE Access*, vol. 8, pp. 33 511–33 521, 2020.

[19] M. V. J. Muthugala, S. B. P. Samarakoon, and M. R. Elara, "Design by robot: A human-robot collaborative framework for improving productivity of a floor cleaning robot," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7444–7450.

[20] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.

[21] S. B. P. Samarakoon, M. V. J. Muthugala, and M. R. Elara, "Global and local area coverage path planner for a reconfigurable robot," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.

[22] W. Cheah, T. B. Garcia-Nathan, K. Groves, S. Watson, and B. Lennox, "Path planning for a reconfigurable robot in extreme environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 087–10 092.

[23] S. B. P. Samarakoon, M. V. J. Muthugala, and M. R. Elara, "Online complete coverage path planning of a reconfigurable robot using glasius bio-inspired neural network and genetic algorithm," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 5744–5751.

[24] I. D. Wijegunawardana, S. B. P. Samarakoon, M. V. J. Muthugala, and M. R. Elara, "Fmea-based coverage-path-planning strategy for floor-cleaning robots," *Advanced Intelligent Systems*, vol. 5, no. 11, p. 2300260, 2023.

[25] C. Amato, G. Chowdhary, A. Geramifard, N. K. Üre, and M. J. Kochenderfer, "Decentralized control of partially observable markov decision processes," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 2398–2405.

[26] A. Wong, T. Bäck, A. V. Kononova, and A. Plaat, "Deep multiagent reinforcement learning: challenges and directions," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5023–5056, Jun 2023. [Online]. Available: https://doi.org/10.1007/s10462-022-10299-x

[27] F. A. Oliehoek, C. Amato *et al.*, *A concise introduction to decentralized POMDPs*. Springer, 2016, vol. 1.

[28] A. Mohtasib, G. Neumann, and H. Cuayáhuitl, "A study on dense and sparse (visual) rewards in robot policy learning," in *Towards Autonomous Robotic Systems*, C. Fox, J. Gao, A. Ghalamzan Esfahani, M. Saaj, M. Hanheide, and S. Parsons, Eds. Cham: Springer International Publishing, 2021, pp. 3–13.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[30] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. M. Bayen, and Y. Wu, "The surprising effectiveness of MAPPO in cooperative, multi-agent games," *CoRR*, vol. abs/2103.01955, 2021. [Online]. Available: https://arxiv.org/abs/2103.01955

[31] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, J. Gonzalez, K. Goldberg, and I. Stoica, "Ray rllib: A composable and scalable reinforcement learning library," *CoRR*, vol. abs/1712.09381, 2017. [Online]. Available: http://arxiv.org/abs/1712.09381

[32] J. K. Terry, B. Black, A. Hari, L. S. Santos, C. Dieffendahl, N. L. Williams, Y. Lokesh, C. Horsch, and P. Ravi, "Pettingzoo: Gym for multi-agent reinforcement learning," *CoRR*, vol. abs/2009.14471, 2020. [Online]. Available: https://arxiv.org/abs/2009.14471