2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Optimal Multi-Robot Planning for Simultaneous Area and Line Coverage

Tianyuan Zheng, Kaiyan Yu, Mingyang Gao, and Jingang Yi

***Abstract*— Robotic coverage tasks often require teams of robots to not only survey regions of interest but also trace and interact with linear features such as cracks, seams, or pipelines. We term this the double coverage problem, where robots must balance two competing roles: wide-area exploration for inspection and precise trajectory following for servicing linear structures. This paper develops an optimal multi-robot planning framework that unifies area coverage and line servicing. We formulate a topological analysis in manifold space and introduce the hierarchical cyclic merging regulation (HCMR) method, for which optimality under a fixed sweep direction is proven. The framework is experimentally validated for a multi-robot crack survey and filling application. Benchmark comparisons demonstrate that HCMR reduces planned path length by at least 10.0%, shortens task completion time by at least 16.9%, and ensures complete crack coverage with virtually conflict-free operation, outperforming state-of-the-art coverage planners. These results highlight the feasibility and efficiency of deploying topology-informed multi-robot planning for practical inspection and repair scenarios.**

I. Introduction

Coverage is a fundamental capability for autonomous robots, underpinning tasks such as infrastructure inspection, environmental monitoring, and agricultural surveying. Two canonical formulations are commonly studied: area coverage, which scans a region, and line coverage, which traces elongated features such as pipelines, roads, or cracks (Fig. 1). Area coverage has been addressed through grid-based and exact approaches [1]. Grid-based methods [2] discretize the workspace, enabling learning-based algorithms but suffering from resolution sensitivity and inefficiency in complex geometries. Exact methods, such as Morse-based cellular decomposition and boustrophedon decomposition [3], [4] partition environments at critical topological points, often using Reeb graphs to structure traversal. While these methods guarantee full coverage, their reliance on specific Eulerian tours can lead to suboptimal paths, and extensions to multi-robot settings [5], [6] remain limited. Line coverage is typically formulated over graph abstractions of feature networks. Road network coverage has been modeled using mixed-integer linear programming [7], while heuristic methods incorporate turning costs and nonholonomic constraints [8].

Little prior research work has solved the multi-robot double coverage (MDC) problem, where robots must simultaneously explore regions and service linear features. Prior studies primarily focused on near-optimal strategies for a single robot in known or unknown environments [9], [10]. This gap is significant because many real-world applications, from bridge deck inspection [11] to agricultural monitoring [12], require both area exploration and feature servicing within the same mission. The goal of this study is to solve MDC in known environments. While Bayesian optimization and consensus-based approaches were initially explored, they often caused conflicts and inefficiencies without strong theoretical guarantees. To establish a rigorous foundation, we study MDC in known environments and introduce hierarchical cyclic merging regulation (HCMR).

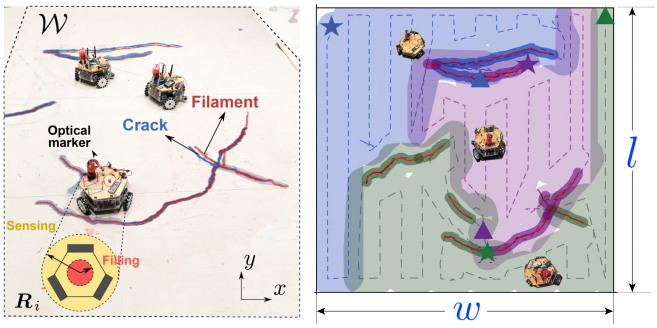

Fig. 1. An example of the MDC problem. Left: real-world experiment, where blue curves denote cracks and three omni-wheel robots coordinate using wide sensing regions and filling footprints. Right: statistics on the same map, where black curves denote cracks and a robot team $\mathcal{R}$ ($N = 3$) ensures complete coverage of $\mathcal{W}$. Robot start and end points are marked by ★ and ▲, respectively. Distinct colors indicate individual robot trajectories.

The HCMR method uses Morse-theoretic analysis to constrain the search space to Morse-bounded paths. The key insight is that area coverage and linear-feature servicing must be treated as a single continuous domain. In Euclidean space, naive attachment of coverage cells and feature segments can break boundaries into disjoint points, making trajectory continuity and topological complexity quantification difficult. By mapping coverage and feature domains to manifolds via quotient and stitching operations, we transform the problem into a well-defined framework suitable for optimization. The HCMR also preserves connectivity and enables continuous, conflict-free trajectories. Building on HCMR, we further develop a control framework coordinating multi-robot motion and nozzle actuation for accurate crack following. The experimental validation demonstrates reductions in path length and task time while maintaining scalability under practical constraints.

The main contributions of this work are as follows. We formulate a novel MDC problem that unifies area exploration and linear servicing. We present the first topological analysis of MDC in manifold space, providing the foundation for all results, and prove that path optimality is equivalent to Morse boundedness. A connection between topology and

T. Zheng and J. Yi are with the Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854 USA (email: tz270@scarletmail.rutgers.edu; jgyi@rutgers.edu).

K. Yu is with the Department of Mechanical Engineering, Binghamton University, Binghamton, NY 13902 USA (email: kyu@binghamton.edu).

M. Gao is with Department of Aerospace Engineering, University of Illinois, Urbana, IL 61801 USA (email: mgao26@illinois.edu).

graph theory is established for the HCMR algorithm, which yields complete coverage with provably optimal solutions.

II. Problem Formulation

A. Notations and Preliminaries

We represent a graph as $\mathbb{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is the vertex set and $\boldsymbol{E} \subset \boldsymbol{V} \times \boldsymbol{V}$ the edge set. For $\boldsymbol{S} \subset \boldsymbol{V}$, $\mathbb{G}/\boldsymbol{S}$ denotes the *vertex contraction* of $\boldsymbol{S}$ and $\mathbb{G}[\boldsymbol{S}]$ the *induced subgraph*. The *cycle* and *bridge* edge sets are denoted by $\boldsymbol{E}_{\text{cycle}}$ and $\boldsymbol{E}_{\text{bridge}}$, respectively. Two cycles are *vertex-adjacent* (resp. *edge-adjacent*) if they share a vertex (resp. an edge). For cycles C_1, C_2, the *symmetric difference* is $C_1 \triangle C_2 = (C_1 \setminus C_2) \cup (C_2 \setminus C_1)$. A *cycle basis* $\mathcal{K}$ is a set of cycles generating all cycles in $\boldsymbol{E}_{\text{cycle}}$ via symmetric differences. For a manifold X, $\text{int}(X)$ and ∂X denote its interior and boundary, $\mu(\cdot)$ the Lebesgue measure, and *almost everywhere* (a.e.) means except on a set of measure zero. A *quotient space* $X/\sim$ identifies points of X under an equivalence relation $\sim$, and $\mathcal{M}_1 \sqcup \mathcal{M}_2$ denotes the disjoint union of manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$.

B. MDC Problem Formulation

We consider MDC in a rectangular workspace $\mathcal{W} \subset \mathbb{R}^2$ of size $l \times w$, where a known set of linear features $\mathcal{C}$ (e.g., cracks) need to be simultaneously explored and serviced. A team of N holonomic robots $\mathcal{R} = \{R_1, \ldots, R_N\}$ is deployed, each operating in two task modes: exploration (T_e) and service (T_s), with coverage footprints $\mathcal{H}$ (radius r) and $\mathcal{F}$ (radius $a \ll r$), respectively. Let $\boldsymbol{\pi}_i$ denote the trajectory of robot R_i, $i = 1, \ldots, N$, and $T_x \in \{T_e, T_s\}$ be the task at point $\boldsymbol{x} \in \boldsymbol{\pi}_i$. The MDC constraints are:

$$\bigcup_{i=1}^{N} \bigcup_{\boldsymbol{x} \in \boldsymbol{\pi}_i} (\boldsymbol{x} \oplus \mathcal{H}) \supset \mathcal{W}, \quad \bigcup_{i=1}^{N} \bigcup_{\boldsymbol{x} \in \boldsymbol{\pi}_i} (\boldsymbol{x} \oplus \mathcal{F}) \mathbb{I}(T_x = T_s) \supset \mathcal{C},$$

where $\oplus$ denotes Minkowski sum and $\mathbb{I}(\cdot)$ is the indicator function. To avoid collisions, we impose:

$$\Big(\bigcup_{\boldsymbol{x} \in \boldsymbol{\pi}_i} \boldsymbol{x}\, \mathbb{I}(T_x = T_e) \Big) \cap \Big(\bigcup_{\boldsymbol{x}' \in \boldsymbol{\pi}_j} \boldsymbol{x}'\, \mathbb{I}(T_{x'} = T_e) \Big) = \emptyset,$$

$$\mu\Big(\Big(\bigcup_{\boldsymbol{x} \in \boldsymbol{\pi}_i} \boldsymbol{x}\, \mathbb{I}(T_x = T_s) \Big) \cap \Big(\bigcup_{\boldsymbol{x}' \in \boldsymbol{\pi}_j} \boldsymbol{x}'\, \mathbb{I}(T_{x'} = T_s) \Big) \Big) = 0,$$

for all $1 \leq i \neq j \leq N$. We assume a fully connected network for centralized decisions with complete knowledge of $\mathcal{C}$. The objective is to determine $\boldsymbol{\pi}_{\mathcal{R}} = \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N\}$ that $\min_{\boldsymbol{\pi}_{\mathcal{R}}} \Big(\sum_{i=1}^{N} L(\boldsymbol{\pi}_i) + \alpha \max_{1 \leq i \leq N} t_{f_i} \Big)$, where $L(\boldsymbol{\pi}_i)$ is the trajectory length, t_{f_i} is the finishing time of R_i, and $\alpha > 0$ balances coverage efficiency and team synchronization.

III. HCMR Planning Algorithm

In MDC, robots must switch between exploring open areas and servicing linear features (e.g., cracks). In the Euclidean workspace, these domains are geometrically disjoint, making attachment analysis insufficient. To address this, we represent each domain as a manifold cell and use quotient mappings to "glue" them into a unified manifold. This manifold perspective enables rigorous reasoning about connectivity and continuity across domains, forming the basis of the proposed HCMR algorithm.

A. Linear Graph and Morse Decomposition

We compute optimal routes using graph-based methods to ensure complete coverage and feasible linear servicing. The linear feature set $\mathcal{C}$ is partitioned into disjoint branches $\{\boldsymbol{b}\}$, with endpoints $\boldsymbol{V}_m$. Each branch $\boldsymbol{b}$ is replaced by a minimal-node path $\boldsymbol{\pi}(\boldsymbol{b}) = (n_1, \ldots, n_m)$ whose a-dilation satisfies $\boldsymbol{\pi}(\boldsymbol{b}) \oplus \mathcal{F} \supseteq \boldsymbol{b}$, as realized by LineOfVisibility in [9]. This reduces graph complexity while preserving servicing coverage. The union defines the crack graph $\mathbb{G}_c = (\boldsymbol{V}_c, \boldsymbol{E}_c)$.

When robots perform linear coverage, the sensed region is $\mathcal{A}_c = \bigcup_{\boldsymbol{e} \in \boldsymbol{E}_c} (\boldsymbol{e} \oplus \mathcal{H})$ and the unexplored region is $\mathcal{A}_w = \mathcal{W} \setminus \mathcal{A}_c$. Applying Morse decomposition on $\mathcal{A}_w$ yields the Reeb graph $\mathbb{G}_w = (\boldsymbol{V}_w, \boldsymbol{E}_w)$, with $\phi : \mathbb{R}^2 \to \boldsymbol{E}_w$ is the bijection mapping each cell to edge. The global graph is formed by first taking $\mathbb{G}_c \sqcup \mathbb{G}_w$, then linking each $v_m \in \boldsymbol{V}_m$ to its corresponding $v_w \in \boldsymbol{V}_w$ [9]. Pairing unmatched vertices yields the augmented graph $\mathbb{G}_{\text{aug}}$, and contracting all matched pairs in $\boldsymbol{V}_m$ produces the quotient graph $\mathbb{G}_{\text{sim}} = \mathbb{G}_{\text{aug}}/\sim$, on which Eulerian tours are constructed. In Fig. 2(a), combined with Morse decomposition, all vertices and green dotted edges form $\mathbb{G}_{\text{aug}}$. Fig. 2(b) shows the quotient graph $\mathbb{G}_{\text{sim}}$. Notably $(n_1, n_2) \oplus \mathcal{F}$ corresponds to edge (c_2, n_2).

A straightforward approach to optimizing trajectories of $\mathcal{R}$ would be to enumerate all possible Eulerian tours. However, this strategy is computationally intractable, as the number of such tours grows combinatorially with graph size [13].

B. Morse Bound and Morse Boundedness

Not all Eulerian tours yield optimal continuous coverage trajectories. Some introduce discontinuities when Boustrophedon decomposition is applied, requiring extra connections, overlaps, or detours. Morse boundedness addresses this by enforcing that the stitched manifold along a tour remains within its intrinsic topological limit, ensuring continuity across area and crack domains without artificial connections.

As shown in Fig. 3(a), the tour $\boldsymbol{e}_1 \to \boldsymbol{e}_2 \to \boldsymbol{e}_7 \to \boldsymbol{e}_3 \to \boldsymbol{e}_4 \to \boldsymbol{e}_8 \to \boldsymbol{e}_5 \to \boldsymbol{e}_6 \to \boldsymbol{e}_9 \to \boldsymbol{e}_{10}$ remains continuous without extra connections. In contrast, the alternative tour in Fig. 3(b) introduces discontinuities at $\boldsymbol{e}_1 \to \boldsymbol{e}_3$, $\boldsymbol{e}_4 \to \boldsymbol{e}_2$, $\boldsymbol{e}_2 \to \boldsymbol{e}_9$, and $\boldsymbol{e}_7 \to \boldsymbol{e}_{10}$, requiring additional connections and causing overlaps, uneven Boustrophedon spacing, and potential conflicts in multi-robot allocation. These examples illustrate an inherent limitation of working purely in $\mathbb{R}^2$: area and crack domains appear disjoint, and ad-hoc stitching collapses shared boundaries into fragmented point sets. As shown in Fig. 2(d), attaching cells $\phi^{-1}(e_1)$ and $\phi^{-1}(e_5)$ breaks their shared boundary into two disconnected points. By defining attachments via quotient maps, connectivity is preserved and boundaries remain well-defined, enabling formal analysis of continuity and complexity.

We map each $\mathbb{R}^2$ cell to a manifold cell via a quotient construction. To preserve the two conditions of Morse decomposition: (1) simple connectivity within each cell, and (2) smooth attachability at critical points, we introduce the quotient map $q_w : \mathcal{A}_w \to \mathcal{M}_w$. Boundary arcs that are glued in $\mathbb{R}^2$ are identified under q_w, and interior points remain

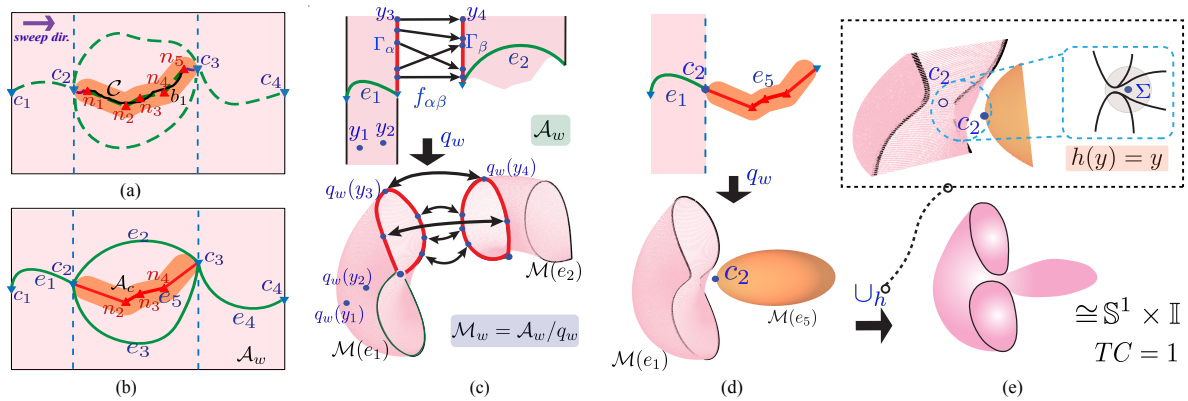

Fig. 2. An example of graph construction and its relationship with Morse theory. (a) The augmented graph, $\mathbb{G}_{\text{aug}}$, is constructed by integrating the Reeb graph and the crack graph. (b) The simplified graph, $\mathbb{G}_{\text{sim}}$, is obtained by contracting $\mathbb{G}_{\text{aug}}$ at critical points. (c) Construction of the Reeb manifold cells corresponding to e_1 and e_2 using the quotient map q_w. (d) One $\mathcal{M}_w$ manifold cell and one $\mathcal{M}_c$ manifold cell. (e) The attachment process of two manifold cells and the corresponding topological complexity.

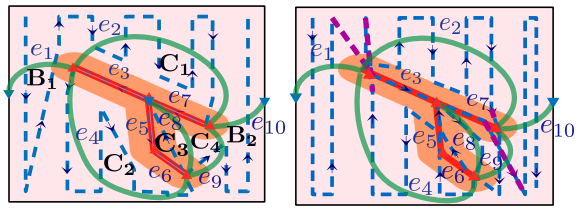

Fig. 3. Two trajectories for a single-robot double coverage task from Eulerian tours of $\mathbb{G}_{\text{sim}}$. Blue dotted lines show robot paths, green Reeb edges, and red linear features. (a) A regulated tour yields continuous trajectories. (b) An unregulated tour causes discontinuities, with purple lines marking extra routes for connectivity.

distinct. For example, in Fig. 2(c), y_3 and y_4 are identified, while y_1 and y_2 remain separate.

Critical points are handled by mapping saddle arcs to loops, while extrema collapse to points. Under these identifications, the quotient space $\mathcal{M}_w$ forms a manifold whose topology mirrors the original Morse decomposition of $\mathcal{A}_w$. We similarly obtain $\mathcal{M}_c$ from $\mathcal{A}_c$, and set $\mathcal{M} = \mathcal{M}_w \sqcup \mathcal{M}_c$.

Robots switching between area exploration and linear-feature servicing are modeled by an *attaching map* $\cup_h$, which connects manifold cells along boundary arcs via a homeomorphism h. Such homeomorphisms are difficult to guarantee in $\mathbb{R}^2$. In the manifold space, we introduce a *stitching* operation Σ to ensure that a valid homeomorphism exists. Specifically, Σ defines the equivalence between $x_c \in \mathcal{M}_c$ and $x_w \in \mathcal{M}_w$ if they lie within the same open ball of radius a centered at the stitching point. Stitching eliminates non-bijective points of measure zero, so each adjacency of $\mathbb{G}_{\text{sim}}$ admits a well-defined attaching map [13]. $\mathcal{M}$ is updated by $\mathcal{M}/\Sigma$ at each stitching. Notably, pointwise stitching along the entire boundary $\partial\mathcal{M}$ makes $\mathcal{M}$ topologically equivalent to a sphere, but this approach is most inefficient because it forces redundant double coverage on $\mathcal{F}$ alone. As illustrated in Fig. 2(e), applying Σ around c_1 separates the closed loop of $\mathcal{M}(\boldsymbol{e}_1)$ into two loops with closure at c_1, while simultaneously merging $\mathcal{M}(\boldsymbol{e}_1)$ and $\mathcal{M}(\boldsymbol{e}_5)$ into a single manifold with two boundaries.

In coverage planning, certain features of the environment directly affect the difficulty of generating continuous, collision-free trajectories. Holes in the domain (e.g., obstacles or uncovered regions) force the robot to make detours, often creating overlaps in Boustrophedon trajectories [14], while boundaries only require potentially discontinuous connections between adjacent cells. To capture this effect, we define the topological complexity (TC) of a stitched manifold $\mathcal{M}^{\text{att}}$ as a measure of the intrinsic "difficulty" induced by holes and boundaries: $TC(\mathcal{M}^{\text{att}}) = N_g(\mathcal{M}^{\text{att}}) + \frac{1}{2}N_{bd}(\mathcal{M}^{\text{att}})$, where N_g and N_{bd} are the number of holes and boundaries, respectively, consistent with the Euler characteristic [15].

Lemma 1: If a sub-path $\boldsymbol{p}'$ contains a discontinuous attachment between manifold cells, then $TC(\mathcal{M}^{\boldsymbol{p}'}) > MB(\boldsymbol{p}')$, where the Morse bound is defined as $MB(\boldsymbol{p}') = \max\{\, TC(\mathcal{M}(\boldsymbol{e})) \mid \boldsymbol{e} \in \boldsymbol{p}' \,\}$, i.e., the maximum topological complexity among the manifold cells traversed by $\boldsymbol{p}'$.

For illustration, the manifold cells in Fig. 2(d) have $MB = 0.5$, as $\mathcal{M}(\boldsymbol{e}_1)$ contains one boundary. After the attachment introduced above, the resulting manifold has two boundaries with $TC = 1$. Hence, the attachment is not Morse bounded.

By Lemma 1, a continuous Eulerian tour must satisfy the *Morse boundedness*, that is, for every sub-path $\boldsymbol{p}' \in \mathcal{T}_{\text{int}}(\boldsymbol{p})$ we require $TC(\mathcal{M}^{\boldsymbol{p}'}) \leq MB(\boldsymbol{p}')$, where $\mathcal{T}_{\text{int}}(\boldsymbol{p})$ denotes the interval topology on $\boldsymbol{p}$. Under a fixed sweep direction, this topological optimality is equivalent to path optimality. We therefore restrict attention to the Morse-bounded set

$$\mathcal{P} = \{\, \boldsymbol{p} \mid \boldsymbol{p} \text{ is Eulerian and Morse bounded} \,\}, \quad (1)$$

which contains all path-length optimal tours. We then obtain optimal multi-robot allocations within $\mathcal{P}$.

C. Regulation Rules on Graph

Based on Morse boundedness, we propose HCMR that regulates Eulerian-tour behaviors by enforcing topological constraints at the graph level. As shown in Fig. 4, HCMR proceeds in two stages: (1) hierarchical decomposition of the simplified graph, (2) cyclic merging search (CMS) regulation and expansion into Morse-bounded Eulerian tours within each component. Regulation rules follow from the bijection between manifold cells and edges of $\mathbb{G}_{\text{sim}}$, ensuring that Eulerian tours remain consistent with the underlying manifold structure. In fact, the tour in Fig. 3(a) is Morse bounded, since it is constrained by $\boldsymbol{B}_1 \to \boldsymbol{C}_1 \to \boldsymbol{C}_2 \to \boldsymbol{C}_3 \to \boldsymbol{C}_4 \to \boldsymbol{B}_2$. The validity conditions for Morse boundedness are:

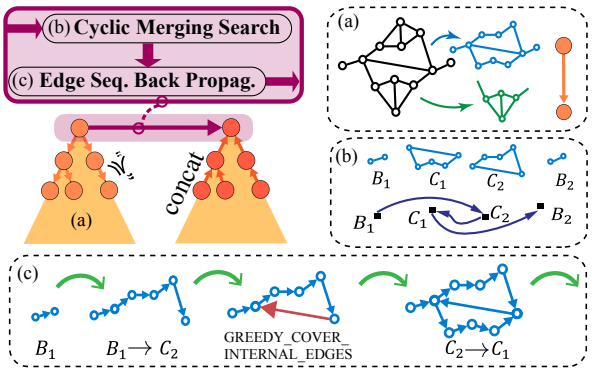

Fig. 4. Overview of the HCMR framework. (a) The simplified graph $\mathbb{G}_{\text{sim}}$ is decomposed hierarchically. (b) CMS regulates edge order within each component. (c) BP expands them into Eulerian tours. Sequences are then concatenated across levels to obtain Morse-bounded tours. The dashed blocks labeled (a)-(c) correspond to steps (a)-(c) in the framework.

Theorem 1: On $\mathbb{G}_{\text{sim}}$: (i) a cycle is valid iff it belongs to a disjoint cycle basis; (ii) two cycles merge validly iff they intersect in a simple path; (iii) a bridge attaches validly to an edge in cycle iff the edge lies in E_w, where a disjoint cycle basis $\mathcal{K}^d$ is defined as a cycle basis in which no cycle's Reeb cells are fully contained within another.

In practice, $\mathcal{K}^d$ is obtained by iterative symmetric-difference operations from a random cycle basis. To resolve larger enclosing cycles before local ones, we organize $\mathcal{K}^d$ into a hierarchy based on containment. At each hierarchy level, the induced subgraph is expanded with adjacent bridges and then decomposed into connected components. As shown in Fig. 4(a), the green component lies fully within the blue one, making it a child in the hierarchy. For each component, CMS and edge sequence back propagation (BP) are applied to regulate Eulerian tours within the subgraph.

1) CMS: On each hierarchical component, CMS (Algorithm 1) begins from the leftmost bridge and explores admissible sequences of cycles and bridges via depth-first search. During merging, it enforces Theorem 1 (ii). In Fig. 4(b), the blue component is decomposed into bridges $\boldsymbol{B}_1, \boldsymbol{B}_2$ and disjoint cycles $\boldsymbol{C}_1, \boldsymbol{C}_2$. Since $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ share only a single edge which is a simple path, once $\boldsymbol{B}_1$ connects to $\boldsymbol{C}_2$, the sequence continues to $\boldsymbol{C}_1$. With no cycles remaining, the search terminates by returning through $\boldsymbol{B}_2$.

2) BP: Each CMS sequence is expanded into a complete Eulerian tour using BP (Algorithm 2). BP exploits cycle commutativity, prioritizing internal edges before spanning the remaining ones. For bridges, it enforces Theorem 1 (iii). In Fig. 4(c), $\boldsymbol{B}_1$ cannot start with the red edge when merging with $\boldsymbol{C}_2$, since it is not in $\boldsymbol{E}_w$. Traversal of $\boldsymbol{C}_2$ ends at its first shared vertex with $\boldsymbol{C}_1$, after which $\boldsymbol{C}_1$ is visited from the shared edge.

Theorem 2: Given Eulerian tours at all hierarchy levels, their concatenation yields an Eulerian tour of $\mathbb{G}_{\text{sim}}$.

Proofs of the lemmas and theorems follow from Morse manifold analysis in [13]. By Theorem 2 and HCMR construction, the Morse-bounded set in Eq. (1) is established.

For multi-robot allocation, each Eulerian tour $\boldsymbol{p} \in \mathcal{P}$ is partitioned into balanced subpaths using [16]. Morse boundedness further enables efficient trajectory generation without exponential search [17] or heuristics [18]. The framework remains collision-free under a mass-point model consideration, except at odd node pairings in $\mathbb{G}_{\text{aug}}$ [13].

Algorithm 1: Cyclic Merging Search (CMS)

Input : Cycle set $\boldsymbol{E}_{\text{cycle}}$, bridge set $\boldsymbol{E}_{\text{bridge}}$
Output: Set of legal sequences $\mathcal{S}$
$\mathcal{S} \leftarrow \emptyset$
while *DFS not terminated* **do**
  Select next segment and merge cycles
  if *cycle merge violates Theorem 1 (ii)* **then**
    prune branch
  if *all cycles visited* **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\text{current sequence}\}$

Algorithm 2: Edge Sequence Back Propagation (BP)

Input : Segment sequence $\boldsymbol{S}$, bridge set $\boldsymbol{E}_{\text{bridge}}$, component $\mathbb{G}_{i,j}$
Output: Eulerian tour p
$p \leftarrow \emptyset$, $\boldsymbol{m} \leftarrow \emptyset$
foreach $S \in \boldsymbol{S}$ **do**
  if S *is a bridge* **then**
    BRIDGEWALK(S), $\boldsymbol{m} \leftarrow \emptyset$
  else if S *is a cycle* **then**
    if $\boldsymbol{m} \neq \emptyset$ **then**
      GREEDYCOVERINTERNALEDGES($\boldsymbol{m}, S$)
    $\boldsymbol{m} \leftarrow \boldsymbol{m} \triangle S$, CYCLETONEXTSEGMENT($S$)

IV. Line Coverage Motion Planning and Control

Fig. 5(a) shows the crack-filling robot. It is a three-wheel omnidirectional platform with holonomic planar motion. Let $\boldsymbol{q}_{r_i} = [x_{r_i}\, y_{r_i}\, \theta_i]^\top$ denote its pose. Wheel angular velocities $\boldsymbol{\omega}_i \in \mathbb{R}^3$ relate to body velocity $\boldsymbol{u}_i = \dot{\boldsymbol{q}}_{r_i}$ via $\boldsymbol{\omega}_i = \boldsymbol{A}_r \boldsymbol{u}_i$, where $\boldsymbol{A}_r$ is the full-rank wheel Jacobian matrix. As shown in Fig. 5(c), the nozzle N_i is rigidly mounted on the robot's side, offset from the center $\boldsymbol{O}_{B_i}$ by distance D and angle φ. The position of N_i in the body frame is $\boldsymbol{q}_{n_i} = [x_{r_i}\, y_{r_i}]^\top + D\boldsymbol{R}_z(\theta_i)[\cos\varphi\ \sin\varphi]^\top$, where $\boldsymbol{R}_z(\theta_i) \in SE(2)$ is the plane rotational matrix about the z-axis with angle θ_i.

During crack filling, the robot center follows sparse paths $\boldsymbol{\pi}(\boldsymbol{b})$, while the nozzle fills the crack (Fig. 5(d)). For each crack branch, the dilated boundary needs to form a smooth simple loop to generate the skeleton. Cusps create non-smooth boundaries that inflate the traced path, while self-intersections break the monotone index and may skip crack segments (Fig. 5(b), n_2, n_3). From offset-curve theory, such singularities arise when the offset radius exceeds the minimum curvature radius, i.e., $\kappa_{\max} a \geq 1$ [19], where $\kappa_{\max}$ is the maximum crack curvature and a the filling radius. We enforce $\kappa_{\max} a \leq 0.6$ to ensure smooth dilation, yielding a one-to-one correspondence between edges in $\boldsymbol{\pi}(\boldsymbol{b})$ and continuous crack intervals. For an edge $\boldsymbol{e} \in \boldsymbol{\pi}(\boldsymbol{b})$, the corresponding crack interval is $\boldsymbol{b}(\boldsymbol{e})$.

Algorithm 3 outlines the nozzle planning and control pipeline. Connected crack segments assigned to each robot are first grouped, and then the robot follows a simplified skeleton path while the nozzle traces the crack under a

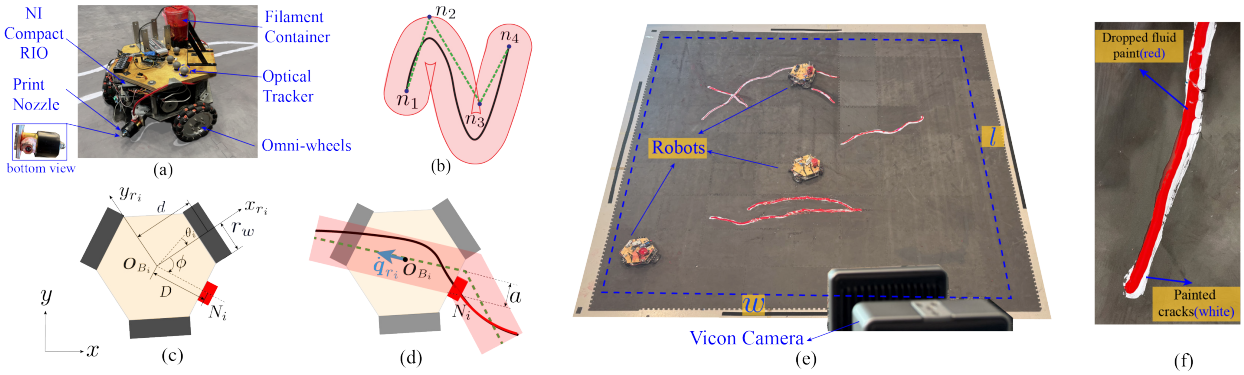

Fig. 5. (a) Side view of the crack-filling robot. (b) Counterexample where an oversized a causes cusps and self-intersections. (c) Schematic of robot kinematics. (d) Illustration of crack filling. The robot center follows simplified paths while the nozzle follows cracks; red lines indicate filled segments and the pink area denotes the a-dilation of the robot trajectory. (e) Indoor experimental setup with optical motion capture. (f) Example of a filling result.

Algorithm 3: Crack Filling

Input : Path $\boldsymbol{\pi}(\boldsymbol{b})$, crack set $\mathcal{C}$, band ε
Output: Twist command $\boldsymbol{q}^{\star}_{r_i}$
$\mathcal{E} \leftarrow$ ClusterEdges$(\boldsymbol{\pi}(\boldsymbol{b}), \mathcal{C})$, aligned$\leftarrow$**false**, $e \leftarrow 1$
while $e \le |\mathcal{E}|$ **do**
  if ***not*** *aligned* **then**
    $(\boldsymbol{q}^{\star}_{r_i}, \text{aligned}) \leftarrow$ AlignMPC$(\boldsymbol{e}_{e,0}, \theta^{\star}_e)$
  else
    $(\boldsymbol{q}^{\star}_{r_i}, \text{aligned}) \leftarrow$ FollowMPC$(\boldsymbol{A}_e, c_e, \varepsilon)$
  $s \leftarrow$ Progress$(\boldsymbol{q}_{n_i}, \boldsymbol{e}_e)$
  if $s \ge 1-\delta$ **then** $e \leftarrow e+1$

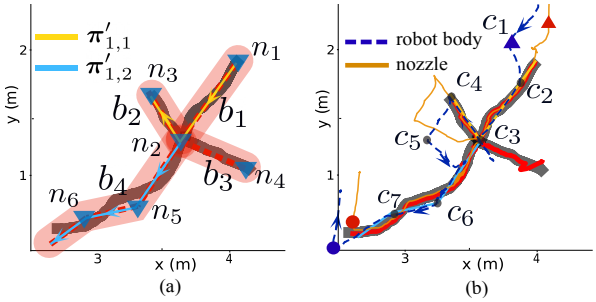

Fig. 6. Schematic of nozzle planning and control. Thick black curves denote cracks. (a) Red dotted lines indicate simplified paths with ▼ as nodes; light-red regions show their $a\oplus$-dilation. Yellow and blue lines mark grouped edges $\boldsymbol{\pi}'_1$ and $\boldsymbol{\pi}'_2$. (b) Red curves show paint-drop trajectories; the blue dotted line denotes the robot body path and the yellow line the nozzle path. Mode transitions are marked by ▲, ● (body) and ▲, ● (nozzle).

model predictive control (MPC). Given an assigned graph path $\boldsymbol{p}_i \subset \boldsymbol{p}$, we apply ClusterEdges to group consecutive filling edges. Formally, a subpath $\boldsymbol{p}'_i \in \mathcal{T}_{\text{int}}(\boldsymbol{p}_i)$ is defined as a component if it consists entirely of filling edges $\boldsymbol{E}_c$, bounded by non-filling edges on both sides, with edges connected head-to-tail. Each component then generates a skeleton path for the robot center $\boldsymbol{\pi}'_{i,m}$, while the nozzle tracks $\boldsymbol{b}'_{i,m} = \bigcup_{\boldsymbol{e}\in\boldsymbol{\pi}'_{i,m}} \boldsymbol{b}(\boldsymbol{e})$. As illustrated in Fig. 6(a), the crack set $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{b}_4\}$ is simplified to $\{(n_1, n_2), (n_2, n_3), (n_2, n_4), (n_2, n_5, n_6, \dots)\}$. For R_1, (n_1, n_2) and (n_2, n_3) form $\boldsymbol{\pi}'_{1,1}$ with $\boldsymbol{b}'_{1,1} = \boldsymbol{b}_1 \cup \boldsymbol{b}_2$, while $\boldsymbol{\pi}'_{1,2} = (n_2, n_5, n_6)$ with $\boldsymbol{b}'_{1,2} = \boldsymbol{b}_4((n_2, n_5)) \cup \boldsymbol{b}_4((n_5, n_6))$. A two-step MPC then drives the robot along $\boldsymbol{\pi}'_{i,m}$ while the nozzle tracks $\boldsymbol{b}'_{i,m}$.

Before filling, we need to bring the robot body near the start of the assigned crack and orient for entry. The alignment MPC is formulated as

$$\min_{\boldsymbol{u}_H} \sum_{k=0}^{H} \left(\|\boldsymbol{q}^k_{n_i} - \boldsymbol{\pi}'^0_i\|^2_{\boldsymbol{Q}_n} + \|\theta^k_{r_i} - \theta^0_{r_i}\|^2_{\boldsymbol{Q}_r} + \|\boldsymbol{u}^k_i\|^2_{\boldsymbol{R}_r} \right)$$
$$\text{s.t. } \boldsymbol{q}^{k+1}_{r_i} = \boldsymbol{q}^k_{r_i} + \Delta t \boldsymbol{u}^k_i,\ \boldsymbol{\omega}^k_i = \boldsymbol{A}_{r_i} \boldsymbol{u}^k_i,$$
$$\|\boldsymbol{\omega}^k_i\| \le \omega^{\max}_r, k = 0, \dots, H,$$

where $\boldsymbol{u}_H = \{\boldsymbol{u}^0_i, \boldsymbol{u}^1_i, \dots, \boldsymbol{u}^H_i\}$, $\boldsymbol{\pi}'^0_i$ is the start of the assigned crack, H the prediction horizon, Δt the step duration, and superscript k the prediction step index.

During filling, the robot tracks $\pi'(\boldsymbol{b})$ on the current edge $\boldsymbol{e}_e$ while constraining the robot body remain within a narrow band around that edge. Each edge is parameterized in normalized implicit form $\boldsymbol{A}_e^T \boldsymbol{r}^{\star}_i + c_e = 0$, where $\boldsymbol{A}_e \in \mathbb{R}^2$ is the unit outward normal vector of the edge ($\|\boldsymbol{A}_e\| = 1$), c_e is the signed offset, and $\boldsymbol{r}^{\star}_i$ denotes the target robot position. The crack-following MPC is formulated as

$$\min_{\boldsymbol{u}_H} \sum_{k=0}^{H} (\|\boldsymbol{q}^k_{n_i} - \boldsymbol{\pi}'_{i,t}(\boldsymbol{b})\|^2_{\boldsymbol{Q}_n} + \|\boldsymbol{u}^k_i\|^2_{\boldsymbol{R}_r})$$
$$\text{s.t. } \boldsymbol{q}^{k+1}_{r_i} = \boldsymbol{q}^k_{r_i} + \Delta t \boldsymbol{u}^k_i,\ \boldsymbol{\omega}^k_i = \boldsymbol{A}_{r_i} \boldsymbol{u}^k_i,$$
$$\|\boldsymbol{\omega}^k_i\| \le \omega^{\max}_r,\ |\boldsymbol{A}_e \boldsymbol{r}^k_i + c_e| \le \varepsilon \ll a,$$
$$\underbrace{\boldsymbol{t}_e^{\top} \boldsymbol{J}_{n_i}(\boldsymbol{q}^k_{r_i}) \boldsymbol{u}^k_i}_{\text{nozzle velocity along edge}} \ge 0, \quad k = 0, \dots, H,$$

where the unit tangent vector is $\boldsymbol{t}_e = \frac{\boldsymbol{e}_e}{\|\boldsymbol{e}_e\|}$ with $\boldsymbol{e}_e = (\boldsymbol{e}_{e,0}, \boldsymbol{e}_{e,1})$. $\boldsymbol{J}_{n_i}(\boldsymbol{q}^k_{r_i}) = [\boldsymbol{I}_2\ D[-\sin(\theta^k_i + \varphi)\ \cos(\theta^k_i + \varphi)]^{\top}]$ denotes the predicted Jacobian of the nozzle tip with respect to the robot yaw motion, and $\boldsymbol{I}_n$ is the $n \times n$ identity matrix.

The tracking edge parameterized by $(\boldsymbol{A}_e, c_e)$ is considered completed when normalized arc length $s = \frac{(\boldsymbol{q}_{n_i} - \boldsymbol{e}_{e,0})^{\top} \boldsymbol{t}_e}{\|\boldsymbol{e}_e\|} \ge 1 - \delta$ for small $\delta > 0$. One real-experiment crack-filling process is illustrated as Fig. 6(b). The robot starts aligning at c_1 and finishes aligning at c_2, where the nozzle locates at the start of $\boldsymbol{b}_1$. Robot body then traverses $\boldsymbol{\pi}'_{1,1}$ while the nozzle fills $\boldsymbol{b}'_{1,1}$. $c_4 \to c_5$ is exploration. From c_5 the robot aligns again and then fills $\boldsymbol{b}'_{1,2}$. The MPC is implemented in CasADi, with a collision-avoidance module to prevent inter-robot conflicts.

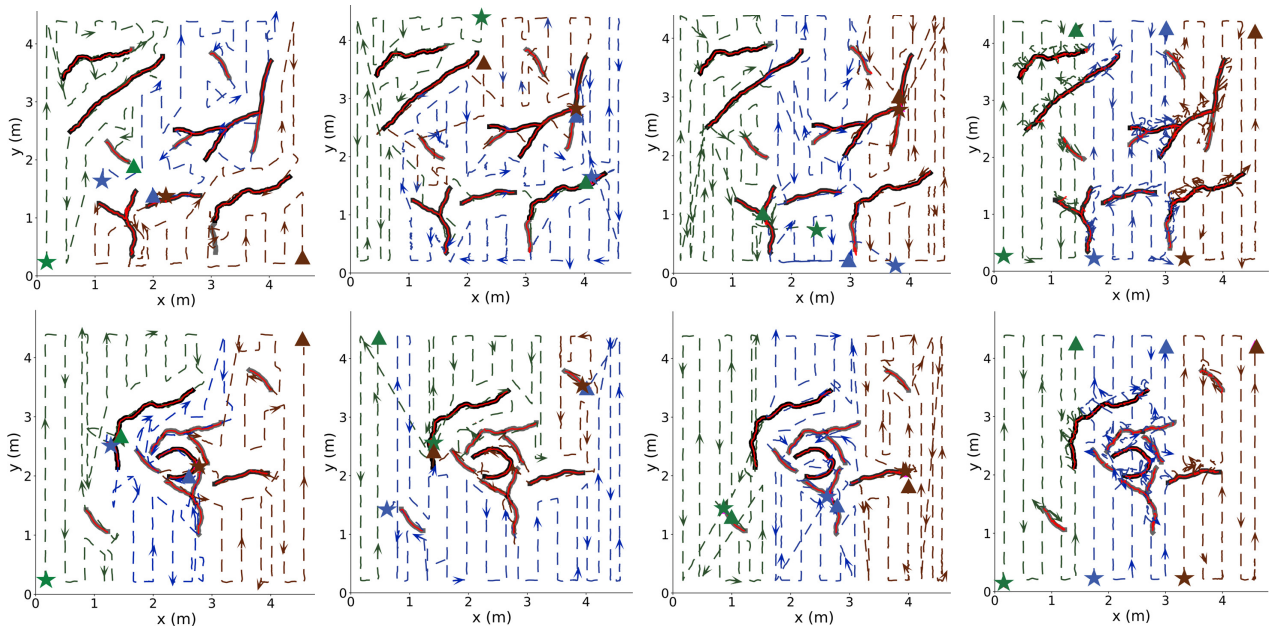

Fig. 8. Comparison of three-robot double coverage on U100 and G100 maps. Each column corresponds to (a) HCMR, (b) G-mSCC, (c) S-mSCC, and (d) mGreedy algorithms. The top row presents results on the U100 map, while the bottom row shows the G100 map. Black thick lines denote the underlying crack maps, and red solid lines indicate the drop-paint trajectories executed by the robots. Robots start at ★ and end at ▲.

V. Experiments and Results

Fig. 5(c) shows the indoor crack-filling testbed. Crack maps were painted on foam mats to emulate road surfaces Robot pose was tracked at 100 Hz using optical markers and an 8-camera motion capture system. Paint was delivered via a hydraulic pump and solenoid valve. The onboard controller ran on a real-time embedded system for low-level motion control. High-level planning and MPC were executed on a centralized computer at 100 Hz. The size of the indoor testing site was 4.7×4.5 m. Three robots were deployed, each with physical parameters $r_w = 7.6$ cm, $d = 21$ cm, $r = 23.6$ cm, $v_e = 0.1$ m/s, $v_s = 0.03$ m/s, $D = 18.6$ cm and $\varphi = -60°$, with maximum curvature of $\kappa_{\max} = 8$ m^{-1} and filling radius $a = 6.9$ cm. MPC parameters were $H = 20$ and $\Delta t = 0.01$s.

To evaluate scalability, crack maps were generated using the dataset in [20], with branch locations and orientations sampled from uniform and Gaussian distributions. Density was defined as the ratio of $\boldsymbol{b} \oplus r$ to workspace area, ranging from 35% to 100%. Each category ("distribution+density") included five simulated maps, with four representative cases (U80, U100, G80, G100) tested experimentally. We benchmarked HCMR against three planners. SCC, originally for the single-robot double coverage problem [9], was extended to multi-robot scenarios using two partitioning strategies: (1) G-mSCC, which balances Eulerian graph partitions [5]; and (2) S-mSCC, which partitions the workspace into equal regions [4] with SCC applied in each. In addition, we implemented the multi-robot Greedy algorithm from [9], denoted mGreedy. Performance was evaluated using five metrics: (1) *Total path length*: sum of arc lengths traveled by all robots; (2) *Task time*: completion time of the slowest robot; (3) *Sensor coverage*: workspace percentage covered by sensors; (4) *Filling coverage*: percentage of crack length filled within 1 cm tolerance; and (5) *Avoidance time*: the cumulative duration of avoidance maneuvers in real experiments, or *conflict counts*: the number of inter-robot path intersections.

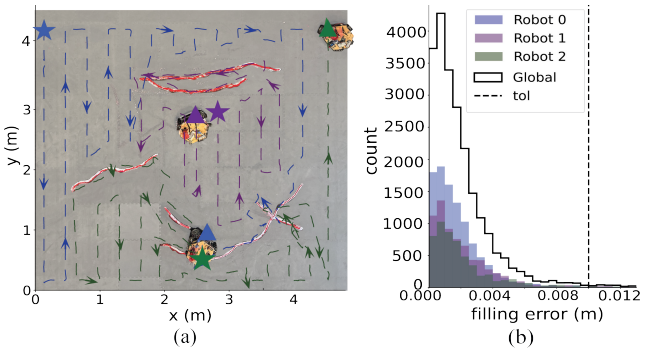

Fig. 7. (a) Top view of three robots performing double coverage on U80 map. White lines represent linear features, red lines indicate serviced regions, dashed curves show robot center trajectories, and arrows mark travel direction. Robots start at ★ and end at ▲. (b) Histogram of filling error.

As for experimental results, we first present experimental results of the HCMR planner with nozzle motion control. Fig. 7(a) shows trajectories of three robots filling a crack map with 80% density under a uniform distribution (U80). The robots complete their tasks efficiently with minimal trajectory intersections. During filling, each robot body follows edges while the nozzle traces the cracks, shortening the body path. The nozzle maintains a tracking error below 1 cm (Fig. 7(b)), achieving 98.7% coverage of $\mathcal{C}$. Fig. 8 compares planners on U100 and G100 maps. All methods achieved full coverage; notably, HCMR consistently produced shorter and more balanced trajectories. Table I shows that HCMR achieved the lowest path length and task time across cases.

TABLE I

EXPERIMENT PERFORMANCE COMPARISON FOR THREE-ROBOT DOUBLE COVERAGE ON FOUR CRACK MAPS

	Total Path Length (m)				**Robot Travel Time** (s)				**Sensor Coverage** (%)				**Filling Coverage** (%)				**Avoidance Time** (s)			
Crack dist.	U80	U100	G80	G100	U80	U100	G80	G100	U80	U100	G80	G100	U80	U100	G80	G100	U80	U100	G80	G100
HCMR	**79**	**81**	**80**	**79**	**370**	**426**	**308**	**371**	99	99	99	**100**	**99**	**99**	**98**	**97**	58	33	36	48
G-mSCC	95	99	88	96	512	520	390	429	**100**	99	**100**	99	98	98	**98**	94	77	23	10	12
S-mSCC	97	112	101	108	508	579	489	594	**100**	**100**	**100**	**100**	94	97	**98**	96	51	33	70	12
mGreedy	103	125	84	105	564	671	329	682	99	99	99	98	90	91	92	90	**0**	**0**	**0**	**0**

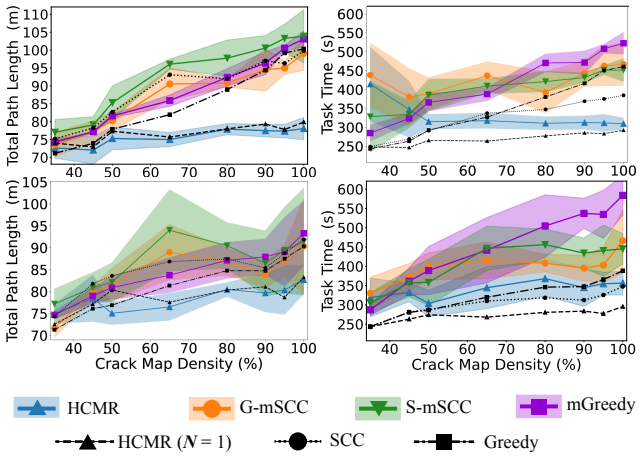

Fig. 9. Comparison of planning algorithms on cracks with varying densities under uniform and Gaussian distributions. Top and bottom rows correspond to uniformly and Gaussian-distributed cracks, respectively. Left: total path length; right: task completion time. Lines indicate mean values and shaded areas one standard deviation. For task completion time, the single-robot curve is normalized by the number of robots N.

The variance of HCMR paths remained small because of Morse boundedness, whereas other planners exhibited increasing path growth with crack density. Frequent switching caused mGreedy to incur the longest paths and travel times.

We further conducted simulations on the same indoor setup. The comparison results over varying densities (Fig. 9) show that total path length scales approximately linearly with density. The three-robot configuration consistently outperformed the single-robot case. For uniform cracks, HCMR achieved the shortest paths. For Gaussian cracks, HCMR remained competitive and maintained shorter task times even when path length differences were small. Task time analysis further demonstrates scalability that HCMR converged toward the ideal $1/N$ scaling relative to single-robot performance. Overall, HCMR improves total path length by at least 10.0%, reduces task time by 16.9% on average.

VI. CONCLUSION

This paper introduced HCMR, an algorithmic framework that constitutes the optimal solution to the multi-robot double coverage problem. By projecting regions induced by the task onto manifolds, HCMR guaranteed complete coverage while generating optimal routes in known environments. Extensive experiments and simulations demonstrated its superior performance across diverse target distributions, significantly reducing both path length and task time compared to state-of-the-art planners. Future research will aim to extend HCMR to the online multi-robot double coverage problem in environments with unknown targets.

REFERENCES

[1] T. M. Cabreira, L. B. Brisolara, and F. J. Paulo R, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, 2019.

[2] M. Wei and V. Isler, "Coverage path planning under the energy constraint," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 368–373.

[3] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *Int. J. Robot. Res.*, vol. 21, no. 4, 2002.

[4] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Ann. Math. Artif. Intell.*, vol. 52, pp. 109–142, 2008.

[5] N. Karapetyan, K. Benson, C. McKinney, P. Taslakian, and I. Rekleitis, "Efficient multi-robot coverage of a known environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2017, pp. 1846–1852.

[6] S. Agarwal and S. Akella, "Area coverage with multiple capacity-constrained robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3734–3741, 2022.

[7] H. Oh, S. Kim, A. Tsourdos, and B. A. White, "Coordinated road-network search route planning by a team of UAVs," *Int. J. Syst. Sci.*, vol. 45, no. 5, pp. 825–840, 2014.

[8] S. Agarwal and S. Akella, "Line coverage with multiple robots: Algorithms and experiments," *IEEE Trans. Robotics*, vol. 40, pp. 1664–1683, 2024.

[9] V. Veeraraghavan, K. Hunte, J. Yi, and K. Yu, "Complete and near-optimal robotic crack coverage and filling in civil infrastructure," *IEEE Trans. Robotics*, vol. 40, pp. 2850–2867, 2024.

[10] K. Yu, C. Guo, and J. Yi, "Complete and near-optimal path planning for simultaneous sensor-based inspection and footprint coverage in robotic crack filling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 8812–8818.

[11] N. Gucunski, B. Basily, J. Kim, J. Yi, T. Duong, K. Dinh, S.-H. Kee, and A. Maher, "RABIT: implementation, performance validation and integration with other robotic platforms for improved management of bridge decks," *Int. J. Intell. Robot. Appl.*, vol. 1, no. 3, pp. 271–286, 2017.

[12] J. Liu *et al.*, "Intermittent deployment for large-scale multi-robot forage perception: Data synthesis, prediction, and planning," *IEEE Trans. Automat. Sci. Eng.*, vol. 21, no. 1, pp. 27–47, 2024.

[13] T. Zheng, J. Yi, and K. Yu, "Optimal planning for multi-robot simultaneous area and line coverage using hierarchical cyclic merging regulation," *arXiv preprint arXiv:2508.04981*, 2025.

[14] 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*, G. Ishigami and K. Yoshida, Eds. Springer, 2021, pp. 277–290.

[15] A. Hatcher, *Algebraic Topology*. Cambridge, UK: Cambridge University Press, 2002.

[16] L. Xu, "Graph planning for environmental coverage," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, August 2011.

[17] J. Xie, L. R. G. Carrillo, and L. Jin, "An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems," *IEEE Control Syst. Lett.*, vol. 3, no. 1, pp. 67–72, 2018.

[18] W.-C. Tung and J.-S. Liu, "Solution of an integrated traveling salesman and coverage path planning problem by using a genetic algorithm with modified operators," *IADIS Int. J. Comput. Sci. Inf. Syst.*, vol. 14, no. 2, pp. 95–114, 2019.

[19] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[20] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 12, pp. 3434–3445, 2016.