Multi-Robot Path Planning for Comprehensive Area Coverage in Complex Environments

Manish Kumar
Department of IT
IIIT - Allahabad
Prayagraj, India
mml2022009@iiita.ac.in

Arindam Ghosh
Department of IT
IIIT - Allahabad
Prayagraj, India
rsi2021001@iiita.ac.in

Muneendra Ojha
Department of IT
IIIT - Allahabad
Prayagraj, India
muneendra@iiita.ac.in

2024 28th International Conference on Methods and Models in Automation and Robotics (MMAR) | 979-8-3503-6234-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/MMAR62187.2024.10680839

***Abstract*—Multi-robot coverage path planning (mCPP) involves devising efficient motion sequences for robots to cover all positions within a workspace, excluding obstacles. This paper focuses on addressing the path-planning challenge of a Divide Area based on Robot's Initial Positions (DARP), where a group of mobile robots is tasked with covering a predefined area containing obstacles. This work proposes an improved DARP algorithm for efficient coverage. The proposed algorithm, combined with the A^* and the spanning tree coverage algorithm, assigns tasks to robots to optimally achieve full coverage of the desired area. It transforms the initial mCPP problem into individual coverage path planning tasks of single-robot, combining their solutions to form the optimal solution for mCPP. This method significantly improves performance by reducing coverage time, minimizing the number of turns taken by robots, and enhancing overall coverage efficiency.**

***Index Terms*—Multi-Robot, Path-Planning, Area Coverage, Spanning Tree Coverage, DARP, mCPP, A^* Algorithm**

I. Introduction

In recent years, the widespread adoption of robotic systems across diverse sectors, including warehouse logistics and disaster response, has surged. A key challenge in deploying these systems involves efficiently navigating and covering complex environments with multiple robots, necessitating robust path-planning strategies. Multi-robot path planning for comprehensive coverage is important for various applications, such as cleaning, demining, and search-and-rescue missions. The coverage path planning problem (CPP) involves finding the best path to cover a defined area while avoiding obstacles. Initially, research predominantly concentrated on single-robot CPP methodologies, yielding diverse approaches like trapezoidal decomposition and grid-based techniques [1], [2]. However, with the expansion of coverage requirements, mCPP emerged to address larger areas, albeit facing communication, coordination, and collision avoidance hurdles.

The term "Comprehensive area coverage" refers to the use of multiple robots to explore and scan a defined area. This allows for better coverage rates, fault tolerance, and synchronized efforts to handle larger areas and perform specific tasks [3]. However, conventional single-robot path planning struggles with complex environments, uncertain terrains, dynamic obstacles, constrained communication, and diverse performance metrics. Additionally, ensuring fair task allocation, minimizing redundant coverage, and synchronizing robot movements further exacerbate the complexities of the problem. Robotics technology has led to the mCPP problem, which is an NP-hard challenge. Researchers are looking at simpler versions and optimization strategies to solve it [4]. The mCPP strategy involves DARP for cell clustering and the STC algorithm for generating optimal paths per robot. However, challenges persist with task area division and assumptions about obstacle areas, hindering efficient coverage.

Researchers have developed various AI, optimization, and robotics approaches for addressing multi-robot path planning challenges. However, DARP sometimes encounters local optima due to obstacles obstructing robots' paths to assigned cells, necessitating longer detours to bypass obstacles and reach target cells. In response to these challenges, the proposed approach enhances the DARP algorithm using the A* algorithm for more rational and uniform cell assignments. This approach reduces robot turns, shortens coverage time, saves energy, and enhances coverage rate, achieving comprehensive coverage in complex environments.

The paper's structure is as follows: Section II reviews prior research. Section III outlines the methodology, including an overview of the problem and fundamental concepts of mCPP and DARP. Section IV presents and discusses the research findings and outcomes. Finally, Section V concludes with our conclusions and suggestions for future research.

II. Related Works

A. *Multi-Robot Balanced Coverage Path Planning for Surveillance Tasks*

In patrol missions, comprehensive area coverage is crucial for effective surveillance and security. A balanced coverage path planning strategy using multiple robots is essential for this task. [5] presents a novel strategy for mCPP for patrol missions. The approach assigns unique non-redundant paths to each robot by constructing a movement graph based on the given map and solving a variant of the traveling salesman problem. Paths are then combined and adjusted to ensure a balanced distribution. A distributed algorithm using expected idleness for multi-robot patrolling has been examined over an area of interest in [6] to improve cooperation efficiency between robots while maintaining fault tolerance and scalability, by focusing on frequency-based patrol strategies and information sharing. The work presented in [7], explores autonomous agents' patrolling environments, designing optimal open-loop trajectories and control laws. It covers graph representation and algorithms for computing minimum refresh time and latency trajectory for chain and tree graphs. The cyclic TSP tours and graph partitioning strategies are investigated in [8]. Cyclic-based strategies perform better with smaller teams despite higher team costs. Partitioning strategies are most efficient when dealing with larger teams and imbalanced graph topologies.

B. Integrated methodologies for autonomous cleaning robot path planning

Autonomous cleaning robots are increasingly used for tasks like floor cleaning. Combined coverage path planning strategies have gained attention to optimize their performance. [9] introduces a method for enabling autonomous cleaning robots to achieve comprehensive coverage path planning in diverse environments. The approach integrates random path planning with comprehensive coverage planning to adapt to various environments and ensure thorough coverage in confined spaces. The biologically inspired neural network approach integrates obstacle avoidance for cleaning robots for real-time map-building and area-covering operations in dynamic environments. The model uses efficient locally connected neurons for collision-free path planning, enabling effective navigation in dynamic environments with limited sensory information [10], [11]. Another innovative method for complete coverage path planning for independent mobile robots is presented in [12] to achieve both efficiency and thoroughness in coverage. It employs templates and heuristics, enabling robots to follow predefined motion paths and optimize their efficiency through search algorithms.

C. Efficiently navigating multiple robots through known terrain for area coverage

The mCPP problem in known terrain is a critical challenge in robotics. It involves coordinating multiple robots to cover an area efficiently while avoiding obstacles. [13] proposes a method to address the mCPP problem in maritime Search and Rescue missions by employing multiple AUVs equipped with Side-Scan Sonar capabilities. It aims to locate targets from sonar images in complex SAR scenarios efficiently by using Morse decomposition to divide the area. A novel algorithm for multi-robot forest coverage has been designed for efficient terrain coverage in [14]. It ensures a balanced coverage of forests, with a maximum cover time that is eight times larger than optimal. [15] addresses the complete coverage path planning problem in robot planning. It presents a geometric algorithm that improves the efficiency of the sweeping line strategy by minimizing extra relocations. An algorithm designed for mCPP, MSTC*, is particularly suited for large-scale tasks in [16]. Building upon the Spiral-STC approach, MSTC* effectively integrates stringent physical limitations such as terrain traversability and material load capacity.

III. Methodology

This section outlines the methodology of the investigation, including the problem formulation, and fundamentals of mCPP, DARP, STC. It provides an overview of the approach taken to address the objectives.

A. Grid map formulation

The environmental map is created using a grid method where each grid represents the robot's dimensions. The area is confined within a rectangular area defined by coordinates (x, y) and divided into a finite number of equally sized cells. Figure 1 shows the grid map of the environment with obstacles.

$$U = \{(x, y) \mid x \in [1, p], y \in [1, q]\} \tag{1}$$

where p, and q are the count of rows and columns post discretizing the area to be covered, respectively. The total number of cells is given by $n = p \times q$. Additionally, it is presumed that no obstacles are positioned in U. After the obstacle cells are placed, it is represented as:

$$B = \{(x, y) \in U \mid (x, y) = \mathit{occupied}\} \tag{2}$$

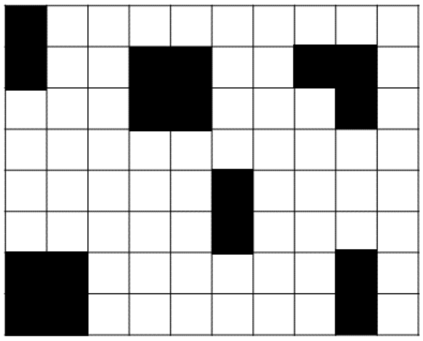

Fig. 1. Cell Discretization with Obstacles

The coverage of required cells is reduced as robots are unable to navigate obstacles. This implies that:

$$L = U \setminus B \tag{3}$$

and the number of cells to cover has been decreased to $l = n - n_0$, where $n_0 = |B|$

Definition 1: Two cells (x_i, y_i) and (x_j, y_j) are considered as the adjacent cells if the following constraint holds:

$$\|x_i - x_j\| + \|y_i - y_j\| \leq 1 \tag{4}$$

Definition 2: Every sequence of cells $X = ((x_1, y_1), ..., (x_m, y_m))$ can be considered a valid robot path of length m where the following constraints are held:

- $(x_i, y_i) \in L, \forall i \in \{1, \dots, m\}$
- Every two consecutive cells, i.e.(x_i, y_i) and $(x_i + 1, y_i + 1)$, are adjacent, $\forall i \in \{1, \dots, m - 1\}$

Additionally, a loop with a length of m represents a closed path. where the additional condition is held: (x_1, y_1) and (x_m, y_m) are adjacent.

The robot positions on the map at time t are defined as:

$$p_i(t) = (x_i, y_i) \tag{5}$$

Where $(x_i, y_i) \in L, \forall i \in 1, \ldots, n_r$, n_r represents the number of active robots. The starting location of the i^{th} robot within L is denoted as $p_i(t_0)$.

The mCPP problem can be formulated by the path of the robots to be covered in which cells must be uniformly distributed among the robots.

$$\begin{aligned} &\min_X \quad \max(|X_i|) \\ &\text{subject to,} \quad X_1 \cup X_2 \cup \cdots \cup X_{n_r} \supseteq L \end{aligned} \tag{6}$$

Where $|X_i|$ represents the path length of i^{th} robot.

An optimal solution for the mCPP problem is composed of a selection of sets $X_1, X_2, ..., X_{n_r}$, iff:

- $X_i \cap X_j = \emptyset, \quad \forall i, j \in \{1, \ldots, n\}, \quad i \neq j$
- $X_1 \cup X_2 \cup \cdots \cup X_{n_r} = L$
- $|X_1| \approx |X_2| \approx \cdots \approx |X_{n_r}|$
- X_i is connected $\forall i \in \{1, \ldots, n_r\}$
- $p_i(t_0) \in X_i$

Firstly, each cell must belong exclusively to one robot's set, guaranteeing a non-backtracking solution. Secondly, the combined sets of all robots must encompass every unblocked cell, ensuring complete coverage. Thirdly, each robot's set should contain a comparable number of cells, optimizing multi-robot dynamics. Additionally, cells within each set must form compact, contiguous regions to facilitate seamless navigation. Finally, each robot's initial position must reside within its respective set, minimizing preparation time and energy. The algorithm ensures a fair partition of the area and allows for the inclusion of arbitrary points. It addresses cases where at least one optimal solution is feasible.

B. Divide Areas Based on Robots Initial Positions (DARP)

The DARP algorithm is crafted to divide a terrain into distinct regions, each dedicated to the initial position of a robot. This solution is derived from optimizing the mCPP problem as outlined in equation (6) and described in [4]. It begins with cell assignment to robots and is determined iteratively using a minimization approach based on an evaluation matrix (E) indicating reachability from each cell to the robot's initial position, followed by region determination for each robot utilizing the assignment matrix (A).

$$A_{x,y} = argmin_{i \in \{1,\ldots,n_r\}} E_i|x, y, \forall (x, y) \in L \tag{7}$$

The algorithm assigns cells to robots, plans operations, adjusts positions, and minimizes a cost function, represented in (8) while equally distributing cells among robots using the Cyclic Coordinate Descent function, formulated in equation (9) and updating correction factors iteratively.

$$J_i = \frac{1}{2}(k_i - f)^2 \tag{8}$$

Where, $k_i = |X_i|, \forall i \in \{1, \ldots, n_r\}$ and f denotes the global "fair share": $f = l/n_r$. The fair share is the number of cells each robot would cover if the cells are equally distributed among the robots. A gradient descent based scalar correction method is employed in the equation (9).

$$m_i = m_i - \eta \frac{\partial J_i}{\partial m_i}, \eta > 0, \forall i \in \{1, \ldots, n_r\} \tag{9}$$

where m_i serves as a scalar correction factor for the i^{th} robot. Therefore, each assessment matrix E_i is updated by using a factor m_i, illustrated in (10) and described in [4], resulting in a more accurate evaluation.

$$E_i = m_i \cdot E_i \tag{10}$$

The algorithm optimizes cell assignments by incentivizing nearby regions, formalizing spatial connectivity with equation (11), and strategically disregarding and reestablishing connectivity to avoid local minima.

$$C_{i,x} = \min(||x, y|| - r) - \min(||x, y|| - q), \forall r \in R_i, q \in Q_i \tag{11}$$

where R_i represents the cohesive group of cells occupied by the i^{th} robot, while C_i encompasses all additional connected sets assigned to the same robot but lacking spatial connectivity with the R_i set. The final refinement in the i^{th} evaluation matrices is devised in equation (12).

$$E_i = C_i \circ (m_i, E_i) \tag{12}$$

where $\circ$ denotes the operation of element-wise multiplication.

C. Overview of the A^*-DARP Algorithm

The DARP algorithm effectively assigns cells to multiple robots by considering the proximity between each robot and the cell. It performs admirably in scenarios where open areas are available near the initial positions of the robots. However, in cases where an obstacle obstructs the path between the cell and the robot's starting point, and this cell happens to be the closest one to the respective robot, DARP fails to provide an optimal solution.

Figure 2 illustrates the scenario properly. The initial positions of Robot 1 and 2 and the difficult Region 1 make it difficult for the DARP algorithm to manage effectively. E_i generated by calculating Euclidean distance is initially assigned Region 1 to Robot 2. However, considering the obstruction of obstacles, it is more practical to assign Region 1 to Robot 1 as Robot 1 can access it more efficiently than Robot 2. Assigning Region 1 to Robot 2 increases turns, prolonging coverage time and causing greater losses. This paper suggests improving the DARP algorithm with A^* integration. By leveraging A^*'s strengths, the approach enhances evaluation matrix

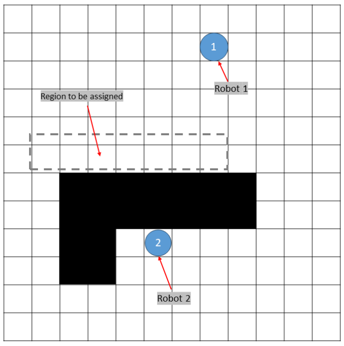

Fig. 2. Difficult scenario for DARP algorithm

generation for rational cell allocation. The A^* algorithm, formulated in equation (13), calculates the shortest paths from each robot to all reachable cells.

$$f(n) = g(n) + h(n) \tag{13}$$

The robots are limited to moving in diagonal directions, and the chosen estimated cost distance $h(n)$ for the A^* algorithm is the Manhattan distance, represented in equation (14), from the initial point to the target end-point.

$$h(n) = |x_1 - x_2| + |y_1 - y_2| \tag{14}$$

The collection of shortest paths obtained serves as the initial assessment matrix for the i^{th} robot.

D. Spanning Tree Area Coverage

A Spanning tree-based approach generates a navigating graph for the robot over a grid map, constructing the minimum path covering the entire operation area from any unoccupied cell, with terrain divided into large square cells with obstructed areas at least four times the size of a cell. Figure 3 represents the grid map decomposition and graph generation decomposition. The obstacle blocks are denoted in black, and the unobstructed blocks are in blue.

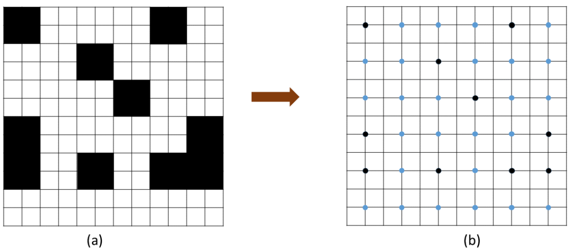

Fig. 3. (a) Grid map decomposition (b) Graph node generation

A minimal spanning tree is created utilizing either Kruskal's or Prim's algorithms. The robot follows the tree in a CW or CCW direction, creating an optimal closed path for coverage time. Figure 4 depicts the construction of the minimal spanning tree for all the unobstructed nodes decomposition of the grid map and graph generation.

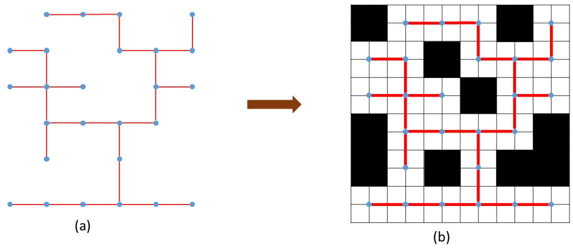

Fig. 4. (a) Construct MST (b) Apply MST to the original map

E. Environment Setup for simulation

The grid system in the simulated environment is created using the PyGame module on the Python platform. It is designed to simulate a virtual space, with dimensions of 20×20 cm, divided into 20 rows and 20 columns of grid cells, each measuring 10×10 mm, that the robots can navigate. Simulations assess the model's performance in varied settings, where black squares represent obstacles occupying at least one grid cell to prevent overlap with free space. The simulations involve variations in the number of robots, obstacles, the initial starting positions of robots, and obstacle placements within the grid world. Three experiments, involving two, three, and four robots respectively, are conducted to plan area coverage paths, with the routes displayed using distinct colorings corresponding to the number of robots involved.

IV. Results and Discussion

This section presents experiment findings, objectively demonstrating and analyzing outcomes while discussing their impact on existing research.

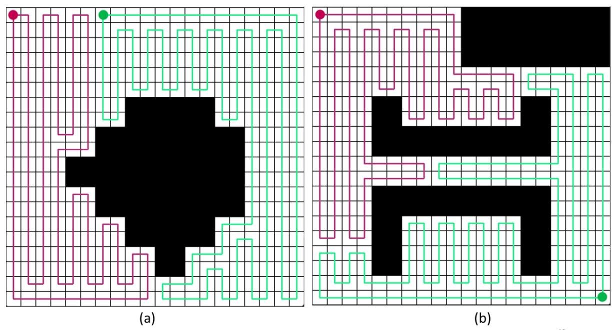

Fig. 5. Simulation result with (a) one obstacle and (b) multiple obstacle for DARP algorithm

A. Simulation Results

The simulation results of robots' coverage paths are presented in a grid-based environment under diverse scenarios. Different configurations are tested, varying the number and positions of obstacles and the number of robots. Figure 5 shows the simulation result in two different environments with two robots for DARP algorithm. The small dot indicates the robot's initial position, which is different in both scenarios.

The A^*-DARP algorithm is applied to plan area coverage paths in an environment with two robots, akin to the setup used for the DARP algorithm. Figure 6 represents the result of the path planning of two robots with A^*-DARP algorithm.

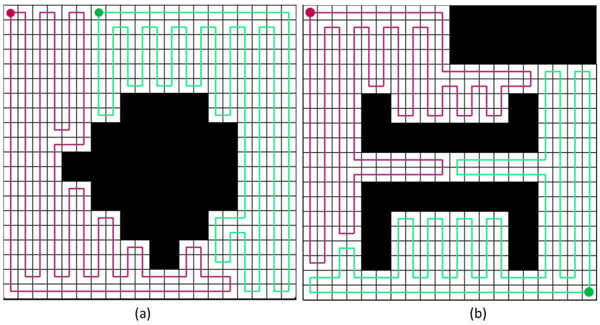

Fig. 6. Simulation result with (a) one obstacle and (b) multiple obstacle for A*-DARP algorithm

Consider a special situation where three robots are assigned to cover a particular region. Two distinct path planning for area coverage algorithms are investigated, namely DARP and proposed A^*-DARP.

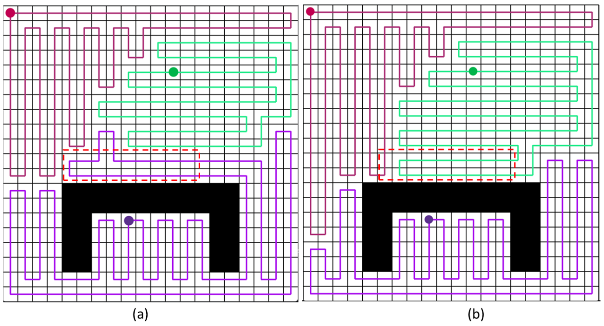

Fig. 7. Assignment of specific region: (a) DARP (b) A*-DARP with 3 robots

In DARP, the purple robot is assigned to cover a region marked by red dashed lines. However, as it approaches this region, it encounters an obstacle blocking its direct path. This obstacle lies between the purple robot and its designated area. Consequently, the purple robot must navigate around the obstacle, which increases the number of turns and coverage time, it needs to take to reach the region. This detour, caused by the obstacle, can slow down the overall path-planning for the area coverage process. Figure 7 depicts the area coverage path planning for DARP and A^*-DARP with three robots. In the A^*-DARP algorithm, the path-planning process is optimized using the A^* algorithm. The region marked by red dashed lines is assigned to the green robot by the proposed algorithm. No obstacles are obstructing the green robot's path to the desired region. As a result, the green robot can navigate directly to its designated area without any detours or additional turns. The efficient path-planning process reduces robot turns, enhancing performance and optimizing area coverage.

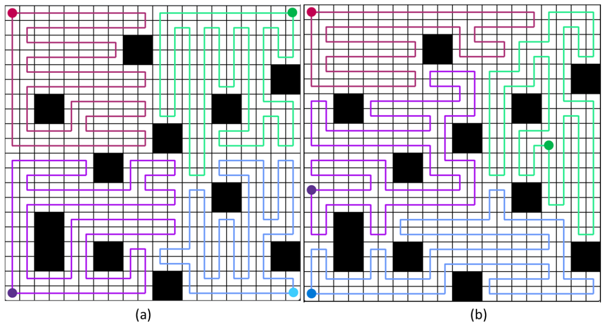

Fig. 8. DARP algorithm with the different initial position of 4 robots

In another experiment, four robots are deployed to cover the designated area. Various initial positions for the robots are explored to assess their impact on path planning. Figure 8 illustrates the outcome of the DARP algorithm, where the area is divided among the four robots for efficient coverage. A^*-DARP optimizes cell division for four robots with diverse initial positions in the same environment. Figure 9 showcases the path planning results achieved through A^*-DARP.

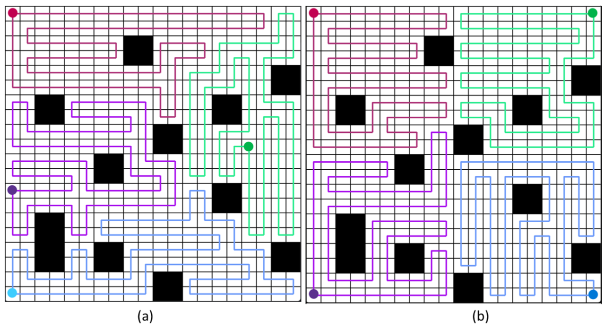

Fig. 9. A*-DARP algorithm with the different initial position of 4 robots

B. Performance Discussion

Evaluating the performance of the proposed methodology is crucial for assessing its effectiveness and efficiency. This involves analyzing metrics such as cell distribution uniformity and robot efficiency in terms of turns taken.

TABLE I
COMPARISON OF CELL DISTRIBUTION

No. of Robots	Total Cells	Occupied Cells	Free Cells	DARP Algorithm		A*-DARP Algorithm	
				Allocated Cells	δ_{max}	Allocated Cells	δ_{max}
2	400	92	308	136, 172	36	152, 156	4
2	400	112	288	132, 156	24	144, 144	0
3	400	40	360	92, 108, 160	68	120, 120, 120	0
4	400	48	352	72, 92, 92, 96	24	88, 88, 88, 88	0
4	400	48	352	72, 88, 92, 100	28	88, 88, 88, 88	0

Table I illustrates a detailed comparison of cell distribution between two distinct algorithms, DARP and A^*-DARP. Efficient cell allocation is crucial in multi-robot path planning for optimal area coverage. In path planning, cells are categorized as occupied or free. Occupied cells are blocked by obstacles, while free cells are for robot navigation. Assigning these cells to robots affects path planning efficiency. δ_{max} quantifies the variability in cell allocation among the robots by capturing the maximum difference observed in the number of cells allocated to each robot. A higher δ_{max} value indicates a greater imbalance in cell distribution among the robots. The DARP algorithm led to uneven area coverage due to non-uniform cell distribution among robots. In contrast, the A*-DARP algorithm achieved balanced coverage by allocating an equal number of cells to each robot. However, the proposed algorithm has a limitation in that it restricts environments with dense, convex-shaped obstacles. A*-DARP is the optimal choice for area coverage with multiple robots in such settings.

TABLE II
COMPARISON OF NUMBER OF TURNS

No. of Robots	DARP Algorithm		A*-DARP Algorithm	
	Turns	δ_{max}	Turns	δ_{max}
2	25, 42	17	31, 34	3
2	33, 37	4	33, 35	2
3	18, 21, 37	9	21, 22, 31	10
4	21, 26, 26, 28	7	21, 22, 24, 26	5
4	17, 28, 29, 35	18	17, 27, 29, 30	13

Another crucial factor in evaluating area coverage path planning performance is the number of turns each robot requires to navigate its allocated cells effectively. Table II compares the turns taken by robots using DARP and A^*-DARP algorithms. The maximum difference observed in the number of turns taken by the robots is captured by δ_{max}. A^*-DARP outperforms DARP in turn efficiency, requiring fewer turns for robots to cover their cells. The integration of A^* enhances path planning, optimizing coverage in critical scenarios. The A*-DARP algorithm outperforms the traditional DARP algorithm in multi-robot path planning tasks, resulting in optimized resource utilization and improved method performance.

V. CONCLUSION

This work aims to achieve comprehensive area coverage by efficiently coordinating a team of multiple robots. The DARP approach, a cell division algorithm, assigns cells to each robot based on their initial positions. By comparing the DARP and A*-DARP algorithms, we highlight the importance of algorithm selection in optimizing path planning efficiency. A*-DARP, utilizing the A* heuristic search, excels in generating more efficient paths with fewer turns for area coverage. These results underscore the potential of A*-DARP to enhance the performance and autonomy of multi-robot systems in practical settings. Future research should focus on refining A*-DARP and exploring alternative approaches to further advance autonomous robotic systems in complex environments.

REFERENCES

[1] 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.

[2] P. T. Kyaw, A. Paing, T. T. Thu, R. E. Mohan, A. V. 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.

[3] 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, pp. 1–24, 2019.

[4] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, "Darp: Divide areas algorithm for optimal multi-robot coverage path planning," *Journal of Intelligent & Robotic Systems*, vol. 86, pp. 663–680, 2017.

[5] S. Lee, "A multi-robot balanced coverage path planning strategy for patrol missions," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 2021, pp. 1567–1569.

[6] C. Yan and T. Zhang, "Multi-robot patrol: A distributed algorithm based on expected idleness," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 1729881416663666, 2016.

[7] 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.

[8] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen, "Finding optimal routes for multi-robot patrolling in generic graphs," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 363–369.

[9] Y. Liu, X. Lin, and S. Zhu, "Combined coverage path planning for autonomous cleaning robots in unstructured environments," in *2008 7th World Congress on Intelligent Control and Automation*, 2008, pp. 8271–8276.

[10] C. Luo, S. Yang, and X. Yuan, "Real-time area-covering operations with obstacle avoidance for cleaning robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2359–2364 vol.3.

[11] C. Luo, S. Yang, and M.-H. Meng, "Real-time map building and area coverage in unknown environments," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 1736–1741.

[12] Y. Mao, L. Dou, J. Chen, H. Fang, H. Zhang, and H. Cao, "Combined complete coverage path planning for autonomous mobile robot in indoor environment," in *2009 7th Asian Control Conference*, 2009, pp. 1468–1473.

[13] C. Cai, J. Chen, Q. Yan, and F. Liu, "A multi-robot coverage path planning method for maritime search and rescue using multiple auvs," *Remote Sensing*, vol. 15, no. 1, p. 93, 2022.

[14] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3852–3857.

[15] Z. Yao, "Finding efficient robot path for the complete coverage of a known space," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3369–3374.

[16] J. Tang, C. Sun, and X. Zhang, "Mstc:multi-robot coverage path planning under physical constrain," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2518–2524.