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

An Efficient Coverage Method for Irregularly Shaped Terrains

Yuxuan Tang, Qizhen Wu, Chunli Zhu and Lei Chen

Abstract— **In mobile robotics, effectively covering known terrains is essential. While grid-based methods surpass exact cell decomposition in path length and multi-robot scalability, they face challenges in irregular areas. Here we develop a model for shortening coverage paths in arbitrary environments using grid-based methods, which redefines the path optimization problem as finding the largest Hamiltonian sub-graph of a given grid graph. Additionally, we present a Hamiltonian cycle expansion strategy to simplify the resolution process and propose a low-repetitive coverage path planner based on the strategy. Our path planner enables the quick finding of an efficient full coverage path in any region. Simulation results show that our algorithm consistently produces efficient coverage paths across diverse settings and demonstrates its adaptability in multi-robot systems.**

I. Introduction

The coverage path planning (CPP) problem, known as an essential robotic technique, is the task of finding a path provided for robots to follow and completely cover the area. Classic applications of the CPP technique include indoor cleaning [1], modern agriculture [2], [3], landmine clearing [4], regional reconnaissance [5], disaster relief [6] and lawn mowing [7]. Planning coverage paths within known areas necessitates rigorous efficiency standards, leading to extensive research aimed at eliminating repeated paths while ensuring complete area coverage [8]–[10]. Among these studies, the CPP approaches of exact cell decomposition (ECD) and approximate cell decomposition (ACD) are predominant, each offering distinct benefits [11].

The ECD methods reflect the precise boundary shape, which ensure the robot's proper avoidance of obstacles and complete coverage of boundary areas during covering process [12]. Boustrophedon cellular decomposition (BCD) is a highly representative approach among ECD methods, which uses back and forth routes to cover each sub-region obtained by BCD [13]. Specifically, [8] and [14] adopts BCD for area segmentation and further partitions the cells to construct Euler paths that uniquely traverse all cells. Building on this work, [15] extends it to multi-robot coverage tasks. Furthermore, [16] introduces a rapid solution for optimizing scan combinations within each cell , resulting in lower path costs compared to conventional coverage planners. Although these applications demonstrate the ability of BCD in shortening coverage paths, they yet fail to address the efficiency declines arising from cell commuting. The irregularity of the designated area and the accuracy in contour representation contribute to the inevitable occurrence of inefficient backtrack pathways.

In contrast, the ACD algorithms, namely grid-based algorithms, inherently avert path backtracking and offer efficient scalability to multi-robot systems. As a leading grid-based method, spanning tree covering (STC) ensures non-repetitive complete coverage in specific shaped areas by finding a spanning tree and forming a Hamiltonian cycle around it [9]. Furthermore, the uniformity of grids facilitates the distribution of coverage loads among robots, leading to the development of numerous superior multi-robot coverage methods derived from STC [17]–[20]. However, when applied to irregularly shaped terrains, the STC algorithm often struggles to maintain its low redundancy advantage. Those improved algorithms focus on handling partially occupied mega-cells to sustain its proficiency in creating efficient coverage paths, but they often perform poorly in complex environments [21]–[23]. Thus, the derived multi-robot coverage methods are limited to coarse coverage tasks due to the constraints of the STC algorithm.

In summary, when generating coverage paths, current studies struggle to achieve complete coverage and minimal repetition simultaneously. To utilize the low redundancy advantage of the ACD methods and ensure complete coverage, it is imperative to develop a new problem formulation for minimizing the path lengths in arbitrary environments instead of patching the spanning tree methods. Building on this concept, the contributions of this paper are detailed as follows:

1) A problem model for shortening coverage paths in arbitrary environments using graph theory principles.
2) A strategy for tackling the computational difficulty, and a derived low-repetitive path planner for comprehensive coverage in any region.
3) A set of trials that validate the efficiency and multi-robot scalability of the proposed method.

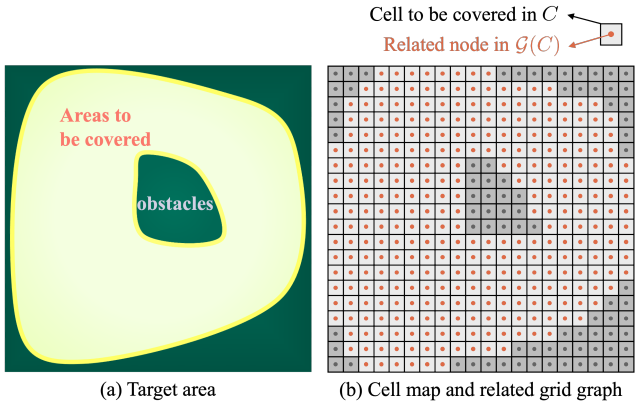
The rest of this paper is organized as follow: Section II describes the model for shortening path in arbitrary environments using grid-based methods. Section III details the foundational principles and specific steps of our algorithm. Section IV presents the simulation results, showing the performance of our algorithm and its capability within multi-robot systems. Finally, Section V concludes the paper and discusses future works.

*This work was supported by the National Key R&D Program of China (2022YFB3207703) and the National Science Foundation of China under Grants 62088101 and 62003015. (Corresponding author: Lei Chen)

Yuxuan Tang, and Lei Chen are with Advanced Research Institute of Multidisciplinary Sciences, Beijing Institute of Technology, Beijing, China. Emails: {tangyx, bit_chen}@bit.edu.cn

Qizhen Wu is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. Email: wuqzh7@buaa.edu.cn

Chunli Zhu is with the State Key Laboratory of CNS/ATM, Beijing Institute of Technology, Beijing, China. Email: chunlizhu@bit.edu.cn



(a) Target area (b) Cell map and related grid graph

Fig. 1. Reflection of real region $\mathcal{R}$ using cell map C and grid graph $\mathcal{G}(C)$.

II. Problem Description

An ideal coverage path should enable a robot to thoroughly cover the area of interest while minimizing revisits to previously covered zones. This paper focuses on finding such a balance for algorithms based on ACD. In this section, we are dedicated to constructing a problem model that minimizes path duplication while guaranteeing full coverage in arbitrary regions under grid-based system.

Adopting a setup similar to the one in [9], we assume the robot carries a square tool with side length D, which remains directionally fixed during movement. The target area $\mathcal{R}$ is mapped using square cells with side length D to represent its size and shape. The cell map $C(\mathcal{R})$ is employed to coarsely depict the target area, constituting a set that includes all cells awaiting coverage. For simplicity, we refer to it as C when not specified. In general, each cell c_i in C corresponds to a robot path point r_i, with coordinates $p_i(x, y)$, abbreviated as p_i, where i is the index of the corresponding cell. Set ρ comprises all path points of robots. Fig.1 shows the grid mapping relationship within an arbitrary area.

To achieve complete coverage and minimal repetition simultaneously, it is necessary to find a sequence of path point $[r_0, r_1, \ldots, r_N]$ such that,

$$\min \left\| [r_0, r_1, \ldots, r_N] \right\|, \tag{1}$$

$$\text{s.t.} \quad r_i \in C, \quad \forall i = 1, 2, \ldots, N, \quad \{r_0, r_1, \ldots, r_N\} \supseteq \rho, \tag{2}$$

where $\left\| [r_0, r_1, \ldots, r_N] \right\|$ represents the length of the path formed by sequentially connecting each path point. (1) indicates the goal of minimizing the total length of the coverage path. (2) specifies that the path should pass through every path point.

(1) and (2) present a model to optimize coverage paths in grid systems, but finding the optimal path through all points is computationally challenging. Given that most cells in the cell map are arranged regularly, the grid graph $\mathcal{G}(C)$ is constructed through C, with each cell corresponding to a node in $\mathcal{G}(C)$. This is visually represented in Fig.1. Since each Hamiltonian graph has a cycle that uniquely covers all nodes, locating such a graph in $\mathcal{G}(C)$ that includes the majority of nodes enables full area coverage with fewer repetitive paths. Therefore, we redefine the problem as finding the largest Hamiltonian sub-graph in $\mathcal{G}(C)$.

To begin with, we define the notion of a Hamiltonian sub-graph. For a given graph Σ, if there exists a graph Π satisfying,

$$\begin{cases} E(\Pi) \subseteq E(\Sigma), \\ V(\Pi) \subseteq V(\Sigma), \\ \Pi \text{ is a Hamiltonian graph}, \end{cases} \tag{3}$$

then the graph Π is called a Hamiltonian sub-graph of Σ, denoted as $\Pi \sim \Sigma$, where $E(*)$ refers to the edge set and $V(*)$ to the vertex set of the graph.

To find the largest Hamiltonian sub-graph in the given graph $\mathcal{G}(C)$, that is, for the sub-graph Z of $\mathcal{G}(C)$,

$$\textit{maximize} \quad |V(Z)|, \tag{4}$$

$$\text{s.t.} \quad Z \sim \mathcal{G}(C), \tag{5}$$

where $|V(*)|$ represents the order of a graph. Since a Hamiltonian cycle ensures the traversal of each node in the graph without repetition, (4) and (5) essentially aim to find a path that doesn't repeat and covers as many cells as possible.

III. Our Algorithm

To precisely reflect the intended coverage area on the cell map, it is imperative to locate the actual waypoints the robot traverses for each cell, ensuring the robot comprehensively covers each cell. The following introduces the method used in this study, which is simple and effective.

Based on what is depicted in Section II, the area to be covered inside a cell c_i is denoted as ζ_i, and the cell's square outline is represented as ct_i. We introduce Γ_i to obtain p_i, where Γ_i represents the range of the robot's working parts when positioned at point r_i. The function $A(*)$ denotes the area of region $*$. When ζ_i has only one connected component, the position of Γ_i follows the rules of

$$\textit{minimize} \quad A(\Gamma_i \cap ct_i), \tag{6}$$

$$\text{s.t.} \quad \Gamma_i \cap \zeta_i = \zeta_i. \tag{7}$$

(6) and (7) stipulate that the robot's working components to envelop the target area precisely. Fig.2(a) shows the examples of determining the robot's path points within a cell according to the equations. The path point r_i related to c_i is situated at the geometric center of Γ_i. A cell c_i is deemed *unoccupied* only if it contains a singular path point located precisely at the centroid ct_i, any deviation classifies it as *occupied*. Furthermore, if all four cells contained within a mega-cell are complete, then the mega-cell itself is deemed *complete*.

Exceptionally, for cases where ζ_i is a multi-connected domain or the geometric center of Γ_i lies outside the area $\mathcal{R}$, multiple path points $r_{i1}, r_{i2}, \ldots$, are utilized to cover c_i. Fig.2(b) and (c) demonstrate the methods for handling these

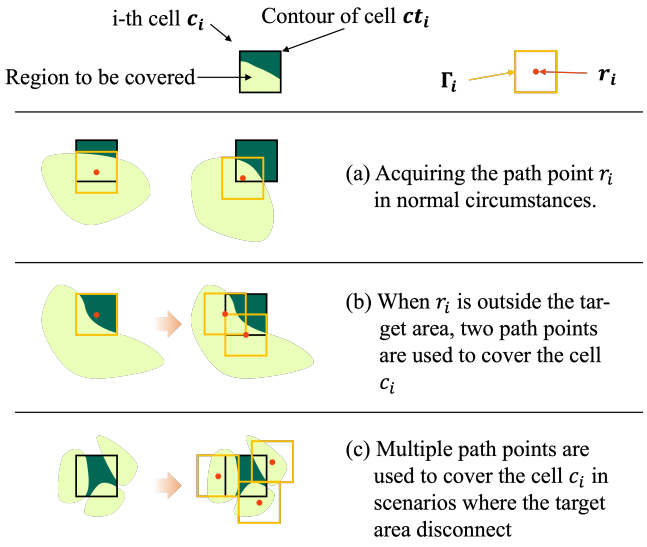


Fig. 2. Diagram demonstrating the process for creating path points. (a) The path points obtained under normal conditions according to (6) and (7). (b) A connected grid area where path points, based on (6) and (7), are outside the target area. Choosing endpoints of two grid edges as path points generally ensures coverage. (c) A grid with a disconnected coverage area, where path points for each connected domain are determined using (6) and (7).

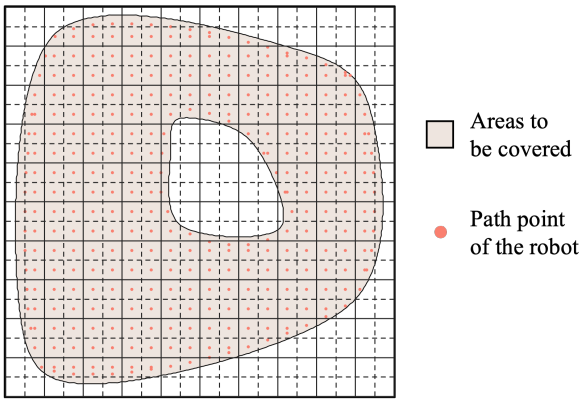


Fig. 3. The distribution of path points within the sample area in Fig.1.

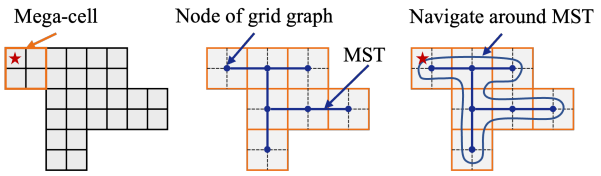


Fig. 4. Graphic representation of the coverage path generated by the STC algorithm.

unusual situations. Regarding the sample area presented in Fig.1, Fig.3 displays the spread of path points across each cell as derived from the outlined criteria.

Planning efficient coverage paths within the cell map is the task of addressing the issue depicted in (4) and (5). However, it has been proven that detecting a Hamiltonian cycle in a grid graph is NP-complete, and to date, no universal algorithm exists that can identify the presence of a Hamiltonian cycle in every grid graph within polynomial time [24]–[26]. Therefore, adopting a simplified approach to find an decent path that is acceptable in most cases is our choice.

We present the STC algorithm as the inspiration for our work. Grouping four cells into mega-cells, the basic STC algorithm uses the mega-cells as nodes to construct a Minimum spanning tree (MST), around which a circular navigation path is developed. Methods like depth first search or PRIM [27] are used to form spanning tree. Fig.4 offers a visual schematic of this process.

Given that the STC algorithm allows for the straightforward acquisition of a Hamiltonian cycle in regions of particular shapes, it's feasible to extend these cycles to create Hamiltonian cycles within more expansive areas. Building on this concept, our endeavor involves refining the conventional STC algorithm by incorporating several rules. We adopt a specific approach based on the complete mega-cells area and seeks to identify the largest possible Hamiltonian sub-graph within the grid graph $\mathcal{G}(C)$. We begin with the Hamiltonian cycle specified by the STC algorithm and progressively extending it to cover all areas. The reasons for the effectiveness of this strategy are explained below.

Several definitions are made as follows. (u, v) represents an edge composed of vertices u and v. Moreover, we use $l(v_s(H), v_t(H))$ to represent a Hamiltonian path starting and ending at vertices $v_s(H)$ and $v_t(H)$, denoted as $l(H)$ for short, where the graph H includes exactly all the vertices of $l(H)$. Besides, since severing a Hamiltonian cycle at any position yields a Hamiltonian path, when $l(H)$ represents a Hamiltonian cycle rather than a path, $v_s(H)$ and $v_t(H)$ denote any connected vertex pair within the said cycle.

Assume that in a given grid graph G_0, there exists a *trunk graph* $\mathcal{T}$ that satisfies $\mathcal{T} \sim G_0$ and contains a Hamiltonian cycle HC_t. Then the *branch graph* $\mathcal{B}$ can be obtained from $\mathcal{B} = G_0 - \mathcal{T}$. If a sub-graph of $\mathcal{B}$ referred to as $\mathcal{B}'_x$ has a Hamiltonian path $l(v_s(\mathcal{B}'_x), v_t(\mathcal{B}'_x))$ satisfying

$$\begin{cases} (\xi_x, \kappa_x) \in E(HC_t), \\ (\xi_x, v_s(\mathcal{B}'_x)) \in E(G_0), \\ (\kappa_x, v_t(\mathcal{B}'_x)) \in E(G_0), \end{cases} \tag{8}$$

then the graph integrated from $\mathcal{T}$ and $\mathcal{B}'_x$ must contain at least one Hamiltonian cycle, where x is the subscript indicating the x-th sub-graph of $\mathcal{B}$ and ξ_x, κ_x represent two vertices customized for $\mathcal{B}'_x$. Fig.5 illustrates the relationship between involved grids.

The aforementioned discourse illustrates how to extend a certain Hamiltonian cycle HC_t of the trunk graph $\mathcal{T}$. In practice, $\mathcal{T}$ may contain multiple Hamiltonian cycles. It describes the potential extension relationships for the trunk graph $\mathcal{T}$ as below,

$$\begin{cases} (\xi_x, \kappa_x) \in E(\mathcal{T}), \\ (\xi_x, v_s(\mathcal{B}'_x)) \in E(G_0), \\ (\kappa_x, v_t(\mathcal{B}'_x)) \in E(G_0). \end{cases} \tag{9}$$

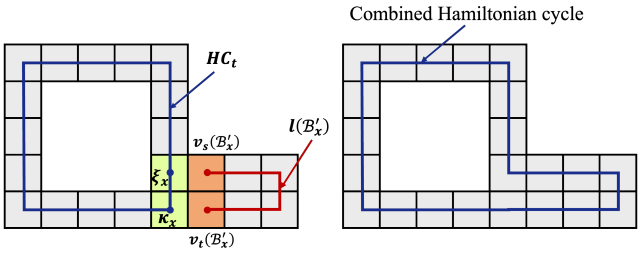


Fig. 5. Illustration of how to extend a Hamiltonian cycle in a grid graph. By severing the existing connection between grids ξ_x and κ_x, and forming new links from ξ_x to $v_s(\mathcal{B}'_x)$ and κ_x to $v_t(\mathcal{B}'_x)$, the original Hamiltonian cycle, HC_t, is expanded into a larger graph.

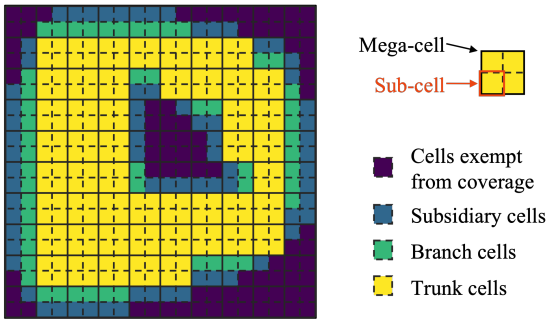


Fig. 6. The classification of cells within the sample area in Fig.1.

Notably, although the Hamiltonian path $l(\mathcal{B}'_m)$ may satisfy (9), it does not necessarily lead to an expansion of $\mathcal{T}$ as $(v_s(\mathcal{B}'_m), v_t(\mathcal{B}'_m))$ may not be an edge of any Hamiltonian cycle.

We present a low-repetitive coverage path planner based on the strategy mentioned above. A classification of cells awaiting coverage is conducted initially. To start with, all occupied cells and cells with multiple path points, known as subsidiary cells, are recorded in the set $\mathcal{S}$. Additionally, as 2-connectivity is a prerequisite for Hamiltonian cycles, cells connecting with only one other cell are also added to the set $\mathcal{S}$. Finally, cells capable of forming complete mega-cells are classified as *trunk cells*. Grid graph $\mathcal{T}$ is used to represent trunk cells. Excluding the trunk and subsidiary cells, the remaining cells are classified as *branch cells*, related to the grid graph $\mathcal{B}$. Regarding the sample area presented in Fig.1, Fig.6 depicts the categorization of each cell as determined by the specified method.

Since $\mathcal{S}$ includes cells that cannot form a grid graph and cells challenging to incorporate into a Hamiltonian graph, our task becomes finding the largest Hamiltonian sub-graph within $\mathcal{G}(\mathcal{T} \cup \mathcal{B})$. As described in [9], it's possible to generate non-repetitive coverage paths within the connected areas of mega-cells, ensuring that a Hamiltonian cycle based on the STC algorithm can be achieved in the trunk graph $\mathcal{T}$. The task then becomes identifying all Hamiltonian path patterns in the branch graph $\mathcal{B}$ that conform to (9). Several simple Hamiltonian cycle structures are engaged in matching with the branch graph. This method is effective because the Hamiltonian sub-graphs within the branch graph predominantly manifest in two forms: single-layer circular nodes and double-layer striped nodes. This distinct feature eases the recognition process of these structures.

To elaborate explicitly, given that the 2-connectivity of a graph is a prerequisite for the presence of Hamiltonian cycles, single-layer grid graphs are required to form a ring structure to meet this fundamental condition. Consequently, multi-layer stacked grid graphs are more inclined to encompass Hamiltonian cycles. Nevertheless, in the branch graph, any solid graph size of 3×3 or greater is unfeasible, as it invariably includes at least one complete mega-cell. Therefore, the $2 \times n$ $(n \geq 2)$ grid graph, containing a Hamiltonian cycle consistently, is the largest solid grid structure possible in the branch graph.

In cases where a Hamiltonian cycle is not possible, we resort to the most basic Hamiltonian path structure, which comprises pairs of adjacent vertices. Hence, our primary task is to sequentially identify several types of graph structures within the branch graph, in the order of rings, $2 \times n$ $(n \geq 2)$ grid graphs, and vertex pairs, ensuring no vertex is reused. Details and examples for searching these special structures are shown in Fig.7. We use $\mathcal{H} = \{\mathcal{B}'_1, \mathcal{B}'_2, \ldots\}$ to record all the sub-graphs in $\mathcal{B}$ that contain these Hamiltonian structures. Grids present in $\mathcal{B}$ yet excluded from $\mathcal{H}$ should be incorporated into $\mathcal{S}$.

Moreover, to fulfill the stipulations of (8), it's essential to adjust the Hamiltonian cycle within the trunk graph. Recognizing that Hamiltonian cycle is feasible in the trunk graph via a spanning tree, if $v_s(\mathcal{B}'_n)$ and $v_t(\mathcal{B}'_n)$ in $l(\mathcal{B}'_n)$, adhering to (9), are connected to adjacent trunk graph nodes in separate mega-cells, then these mega-cells must be linked in the spanning tree. Should this particular connection type breach the criteria for spanning tree generation, all nodes involved in $\mathcal{B}'_n$ are to be regarded as subsidiary cells.

Following the discussion, we introduce an effective method for swiftly planning a non-repetitive coverage path that is capable of covering all the trunk cells and a major portion of the branch cells. Furthermore, to ensure full coverage, the path needs to be expanded to encompass all subsidiary cells contained within $\mathcal{S}$.

In particular, we transfer the derived Hamiltonian cycle HC to the cell map C using $\mathcal{L}(HC)$. The $\mathcal{L}(HC)$ represents a trajectory following the robot path points, with each point relating to a vertex in HC. To reduce redundancy, we insert a certain remained path point r_{s_x} between two consecutive path points in $\mathcal{L}(HC)$, and the break point is located by

$$\operatorname{argmin} \quad (\|\Delta(\mathcal{L}(HC), r_{s_x})\| - \|\mathcal{L}(HC)\|), \qquad (10)$$

where r_{s_x} indicates the x-th path point related to cells in $\mathcal{S}$, and $\Delta(\mathcal{L}(HC), r_{s_x})$ represents combination result of $\mathcal{L}(HC)$ and r_{s_x}. This process is repeated until all path points are incorporated into the coverage path.

In conclusion, the specific steps of the proposed low-repetitive coverage path planner are shown in Algorithm 1.

IV. Simulation Results

To validate the effectiveness of the proposed method, numerous simulation experiments are conducted under Python

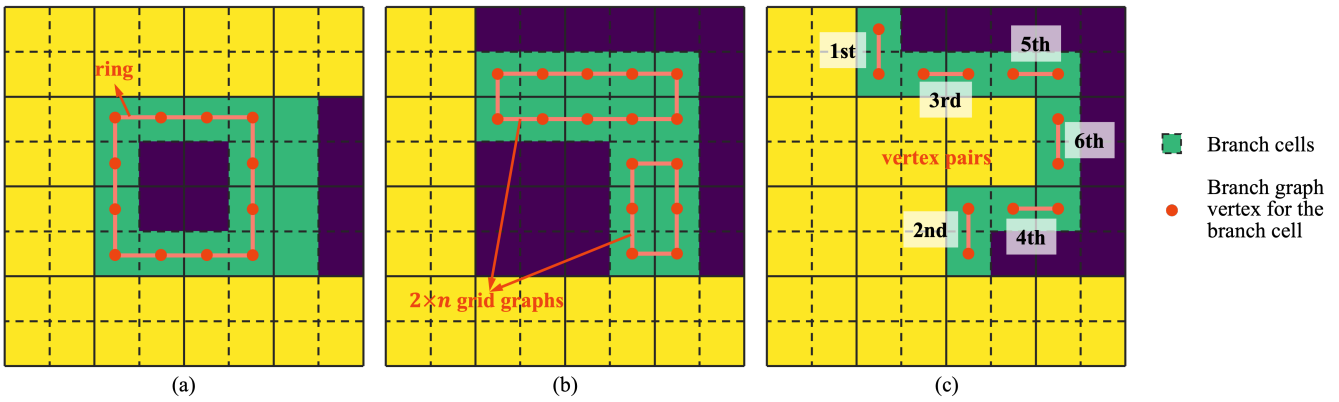


Fig. 7. Examples of searching Hamiltonian structures. In (a), an example is presented of searching for the smallest rings that capable of enclosing the holes within the branch graph. (b) shows the detection of all $2 \times n$ grid graphs in the branch graph. In the absence of Hamiltonian cycle structures, vertex pairs are marked in sequence, focusing first on vertices with a single connection in the branch graph, as shown in (c), which outlines the approach for identifying and sequencing vertex pairs in the branch graph.

Algorithm 1: Low-repetitive coverage path planner

Input: $\mathcal{T}, \mathcal{S}, \mathcal{H}, C$
Output: A low-repetitive path that traverses all path points
$k \leftarrow$ number of Hamiltonian structures in $\mathcal{H}$
$\mathcal{E} \leftarrow \emptyset$; `// rules of tree formation`
do
  for $j \leftarrow 1$ **to** k **do**
    if $\mathcal{B}'_j \notin \mathcal{T}$ ***and*** $l(\mathcal{B}'_j)$ *fits (9)* **then**
      $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{B}'_j$
      $E_j \leftarrow$ extra rules due to the link
while $\mathcal{T}$ *has been augmented with new parts*;
$ST \leftarrow$ an MST formed by PRIM in the mega-cell grid graph, prioritizing all rules in $\mathcal{E}$;
while *any* E_x *in* $\mathcal{E}$ *mandates using cells that are already utilized in the MST.* **do**
  $\mathcal{T} \leftarrow \mathcal{T} - \mathcal{B}'_x$
  $\mathcal{S} \leftarrow$ cells in $\mathcal{B}'_x$
$HC_t \leftarrow$ circular path in $\mathcal{T}$ obtained through ST
$HC_t \leftarrow$ connect all Hamiltonian paths complying with PRIM to HC_t according to (8)
$L \leftarrow \mathcal{L}(HC_t)$
$m \leftarrow$ number of remained path points
for $n \leftarrow 1$ **to** m **do**
  $\Delta(L, r_{s_n}) \leftarrow \text{argmin}\ (\|\Delta(L, r_{s_n})\| - \|L\|)$
  $L \leftarrow \Delta(L, r_{s_n})$
return L

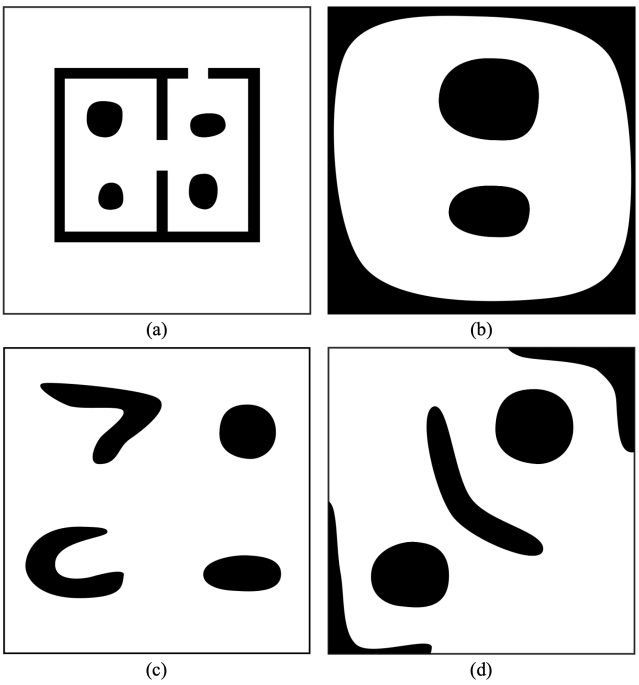


Fig. 8. Diverse experimental settings. (a) An indoor environment. (b) An environment with an irregular lawn. (c) and (d) Arbitrary environments filled with concave and convex obstacles. All the simulation environments are squares with a side length of 6m, and the robot tool size is 0.2m.

environment. Specifically, the algorithm is tested under three different scenarios. Firstly, inspired by service robots like vacuum cleaners, a simulation is carried out in an office-like environment as shown in Fig.8(a). Secondly, to replicate the working environment of a weeding robot, a simulation is conducted in an irregularly shaped area as depicted in Fig.8(b). Lastly, to assess the adaptability of the algorithm in diverse conditions, a simulation is performed in an area with several concave and convex obstacles, as illustrated in Fig.8(c) and (d). The test areas are all $6m \times 6m$ square regions, with the robot tool size D being $0.2m$.

To verify the superiority of the proposed algorithm, we compare it with the one introduced in [16]. Based on BCD, this algorithm generates efficient complete coverage paths in complex environments. We select path length as the comparison metric. Fig.9 displays the coverage paths generated by the two algorithms in the aforementioned environments, and Table I provides the quantitative comparison results. The results indicate that in all scenarios, the coverage paths

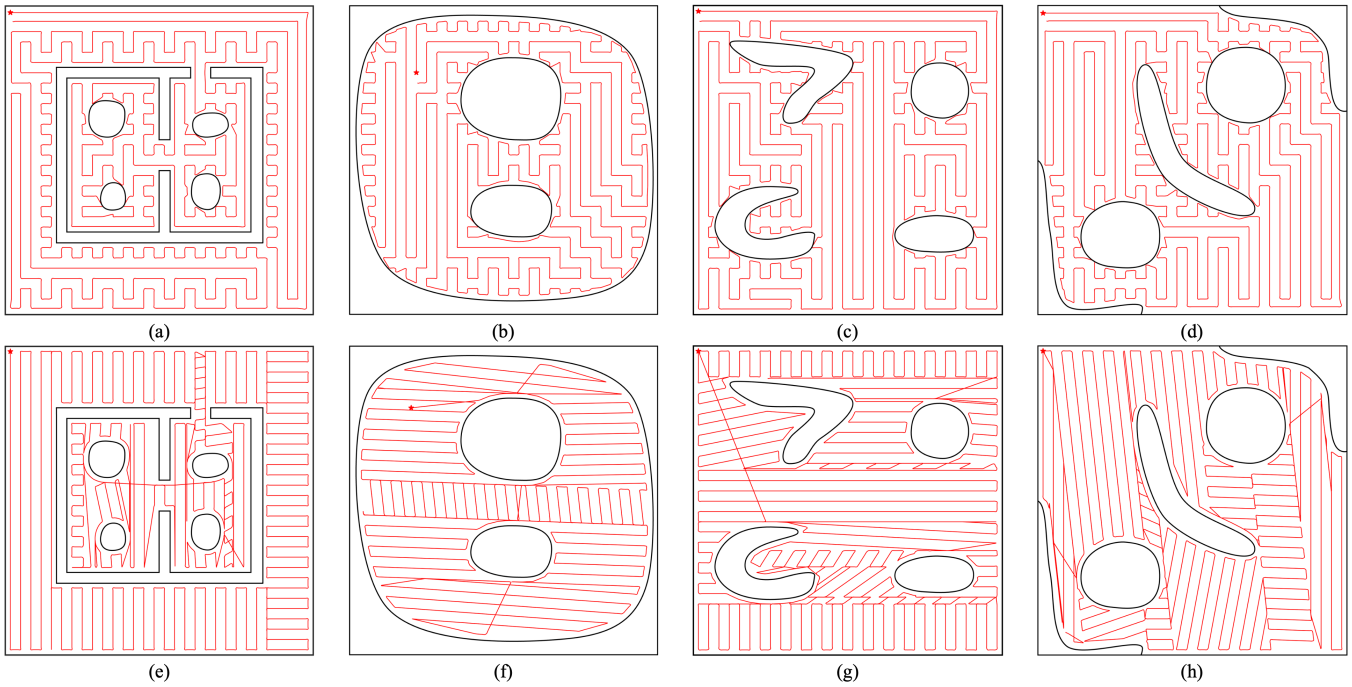


Fig. 9. Coverage paths generated by the two planners. (a)-(d) represent paths generated by our planner, and (e)-(h) represent paths generated by the planner in [16].

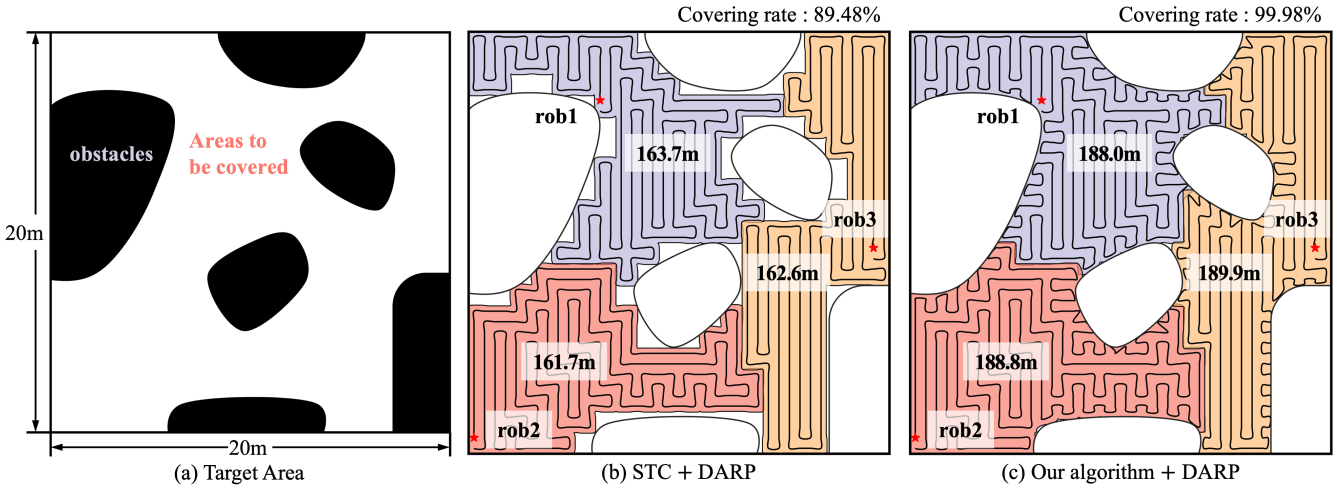


Fig. 10. Experimental setting and comparison result for the multi-robot coverage task. (a) An arbitrary environment of $20m \times 20m$, with each robot's tool size being $0.5m$. (b) The results of multi-robot coverage path planning combining STC and DARP. (c) The results of multi-robot coverage path planning combining our algorithm with DARP. Annotations in the result figures display the path lengths of the robots and the coverage rate.

generated by our algorithm are shorter than those produced by the algorithm in [16].

TABLE I
PATH LENGTH COMPARISON OF TWO PLANNERS

Scenario	Our planner's path length (m)	[16] planner's path length (m)	Path reduction by our planner
Fig8.(a)	158.83	187.33	15.20%
Fig8.(b)	127.17	143.09	11.13%
Fig8.(c)	160.31	202.89	20.99%
Fig8.(d)	145.63	175.37	16.96%

Additionally, one of the key goals of this study is to adapt the STC-based multi-robot coverage planning algorithm for the use in arbitrarily shaped regions. The DARP algorithm is a multi-robot coverage path planning algorithm based on the STC algorithm. It partitions the area to be covered into several disjoint sections, maintaining a similar number of cells in each section to ensure that the path length for each robot during coverage remains roughly equal [18]. To assess the suitability of the proposed algorithm for multi-robot coverage planning, we employ a combination of the DARP algorithm with our proposed coverage path planner. Notably, during the cell allocation process, incomplete mega-

cells are adjusted by a factor that indicates their area as a fraction of a complete mega-cell. The comparison of results is shown in Fig.10.

As shown in Fig.10(a), within a $20m \times 20m$ square area filled with randomly placed irregular obstacles, three robots are scattered, each with a tool size D of $0.5m$. Fig.10(b) showcases the pathways forged through the synergy of the DARP and STC algorithms, while Fig.10(c) reveals the paths shaped by integrating the DARP algorithm with our approach. The results show that while the collaboration between the DARP and STC algorithms manages to cover the majority of the area with comparable path lengths for each robot, it stops short of achieving complete area coverage. In contrast, the fusion of the DARP algorithm with our strategy not only maintains near-uniform path lengths across all robots but also secures comprehensive coverage of the entire area.

V. Conclusion

We propose a method for planning efficient complete coverage paths for irregularly shaped terrains. To leverage grid-based methods for low redundancy and ensure computational feasibility, we first derive a model for finding the largest Hamiltonian sub-graph, aiming to reduce the lengths of route traversing all cells. Based on the model, a Hamiltonian cycle expansion strategy and a low-repetitive coverage path planner are then proposed. Despite the compromise in path lengths superiority, the proposed method facilitates the quick attainment of considerable results. Simulation results validate the proposed method's coverage capability in arbitrary regions and its scalability in multi-robot systems.

In the future, we plan to develop a more comprehensive coverage planning framework encompassing all special circumstances such as narrow corridors and closely clustered obstacles. In addition, we will extend the proposed method to real-world applications.

References

[1] Y. Yun, L. Hou, Z. Feng, W. Jin, Y. Liu, H. Wang, R. He, W. Guo, B. Han, B. Qin, *et al.*, "A deep-learning-based system for indoor active cleaning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7803–7808.

[2] A. S. A. Ghafar, S. S. H. Hajjaj, K. R. Gsangaya, M. T. H. Sultan, M. F. Mail, and L. S. Hua, "Design and development of a robot for spraying fertilizers and pesticides for agriculture," *Materials Today: Proceedings*, vol. 81, pp. 242–248, 2023.

[3] D. Albani, D. Nardi, and V. Trianni, "Field coverage and weed mapping by uav swarms," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Ieee, 2017, pp. 4319–4325.

[4] M. Ghaffari, D. Manthena, A. Ghaffari, and E. L. Hall, "Mines and human casualties: a robotics approach toward mine clearing," in *Intelligent Robots and Computer Vision XXII: Algorithms, Techniques, and Active Vision*, vol. 5608. SPIE, 2004, pp. 306–312.

[5] N. Li, Z. Su, H. Ling, M. Karatas, and Y. Zheng, "Optimization of air defense system deployment against reconnaissance drone swarms," *Complex System Modeling and Simulation*, vol. 3, no. 2, pp. 102–117, 2023.

[6] R. Hewett and S. Puangpontip, "On controlling drones for disaster relief," *Procedia Computer Science*, vol. 207, pp. 3703–3712, 2022.

[7] I. A. Hameed, "Coverage path planning software for autonomous robotic lawn mower using dubins' curve," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2017, pp. 517–522.

[8] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *2010 IEEE International conference on robotics and automation*. IEEE, 2010, pp. 5525–5530.

[9] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of mathematics and artificial intelligence*, vol. 31, pp. 77–98, 2001.

[10] X. Chen, T. M. Tucker, T. R. Kurfess, and R. Vuduc, "Adaptive deep path: efficient coverage of a known environment under various configurations," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3549–3556.

[11] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, pp. 113–126, 2001.

[12] 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.

[13] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[14] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Optimal complete terrain coverage using an unmanned aerial vehicle," in *2011 IEEE International conference on robotics and automation*. IEEE, 2011, pp. 2513–2519.

[15] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, "Efficient multi-robot coverage of a known environment," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1846–1852.

[16] R. Bähnemann, N. Lawrance, J. J. Chung, M. Pantic, R. Siegwart, and J. Nieto, "Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem," in *Field and Service Robotics: Results of the 12th International Conference*, 2021, pp. 277–290.

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

[18] 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.

[19] W. Dong, S. Liu, Y. Ding, X. Sheng, and X. Zhu, "An artificially weighted spanning tree coverage algorithm for decentralized flying robots," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1689–1698, 2020.

[20] J. Lu, B. Zeng, J. Tang, T. L. Lam, and J. Wen, "Tmstc*: A path planning algorithm for minimizing turns in multi-robot coverage," *IEEE Robotics and Automation Letters*, pp. 5275–5282, 2023.

[21] Y. Gabriely and E. Rimon, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry*, vol. 24, no. 3, pp. 197–224, 2003.

[22] T. Ranjitha and K. Guruprasad, "Pseudo spanning tree-based complete and competitive robot coverage using virtual nodes," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 195–200, 2016.

[23] H. V. Pham, P. Moore, and D. X. Truong, "Proposed smooth-stc algorithm for enhanced coverage path planning performance in mobile robot applications," *Robotics*, vol. 8, no. 2, pp. 1–19, 2019.

[24] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter, "Hamilton paths in grid graphs," *SIAM Journal on Computing*, vol. 11, no. 4, pp. 676–686, 1982.

[25] R. I. Nishat, "Reconfiguration of hamiltonian cycles and paths in grid graphs," Ph.D. dissertation, University of Victoria, 2020.

[26] F. Keshavarz-Kohjerdi and A. Bagheri, "A linear-time algorithm for finding hamiltonian cycles in rectangular grid graphs with two rectangular holes," *Optimization Methods and Software*, pp. 1–35, 2023.

[27] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.