

Efficient Multi-Robot Coverage Path Planning of Non-Convex Regions of Interests

Abstract—This letter presents a framework for multi-robot coverage path planning in large, nonconvex Regions of Interest containing obstacles and no-fly zones. The proposed method begins by determining an optimal sweep orientation through the minimum-area enclosing rectangle computed using a rotating-calipers procedure. Using this orientation, a minimum-turn swath generation algorithm produces parallel coverage lines that are clipped against the obstacle-free space and sorted along the field’s minor axis to ensure spatial continuity and reduce turn transitions. Both the outer ROI boundary and all internal exclusion zones are expanded using analytically derived headland buffers, producing a reduced operable region that guarantees sufficient clearance for turning maneuvers and prevents boundary violations. A visibility-roadmap (VRM) structure provides geometric connectivity between disjoint swath segments, enabling feasible transitions that remain entirely inside the buffered region. A modified multiple Traveling Salesman Problem (mTSP) formulation assigns the ordered swaths to robots, balancing workloads and minimizing total mission time. Simulations and real-world experiments demonstrate that the proposed framework significantly reduces the number of turns, energy consumption, and computation time compared to state-of-the-art planners, while maintaining strong scalability across increasing fleet sizes. The MRCPP framework will be released as an open-source package and videos of real-world and simulated experiments are available at <https://mrc-pp.github.io/>.

Index Terms—Multi-Robot Coverage Path Planning, Non-Convex Environments, Obstacle-Aware Planning

I. INTRODUCTION

Coverage Path Planning (CPP) enables autonomous robots to systematically traverse a Region of Interest (ROI), supporting applications in precision agriculture [1], environmental monitoring [2], infrastructure inspection [3], and search-and-rescue [4]. Efficiency is quantified by path length, number of turns, mission duration, and energy expenditure, metrics critical to large-scale multi-robot deployments in complex environments.

Classical CPP methods, including boustrophedon decomposition [5] and grid-based sweeping [6], perform well in convex, obstacle-free domains but struggle in non-convex ROIs with irregular boundaries and internal exclusions. In such settings, naive decomposition produces fragmented swaths, excessive turns, and redundant overlaps, increasing energy consumption and mission time. Multi-Robot CPP (MRCPP) frameworks parallelize coverage but remain challenged by three interdependent requirements: optimal geometric decomposition, obstacle-aware trajectory generation, and balanced task allocation.

Existing MRCPP methods often address turn minimization and path continuity inadequately. Partition-based approaches such as DARP+MST [7, 8] decompose the ROI spatially without considering sweep orientation, yielding subregions with inconsistent headings and high turn counts. Large-scale solvers like POPCORN+SALT [9] use cell-based GTSP formulations that do not enforce globally aligned swaths, resulting in



(a) AAV coverage

(b) ASV coverage

Fig. 1: Multi-robot coverage path planning in real-world Autonomous Aerial Vehicle (AAV) and Autonomous Surface Vehicles (ASV) experiments using the proposed method. In Fig. 1a, the white polygon denotes the Region of Interest (ROI) to be covered. Two DJI Mavic Air AAVs execute the coverage task, with individual coverage paths shown in orange and blue, while the red shaded circular region indicates a no-fly zone. In Fig. 1b, two Autonomous Surface Vehicles (ASVs) execute coverage paths shown in red and green over satellite imagery of an aquatic environment.

disjointed coverage with frequent heading reversals. Energy-aware planners such as EAMCMP [10] incorporate flight dynamics, but lack explicit turn-minimizing swath geometry, limiting performance in cluttered environments. A unified framework that directly optimizes turn count, ensures obstacle-aware connectivity, and balances workload is needed.

This letter presents an MRCPP framework integrating analytic geometry, graph-based connectivity, and optimization-based allocation for efficient coverage of non-convex environments. First, rotating calipers identify the minimum-area enclosing rectangle to determine the optimal sweep direction, minimizing turns by aligning swaths with the ROI’s principal axis. Parallel swaths are then generated and decomposed to respect headland buffers around obstacles. A Visibility–Roadmap (VRM) graph connects disjoint swath segments via collision-free detours, ensuring continuous traversal. Finally, a modified multiple Traveling Salesman Problem (mTSP) allocates swaths to minimize mission time and eliminate redundant transitions.

The contributions of this letter are as follows:

- 1) An **orientation-optimized swath generation algorithm** that minimizes turns by aligning coverage with the ROI’s principal axis while maintaining obstacle and headland clearance.
- 2) An **obstacle-aware connectivity framework** using VRM graphs to enable continuous, collision-free transitions between disjoint swath segments.
- 3) A **workload-balanced mTSP formulation** for multi-robot allocation that minimizes mission duration and travel cost.

The remainder of this letter is organized as follows. Section II reviews related work in single and multi-robot coverage path planning. Section III formulates the multi-robot coverage problem. Section IV presents the minimum-turn swath generation method for non-convex regions. Section V describes the multi-robot allocation and connectivity framework. Section VI provides the algorithmic analysis. Section VII reports simu-

lation and real-world experimental results, and Section VIII concludes the letter.

II. RELATED WORK

CPP has become a cornerstone of autonomous robotics, supporting applications from precision agriculture to infrastructure inspection [11]. Early research focused on exact cellular decomposition, with boustrophedon decomposition [5] emerging as a key advancement over trapezoidal methods [12]. This approach efficiently handles regions with obstacles and no-fly zones while reducing the number of generated sub-cells [6]. Later work introduced hybrid methods combining decomposition with spiral patterns [13] and approaches that construct graph representations over decomposed regions to cluster coverage tasks [14] to tackle irregular terrains. Still, these decomposition techniques may lead to fragmented paths and increased turning effort in highly non-convex or cluttered regions.

To address these limitations, researchers have reformulated CPP as a routing optimization problem similar to the Generalized Traveling Salesman Problem [15]. In [6], the authors showed how the combination of decomposition with GTSP-based sequencing can minimize the total flight time. Yet optimizing only for distance overlooks a critical factor: Autonomous Aerial Vehicle (AAV) energy consumption is heavily influenced by velocity changes, acceleration, and turns [16]. Energy-aware planners [13, 10] account for these dynamics by incorporating detailed power models and solving clustered TSP variants [17]. Many classical and decomposition-based methods still use fixed swath orientations rather than optimizing sweep direction, a decision that significantly affects both turn count and energy use.

Minimizing turns has gained attention as a direct path to reducing energy and mission time, since every heading change forces the vehicle to slow down, rotate, and accelerate again [10]. While several approaches explicitly target turn reduction [18, 19], they typically work within predetermined cell boundaries or fixed grid layouts. For instance, In [19] added turn awareness to DARP, but the underlying partition-first structure limits the ability to globally coordinate sweep alignment across sub-regions.

Multi-robot coverage introduces new complications around balancing workload and coordinating between agents. Early solutions [8, 20, 21, 22] resulted in uneven task distribution across irregular terrain. Voronoi-based methods [23, 24, 25] provide elegant mathematical guarantees but were designed for static sensor placement rather than path planning, ignoring factors like starting positions and travel distances. DARP [7] explicitly considers these elements along with obstacles and workload balance. However, both DARP and its recent improvements [19] partition the space first and plan sweeps second, which can lead to disconnected regions with conflicting sweep directions.

SMT-based decomposition to coordinate multiple AAVs is used in large-scale deployments [14, 9]. However, their main limitation is enforcing independence between sub-areas, which prevents globally aligned swaths and can create fragmented

coverage patterns. Spanning-tree approaches [26, 27] maintain coverage guarantees with provable bounds on solution quality, but they don't explicitly optimize for turns or continuous swath structures.

More recently, researchers have explored how communication constraints, energy budgets, and coordination interact in multi-robot systems. Several integrated frameworks [10, 7, 19] combine DARP-based partitioning with spanning-tree coverage while accounting for factors such as inter-robot connectivity, feasible traversal distance, and balanced workload distribution during multi-robot missions.

III. PROBLEM FORMULATION

Consider a team of N_R mobile robots tasked with covering a bounded, possibly nonconvex ROI $P \subset \mathbb{R}^2$. The ROI may contain N_{obs} exclusion zones such as obstacles, no-fly regions, or restricted areas, modeled as polygonal sets $o_i \subset \mathbb{R}^2$ for $i = 1, \dots, N_{\text{obs}}$. For simplicity, all exclusion zones are either given as polygons or approximated as such to support polygon-based decomposition. The free space F available for coverage is then computed by subtracting obstacle regions from P such that $F = P \setminus \bigcup_{i=1}^{N_{\text{obs}}} o_i$.

Each robot has a coverage width of w that defines the distance between adjacent parallel paths. A swath refers to the path segment a robot follows when traversing the field. In this configuration, swaths remain constant. A point in F is covered if it falls within the area swept by a robot moving along a swath. Formally, let S be the set of all swath segments such that $S = \{S_1, S_2, \dots, S_{N_S}\}$. Each swath segment $S_m \in S$ is defined by its endpoints via $S_m = [a_m, b_m]$, with geometric length $\ell_m = \|b_m - a_m\|$.

To achieve energy efficiency, swaths should be generated in a way that minimizes the number of turns and the total path length [10]. Given a set of swaths S produced by a minimum-turn algorithm, the objective of the multirobot coverage planner is to allocate a subset of swaths $S_R \subset S$ to the J^{th} robot such that every swath segment in S is visited by exactly one robot following a continuous, collision-free sequence. The robots start from given depot locations (or may start from any point on the boundary of F) and must traverse their assigned swaths in some order. Between the end of one swath and the start of the next assigned swath, each robot follows a shortest collision-free transition path that remains inside a slightly enlarged feasible space F' (e.g., F offset inward by the robot radius). Let d_{trans} denote the length of such a transition path. Therefore, the total distance traveled by robot r can be computed by

$$L_r = \sum_{S_m \in \mathcal{T}_r} \ell_m + \sum_{\text{transitions of } r} d_{\text{trans}}, \quad (1)$$

Assuming all robots operate under a uniform speed model, the overall mission completion time is dictated by the robot with the maximum total executed path length.

Problem. Find a partition of the swath set S into N_R subsets $\mathcal{T}_1, \dots, \mathcal{T}_{N_R}$ and, for each robot r , a collision-free ordering

of its assigned swaths together with corresponding transition paths, such that

$$\min_{\Pi} \sum_{r=1}^{N_R} L_r, \quad (2)$$

where Π denotes the set of all valid assignments over coverage paths.

IV. SWATH GENERATION FOR NONCONVEX FIELDS

When a team of cooperative robots needs to cover a nonconvex ROI, the swath-generation process must ensure that the coverage lines (swaths) are well-aligned with the field's geometry while minimizing overlaps, gaps, and unnecessary maneuvers. Here, we introduce a minimum-turn swath-generation algorithm that identifies a minimum-area rotating rectangle for a non-convex ROI and then generates swaths inside the ROI parallel to the long axis of the rotating rectangle. The following subsections describe the procedure for determining the optimal swath orientation and generating parallel candidate lines that guarantee complete coverage, even when the field is nonconvex.

A. Minimum Rotating Rectangle over a Non-Convex ROI

Given a polygonal region of interest $P \subset \mathbb{R}^2$, the objective is to determine the minimum-area enclosing rectangle that completely contains P under any possible rotation. Since the minimum bounding rectangle must be tangent to the outermost boundary of the shape, its computation depends solely on the convex hull $H = \text{ConvHull}(P)$. We employ the rotating calipers method [28] to identify the orientation of the rectangle that yields minimal area. This algorithm evaluates each edge of the convex hull as a potential alignment for one side of the rectangle, computing the corresponding bounding rectangle dimensions in that orientation, and runs in linear time with respect to the number of hull vertices n_H . The resulting minimum-area enclosing rectangle R_{\min} is characterized by its principal orientation vectors ($\hat{\mathbf{u}}, \hat{\mathbf{v}}$) and its dimensions (w_{\min}, h_{\min}) , which bound both the convex hull and the original region P .

B. Minimum-turn Swath Generation

Once a minimum rotating rectangle is found, we then partition the ROI into parallel coverage lines aligned with the major axis u . Next, we project all vertices of P onto the perpendicular axis v to maximize straight-line traversal while reducing the turning effort as:

$$\eta_{\min} = \min_{p \in V_P} (p \cdot v), \quad (3)$$

$$\eta_{\max} = \max_{p \in V_P} (p \cdot v), \quad (4)$$

where V_P denotes the set of all vertices of the polygonal region P and the total projection span is $\Delta\eta = \eta_{\max} - \eta_{\min}$.

Given a swath width w , the number of required swaths is

$$n_s = \left\lceil \frac{\Delta\eta}{w} \right\rceil. \quad (5)$$

The k -th swath center offset is

$$c_k = \eta_{\min} + \frac{w}{2} + (k-1)w, \quad k = 1, \dots, n_s, \quad (6)$$

where k indexes the swaths and n_s is the total number of swaths computed from the projection span $\Delta\eta$ and the corresponding infinite swath line is given by $x(t) = c_k v + tu$ for $t \in \mathbb{R}$.

To obtain valid in-field segments, each swath line is clipped against the polygonal boundary of P . Each edge is $E_\mu = [a_\mu, b_\mu]$ for $\mu = 1, \dots, n_e$, with parameterization $e_\mu(s) = a_\mu + s(b_\mu - a_\mu)$ for $s \in [0, 1]$.

Intersecting the swath line with each edge yields

$$t_\mu = \frac{(a_\mu - c_k v) \times (b_\mu - a_\mu)}{u \times (b_\mu - a_\mu)}, \quad (7)$$

$$s_\mu = \frac{(a_\mu - c_k v) \times u}{u \times (b_\mu - a_\mu)}. \quad (8)$$

All intersections satisfying $s_\mu \in [0, 1]$ are retained and sorted into the ordered points q_1, \dots, q_{m_k} . The candidate swath segments are

$$S_{k,\nu} = [q_\nu, q_{\nu+1}], \quad \nu = 1, \dots, m_k - 1. \quad (9)$$

A segment is accepted if its midpoint lies inside the ROI according to the even–odd rule. The complete set of minimum-turn swaths is written compactly as $S = \bigcup_{k=1}^{n_s} \{S_{k,\nu} \subset P\}$.

V. MULTIROBOT COVERAGE PATH PLANNING

Given the set of minimum-turn swaths S , the objective is to allocate swaths among robots and construct continuous, collision-free coverage paths entirely contained within the buffered free space.

A. Headland Buffer Generation

We dilate the ROI boundary and all exclusion zones to ensure safe turning clearance and obstacle avoidance. The dilation is performed by offsetting a headland width h using Minkowski operation. Let $B_h = \{x \in \mathbb{R}^2 : \|x\| \leq h\}$ denote the closed disk of radius h . The inward offset of the ROI is computed as the Minkowski difference

$$P' = P \ominus B_h, \quad (10)$$

while each exclusion zone is expanded outward using the Minkowski sum

$$o'_i = o_i \oplus B_h, \quad i = 1, \dots, N_{\text{obs}}. \quad (11)$$

Therefore, we can compute the feasible coverage space as:

$$F' = P' \setminus \bigcup_{i=1}^{N_{\text{obs}}} o'_i. \quad (12)$$

These offset boundaries also provide nodes for constructing the visibility–roadmap used in the subsequent transition-planning stage.

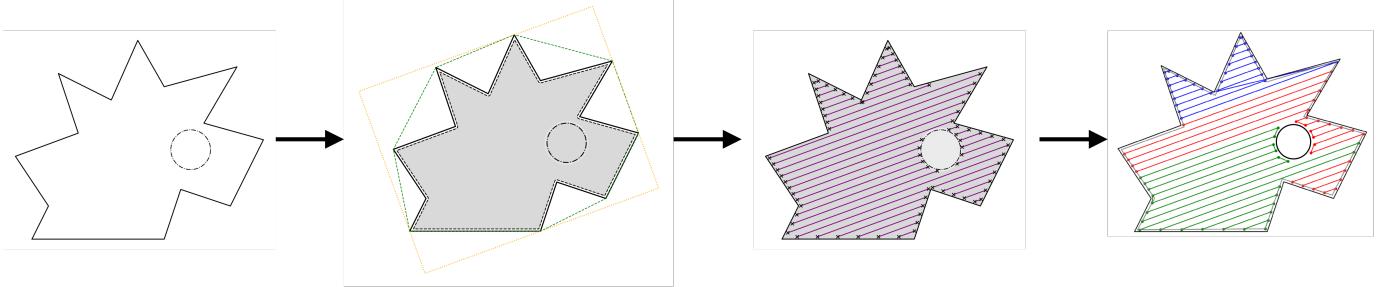


Fig. 2: Coverage path planning for multi-robot systems: ROI with exclusion zone (left), headland boundaries defining the operable area in shaded region (center-left), parallel swath decomposition (center-right), and task allocation to three robots shown in green, blue, and red (right).

B. Multi-Robot Swath Allocation

The multi-robot swath allocation module distributes the set of minimum-turn swaths among the robots to balance the overall mission workload and minimize the total mission completion time. For this purpose, each swath $S_m = [a_m, b_m]$ is assigned a centroid as:

$$c_m = \frac{a_m + b_m}{2}, \quad (13)$$

which serves as a representative geometric point for the swath and is used to determine its relative position within the region. The swaths are then ordered according to the projection $z_m = c_m \cdot v$, which sorts the swaths along the sweep normal direction and enforces a consistent global traversal order across the field. Depending on the robot heading angle, it enters the swath of its two endpoints, $E_m = \{a_m, b_m\}$. To estimate the true travel cost between two swath endpoints, feasible transitions inside F' are defined as:

$$d_{F'}(x, y) = \begin{cases} \|x - y\|, & xy \subset F', \\ \text{VRM shortest path,} & \text{otherwise.} \end{cases} \quad (14)$$

As shown in Eqn. (14), if a direct straight-line segment lies entirely inside the feasible region, the Euclidean distance is used; otherwise, the shortest collision-free path obtained from the visibility–roadmap is substituted.

Next, the transition cost between two swaths is computed as:

$$c_{m,m'} = \min_{x \in E_m, y \in E_{m'}} d_{F'}(x, y). \quad (15)$$

This cost selects the minimum feasible transition distance over all possible entry–exit endpoint combinations of the two swaths.

Finally, we assign swaths to multiple robots by minimizing the maximum individual robot travel cost. To balance the workload across the robot fleet, we model this task as a multiple Traveling Salesman Problem (mTSP) with open tours:

$$\min_{\Pi_1, \dots, \Pi_{N_R}} \max_r L_r, \quad (16)$$

where L_r includes both the swath lengths and the transition distances for robot r . We use the LKH solver [29] to efficiently solve the mTSP problem in Eqn. (16).

C. Sweep Path Generation using Alternating-Edge Traversal and Visibility–Roadmap (VRM) Detouring

Once swath assignments step is completed, we then compute continuous coverage paths using alternating-edge traversal and visibility–roadmap (VRM) based detouring. Let $I_r = \{k_1, k_2, \dots, k_{|I_r|}\}$ denote the swaths assigned to robot r . This index set represents the ordered subset of swaths allocated to robot r by the mTSP solver. To generate a continuous sweep path, we choose the waypoints based on the robot heading direction along a swath segment in such a way that produces back-and-forth traversal along two consecutive swaths as:

$$(w_{2i-2}^{(r)}, w_{2i-1}^{(r)}) = \begin{cases} (a_{k_i}, b_{k_i}), & i \text{ odd,} \\ (b_{k_i}, a_{k_i}), & i \text{ even.} \end{cases} \quad (17)$$

This avoids unnecessary reorientation.

This produces the preliminary waypoint sequence

$$R_r = \langle w_0^{(r)}, \dots, w_{2|I_r|-1}^{(r)} \rangle. \quad (18)$$

The sequence R_r represents the ordered set of entry and exit waypoints for robot r before obstacle-aware refinement is applied.

The visibility–roadmap graph $G_{\text{VRM}} = (V_{\text{VRM}}, E_{\text{VRM}})$ is constructed using all swath endpoints together with uniformly sampled intermediate points along the buffered obstacle and boundary polygons. Edges are added between mutually visible vertex pairs that lie entirely inside the feasible region F' , enabling efficient computation of shortest collision-free transition paths between disconnected swaths [5]. Straight-line transitions are used whenever feasible; otherwise, a VRM shortest path is computed.

The refined coverage plan for robot r is defined as

$$\Pi_r = \text{VRMRefine}(R_r), \quad (19)$$

where Π_r denotes the complete coverage path for robot r obtained by replacing obstructed straight-line transitions in R_r with VRM-based shortest paths.

The final multirobot coverage solution is

$$\{\Pi_1, \Pi_2, \dots, \Pi_{N_R}\},$$

which represents the set of executable coverage paths for all robots in the team.

Algorithm 1 Minimum-Turn Swath Generation

Require: ROI P , exclusion zones $\{o_i\}$, headland width h , swath width w

Ensure: Set of feasible swaths $S = \{S_m\}$

- 1: Compute convex hull $H = \text{ConvHull}(P)$.
- 2: Using rotating calipers, obtain minimum-area rectangle R_{\min} and axes (u, v) (see Sec. II).
- 3: Compute inward and outward Minkowski offsets using (Eq. 10) and (Eq. 11).
- 4: Form feasible region F' using (Eq. 12).
- 5: Compute projection extrema η_{\min} and η_{\max} using (Eq. 3)–(Eq. 4).
- 6: Compute number of swaths n_s via (Eq. 5).
- 7: **for** $k = 1$ to n_s **do**
- 8: Compute swath center offset c_k using (Eq. 6).
- 9: Define the swath line $x_k(t)$ as described in IV-B
- 10: Intersect x_k with polygon edges using (Eq. 7) and (Eq. 8).
- 11: Sort all valid intersection points.
- 12: Form candidate segments $S_{k,\nu}$ using (Eq. 9).
- 13: Accept segments whose midpoint lies inside P (even–odd rule).
- 14: **end for**
- 15: Renumber all accepted segments as $S = \{S_m\}$.
- 16: **return** S .

VI. ALGORITHM ANALYSIS

Unlike metaheuristics for coverage planning, proposed MRCPP utilizes an mTSP algorithm to generate optimal coverage paths for multiple robots. Here, we will show that the coverage paths generated by the MRCPP algorithm are the shortest coverage paths. We will start with following lemma:

Lemma 1 (Subpaths of shortest paths are shortest paths [30]). *Given a weighted, directed graph $G = (V, E)$ with weight function $w : E \rightarrow \mathbb{R}$, let $p = \langle v_0, v_1, \dots, v_k \rangle$ be a shortest path from vertex v_0 to vertex v_k . For any i and j such that $0 \leq i \leq j \leq k$, let $p_{ij} = \langle v_i, v_{i+1}, \dots, v_j \rangle$ be the subpath of p from vertex v_i to vertex v_j . Then p_{ij} is a shortest path from v_i to v_j .*

In [30] authors showed that if we have a shortest path from v_0 to v_k , then the middle portion from v_i to v_j must be the shortest possible path between those two vertices. Here, we extend this Lemma 1 to multirobot coverage path problem as follows:

Theorem 1 (Subpaths of shortest minimum-turn coverage paths are shortest minimum-turn coverage paths). *Given a weighted, directed graph $G = (V, E)$ with weight function $w : E \rightarrow \mathbb{R}$, let $P_{\text{cov}} = \langle x_0, x_1, \dots, x_k \rangle$ be a shortest minimum-turn coverage path from x_0 to x_k , where x_i are consecutive coverage waypoints generated from the ordered swath endpoints and VRM transitions. For any indices i and j such that $0 \leq i \leq j \leq k$, let $P_{\text{cov}}^{ij} = \langle x_i, x_{i+1}, \dots, x_j \rangle$ be the assigned coverage subpath of P_{cov} from x_i to x_j by the mTSP algorithm. Then P_{cov}^{ij} is a shortest minimum-turn coverage path from x_i to x_j .*

Proof. We decompose an ROI P into a set of minimum-turn swaths S . If the mTSP algorithm optimally assigns S into

$$x_0 \xrightarrow{p_{0i}} x_i \xrightarrow{p_{ij}} x_j \xrightarrow{p_{jk}} x_k,$$

then we have

$$w(p) = w(p_{0i}) + w(p_{ij}) + w(p_{jk}).$$

Assume that there exists a minimum-turn coverage path p'_{ij} from x_i to x_j with weight $w(p'_{ij}) < w(p_{ij})$. Then the coverage path

$$x_0 \xrightarrow{p_{0i}} x_i \xrightarrow{p'_{ij}} x_j \xrightarrow{p_{jk}} x_k$$

has weight

$$w(p_{0i}) + w(p'_{ij}) + w(p_{jk}) < w(p),$$

contradicting the assumption that P_{cov} is a shortest minimum-turn coverage path from x_0 to x_k . \square

VII. EXPERIMENTAL RESULTS

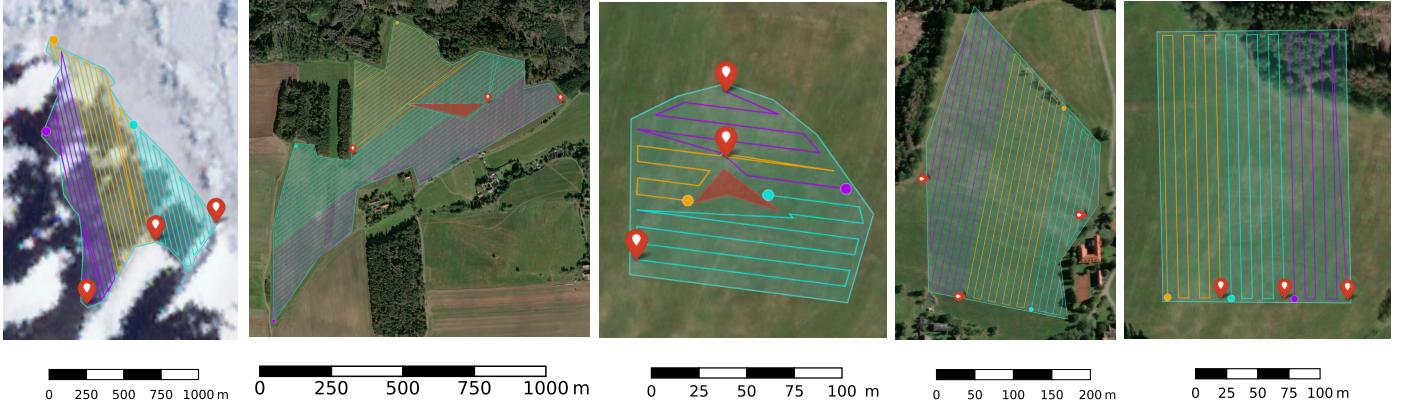
To validate the effectiveness of our proposed method, we conducted both simulated and real-world experiments. We compared our approach against state-of-the-art methods in simulation environments. For energy consumption estimation E_o , we adopt the trajectory-free algorithm proposed by [10], which estimates energy without requiring full trajectory generation by analyzing turn properties and speed profiles along path segments. These simulation experiments demonstrated the accuracy of this energy estimation algorithm when compared to estimations E_t derived from trajectory-based approaches such as TOPPRA [31]. We performed a comprehensive series of real-world experiments to further evaluate our method’s performance.

All experiments were conducted on a desktop computer running Ubuntu 22.04 LTS, equipped with an AMD Ryzen™ 9 7950X processor, an NVIDIA GTX 4070 GPU, and 32 GB of RAM. For the simulation experiments, we utilized a custom-built AAV based on the Tarot T650 frame with the same motor-propeller configuration as [10]. (Tarot 4114 320KV motor paired with Tarot 1555 carbon fiber propeller).

Following the approach in [10], the calculated optimal speed for this AAV platform was determined to be 8.39 m/s, which closely approximates the maximum autonomous flight speed achievable with the MRS system. This velocity was consistently applied across all simulation experiments as the maximum speed input allowed to the trajectory generation algorithm. The energy estimation requires platform-specific parameters that characterize the AAV’s power consumption profile. We use the values empirically determined by [10] for this configuration: reference velocity $v_r = 8.39$ m/s, hovering power $P_h = 426.03$ W, reference power $P_r = 465.23$ W, and maximum acceleration $a_{\max} = 2$ m/s².

A. Benchmark

The performance of our proposed MRCPP framework was evaluated through two sets of experiments: (i) comparative benchmarking against state-of-the-art planners across diverse environments, and (ii) a scalability assessment in



(a) Cape (b) Complex (c) Island (d) Simple (e) Rectangle
 Fig. 3: This figure shows representative benchmark environments with coverage paths for three AAVs in yellow, violet, and cyan. The AAV paths are generated by our MRCPP method within the given polygons. In Figures 3b and 3c, No-Fly Zones (NFZs) are shown as shaded red polygons. Scale bars are shown in meters.

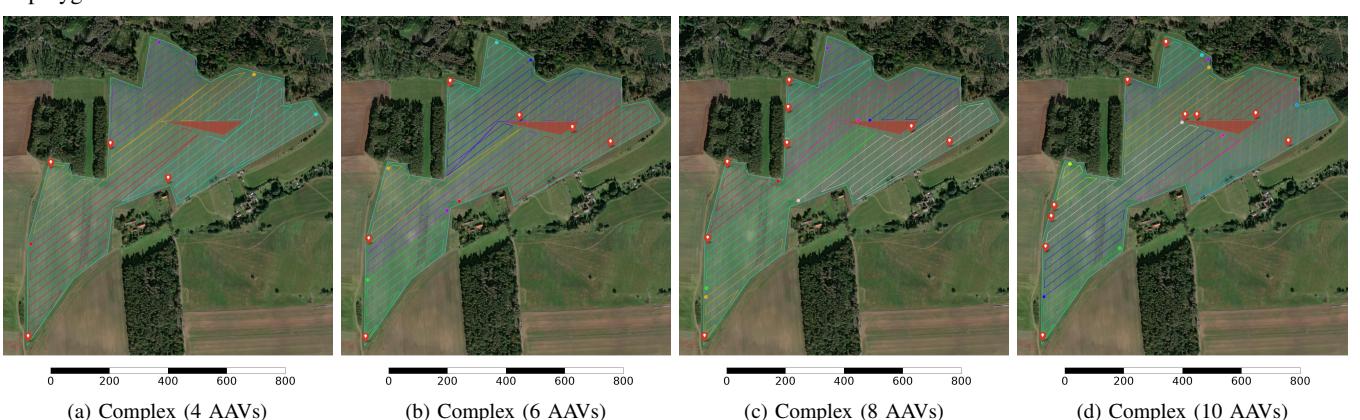


Fig. 4: This figure demonstrates the scalability of our MRCPP method by showing coverage path planning with varying numbers of AAVs (4, 6, 8, and 10) in the same polygon. Different colored paths represent different AAV trajectories.

which the number of AAVs increases while operating in a fixed NFZ-constrained scenario. All planners were evaluated using the same aerodynamic energy model and trajectory-parameterization settings to ensure fair comparison. POP-CORN+SALT [9] is excluded from the scalability analysis because it does not support NFZ constraints.

For benchmarking, we compared our method MRCPP against three state-of-the-art CPP frameworks: (i) the EAM-CMP algorithm from [10] (C++), (ii) the POPCORN+SALT framework [9] (Python), and (iii) the DARP+MST solution approach from [8] (Java). Each algorithm was executed to generate coverage paths for every test environment. The benchmark environments, adopted from [10], include six distinct scenarios with varying geometric complexity and operational constraints: *Cape* represents Cape Crozier on Ross Island in Antarctica with a 31m operational footprint, one of the areas of interest used in [14, 10]; *Island* features a simple shape with one No-Fly-Zone (NFZ) in the middle and an 8m operational footprint width; *Rect* is a rectangle with no NFZ and a 4m operational footprint width; *Complex 22* presents a complex shape with an NFZ and a 22m operational footprint width; *Complex 12* uses the same shape as Complex 22 but with a 12m operational footprint width; and *Simple* is a simple shape with no NFZ and a 9m operational footprint width. Figure 3

illustrates all scenarios with MRCPP-generated paths, each annotated with a metric scale in meters.

Table I summarizes the benchmark results for these environments. Across nearly all scenarios, MRCPP achieves the lowest computation time, often outperforming other planners by large margins. For instance, MRCPP completes planning in 0.503 s for *Cape* and 0.884 s for *Complex 22*, whereas DARP+MST requires 33.818 s and 18.217 s, respectively. EAMCMP occasionally attains faster runtimes in the smallest environments, but its performance is less stable as geometric complexity increases.

Energy consumption exhibits a similarly strong trend. MRCPP achieves the lowest E_o and E_t in all environments except *Complex 12*, where DARP+MST obtains a slightly lower E_t (434.163 Wh vs. 440.339 Wh). However, DARP+MST requires more than triple the computation time and only marginally reduces the geometric path length. In every other environment, MRCPP produces the most energy-efficient coverage paths. For example, in *Simple*, MRCPP reduces average per-AAV energy ($E_{o_{avg}}$) by more than 40% compared to EAM-CMP and by approximately 23% compared to DARP+MST.

Average per-AAV energy values ($E_{o_{avg}}$ and $E_{t_{avg}}$) further illustrate the quality of workload distribution. In all benchmark environments, MRCPP provides the lowest average per-AAV

Scenario	Planner	Time [s]	E_o [Wh]	E_t [Wh]	$E_{o_{avg}}$ [Wh]	$E_{t_{avg}}$ [Wh]	Path [km]
Cape	MRCPP (Ours)	0.503	500.780	502.985	166.927	167.662	31.187
	EAMCMP[10]	4.905	594.200	567.693	198.067	189.231	36.061
	DARP+MST[8]	33.818	627.110	578.055	209.037	192.685	35.397
	POPCORN+SALT[9]	55.872	689.690	607.341	229.897	202.447	30.916
Complex 12	MRCPP (Ours)	13.960	485.610	440.339	161.870	146.780	28.185
	EAMCMP[10]	0.073	571.160	539.665	190.387	179.888	33.358
	DARP+MST[8]	45.307	487.400	434.163	162.467	144.721	24.965
	POPCORN+SALT[9]	—	—	—	—	—	—
Complex 22	MRCPP (Ours)	0.884	271.670	244.131	90.557	81.377	15.731
	EAMCMP[10]	1.246	419.690	401.813	139.897	133.938	23.549
	DARP+MST[8]	18.217	272.440	253.906	90.813	84.635	13.609
	POPCORN+SALT[9]	—	—	—	—	—	—
Island	MRCPP (Ours)	0.177	28.930	28.048	9.643	9.349	1.272
	EAMCMP[10]	0.275	36.800	35.662	12.267	11.887	1.596
	DARP+MST[8]	5.205	32.960	28.585	10.987	9.528	1.325
	POPCORN+SALT[9]	—	—	—	—	—	—
Rect	MRCPP (Ours)	0.205	67.550	64.948	22.517	21.649	3.708
	EAMCPP[10]	0.017	85.160	78.815	28.387	26.272	4.347
	DARP+MST[8]	4.746	87.560	103.515	29.187	34.505	3.807
	POPCORN+SALT[9]	20.714	131.000	119.013	43.667	39.671	3.712
Simple	MRCPP (Ours)	0.248	108.790	108.411	36.263	36.137	6.186
	EAMCMP[10]	0.018	183.210	173.527	61.070	57.842	10.283
	DARP+MST[8]	5.700	142.050	133.504	47.350	44.501	6.288
	POPCORN+SALT[9]	32.283	230.700	212.720	76.900	70.907	6.579

TABLE I: Comparison of all evaluated methods across multiple environments for a team of three AAVs. The reported metrics include: total and average energy consumption during trajectory generation, total and average energy consumption without trajectory generation, the resulting path length, and the overall computation time. Best values for each environment are highlighted in **bold**. A dash (—) indicates that no solution exists for that method in the specific environment.

Scenario	MRCPP				EAMCMP[10]				DARP+MST[8]			
	t_c [s]	$E_{o_{avg}}$ [Wh]	$E_{t_{avg}}$ [Wh]	ℓ [km]	t_c [s]	$E_{o_{avg}}$ [Wh]	$E_{t_{avg}}$ [Wh]	ℓ [km]	t_c [s]	$E_{o_{avg}}$ [Wh]	$E_{t_{avg}}$ [Wh]	ℓ [km]
complex (4 AAVs)	3.36	67.26	59.75	15.60	0.98	75.82	70.75	16.83	14.40	69.64	62.08	13.86
complex (5 AAVs)	0.84	54.01	47.52	15.71	1.25	62.14	57.43	17.29	22.88	54.87	48.82	13.49
complex (6 AAVs)	0.95	43.95	39.24	15.33	1.05	51.97	48.01	17.36	37.04	49.68	43.92	15.10
complex (7 AAVs)	1.03	35.93	32.48	14.51	1.50	43.04	41.18	17.12	56.13	44.24	38.98	15.64
complex (8 AAVs)	1.22	32.12	28.46	14.91	1.14	40.24	37.98	18.39	66.04	40.17	35.41	15.96
complex (9 AAVs)	1.00	27.10	24.30	14.11	0.73	39.82	38.17	20.56	48.32	37.77	33.38	16.57
complex (10 AAVs)	1.46	24.25	21.87	14.00	0.07	57.12	53.97	33.36	—	—	—	—

TABLE II: Scalability comparison in the Complex 22 environment for fleets ranging from 4 to 10 AAVs. The table reports computation time (t_c), average energy consumption per AAV ($E_{o_{avg}}$ and $E_{t_{avg}}$), and total coverage path length (ℓ) for MRCPP (Ours), EAMCMP[10], and DARP+MST[8].

energy. In *Complex 22*, for example, MRCPP requires only 90.557 Wh per AAV, compared to 139.897 Wh for EAMCMP and 90.813 Wh for DARP+MST.

Path-length results reinforce these observations. Although POPCORN+SALT occasionally yields the shortest geometric paths (e.g., *Cape*), MRCPP remains competitive across all environments. In structured environments such as *Island*, *Rect*, and *Simple*, MRCPP generates the shortest overall coverage routes.

Table II reports the scalability results as the number of AAVs increases from 4 to 10 within the same NFZ-constrained *Complex 22* environment. MRCPP demonstrates strong scalability in all metrics. The computation time remains low (0.84–3.36 s). By contrast, DARP+MST exhibits sharp runtime escalation, exceeding 56 s for 7 AAVs and 66 s for 8 AAVs. EAMCMP shows low runtimes for the smallest and largest fleet sizes, but MRCPP’s runtime was competitive.

Energy scalability results highlight the benefits of MRCPP’s integrated geometric reasoning and balanced allocation. MRCPP achieves the lowest average energy per AAV for every fleet size, improving $E_{o_{avg}}$ by approximately 11–58% compared to EAMCMP and 1.5–28% compared to DARP+MST. Moreover, MRCPP’s energy usage decreases monotonically as the number of AAVs increases, reflecting consistently balanced

task distribution. In contrast, EAMCMP and DARP+MST exhibit slower or irregular energy reductions due to uneven partitioning and longer ferrying transitions.

Path-length results show a complementary pattern. MRCPP generates the shortest coverage paths for 7–10 AAVs and remains highly competitive for smaller fleets. While DARP+MST occasionally produces shorter geometric paths in low-AAV cases (4–6), these gains come at the cost of significantly higher computation time and greater energy expenditure. EAMCMP consistently produces the longest paths due to its less efficient coverage decomposition.

B. Real World Experiment

We conducted two categories of real-world experiments to validate the proposed framework: aerial coverage using AAVs and surface coverage using ASVs. For the aerial experiments, we deployed two DJI Mavic Air AAVs to cover a polygonal Region of Interest (ROI). We uploaded the planned coverage paths to the Litchi mission-planning software, which then executed each trajectory autonomously on the corresponding AAV. The two AAVs together covered 615 m, with AAV 1 flying 306 m and AAV 2 flying 309 m. Because we deployed both vehicles concurrently, they completed the mission in 78 s. Their average flight speeds were 4.25 m/s for AAV 1 and

4.41 m/s for AAV 2. As shown in Fig. 1, the executed paths appear in orange and blue, and the circular no-fly zone (NFZ) appears in red. Some segments of the blue and orange paths are not perfectly straight because wind disturbances affected the AAVs during flight.

We further evaluated the proposed method in a coastal bay environment using two autonomous surface vehicles (ASVs), each equipped with a YSI EXO2 sonde for in situ water-quality sensing. The ASVs were assigned to cover a designated Region of Interest (ROI) analogous to the aerial experiments. As shown in Fig. 1b, the left panel presents a live video feed from the ASVs, while the right panel displays their real-time positions and executed coverage paths overlaid on satellite imagery. This interface facilitated real-time monitoring and verification of coverage execution throughout the mission. The path length for ASV1 was 138.15 m (marked in red), and the path length for ASV2 was 125.71 m (marked in green), as shown in the right panel.

VIII. CONCLUSION

This letter introduces an efficient multi-robot coverage path planning framework for complex environments with obstacles and no-fly zones. The proposed framework achieves this through optimal orientation swath generation, obstacle-aware path connection, and workload-balanced task assignment to generate optimal coverage paths. The framework is evaluated on several key metrics, including computation time, path length, and energy consumption (both trajectory-based and trajectory-free). Extensive simulations demonstrated that the proposed method outperformed state-of-the-art planners by an order of magnitude. The framework is validated in experiments using multiple AAVs and ASVs. These experiments confirm the framework's capability to generate reliable, well-balanced coverage plans under realistic conditions. Future work will include trajectory planning to efficiently avoid dynamic obstacles during full coverage in complex environments.

REFERENCES

- [1] D. I. Koutras, A. C. Kapoutsis, and E. B. Kosmatopoulos, "Autonomous and cooperative design of the monitor positions for a team of uavs to maximize the quantity and quality of detected objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4986–4993, 2020.
- [2] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, "A distributed, plug-n-play algorithm for multi-robot applications with a priori non-computable objective functions," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 813–832, 2019.
- [3] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khereishah, and M. Guizani, "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges," *IEEE access*, vol. 7, pp. 48 572–48 634, 2019.
- [4] D. Erdos, A. Erdos, and S. E. Watkins, "An experimental uav system for search and rescue challenge," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 5, pp. 32–37, 2013.
- [5] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.
- [6] R. Bähnemann, N. Lawrence, 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*. Springer, 2021, pp. 277–290.
- [7] 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, no. 3, pp. 663–680, 2017.
- [8] S. D. Apostolidis, P. C. Kapoutsis, A. C. Kapoutsis, and E. B. Kosmatopoulos, "Cooperative multi-uav coverage mission planning platform for remote sensing applications," *Autonomous Robots*, vol. 46, no. 2, pp. 373–400, 2022.
- [9] K. Shah, A. E. Schmidt, G. Ballard, and M. Schwager, "Large scale aerial multi-robot coverage path planning," *Field Robotics*, vol. 2, pp. 1971–1998, 2022.
- [10] D. Datsko, F. Nekovar, R. Penicka, and M. Saska, "Energy-aware multi-uav coverage mission planning with optimal speed of flight," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2893–2900, 2024.
- [11] 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.
- [12] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and service robotics*. Springer, 1998, pp. 203–209.
- [13] C. Di Franco and G. Buttazzo, "Coverage path planning for uavs photogrammetry with energy and resolution constraints," *Journal of Intelligent & Robotic Systems*, vol. 83, no. 3, pp. 445–462, 2016.
- [14] K. Shah, G. Ballard, A. Schmidt, and M. Schwager, "Multidrone aerial surveys of penguin colonies in antarctica," *Science Robotics*, vol. 5, no. 47, p. eabc3000, 2020.
- [15] M. Fischetti, J. J. S. González, and P. Toth, "The symmetric generalized traveling salesman polytope," *Networks*, vol. 26, no. 2, pp. 113–123, 1995.
- [16] L. Bauersfeld and D. Scaramuzza, "Range, endurance, and optimal speed estimates for multicopters," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2953–2960, 2022.
- [17] F. Nekovář, J. Faigl, and M. Saska, "Multi-tour set traveling salesman problem in planning power transmission line inspection," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6196–6203, 2021.
- [18] I. Vandermeulen, R. Groß, and A. Kolling, "Turn-minimizing multirobot coverage," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1014–1020.
- [19] A. Stefanopoulou, E. K. Raptis, S. D. Apostolidis, S. Gkelios, A. C. Kapoutsis, S. A. Chatzichristofis, S. Vrochidis, and E. B. Kosmatopoulos, "Improving time and energy efficiency in multi-uav coverage operations by optimizing the uavs' initial positions," *International Journal of Intelligent Robotics and Applications*, vol. 8, no. 3, pp. 629–647, 2024.
- [20] I. Maza and A. Ollero, "Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms," in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 221–230.
- [21] A. Barrientos, J. Colorado, J. d. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," *Journal of Field Robotics*, vol. 28, no. 5, pp. 667–689, 2011.
- [22] D. Puig, M. A. García, and L. Wu, "A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 635–653, 2011.
- [23] A. Breitenmoser, M. Schwager, J.-C. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 4982–4989.
- [24] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [25] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM computing surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.
- [26] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1, pp. 77–98, 2001.
- [27] N. Agmon, N. Hazon, and G. A. Kaminka, "Constructing spanning trees for efficient multi-robot coverage," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE, 2006, pp. 1698–1703.
- [28] G. T. Toussaint, "Solving geometric problems with the rotating calipers," in *Proc. IEEE Melecon*, vol. 83, no. 83, 1983, p. A10.
- [29] K. Helsgaun, "An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems," *Roskilde: Roskilde University*, vol. 12, pp. 966–980, 2017.
- [30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.
- [31] H. Pham and Q.-C. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, 2018.