Coverage Path Planning Using a Group of UAVs

Bouras Abdelwahhab[1][a], Bouzid Yasser[2][b], Cherifi Youcef[2] and Guiatni Mohamed[2][c]
[1]*Ecole Supérieure Ali Chabati, Reghaia, Algiers, Algeria*
[2]*Ecole Militaire Polytechnique, Bordj El Bahri, Algiers, Algeria*
fi

Keywords: Coverage Path Planning, Unmanned Aerial Vehicles, Traveling Salesman Problem, Genetic Algorithms.

Abstract: This article introduces a novel methodology of path planning within a group of Unmanned Aerial Vehicles (UAVs) for aerial detection. The primary aim of this method is to ensure comprehensive coverage of a designated Region of Interest (RoI) while taking measurements from the entire region. The proposed methodology operates through a structured yet adaptive three-phase process. First, the RoI is transformed into a discrete representation using a meshing algorithm, ensuring a well-defined and homogeneous spatial structure for subsequent planning. This discretized space is then well partitioned into subregions via the K-means clustering algorithm, optimizing workload distribution among UAVs while preserving spatial coherence. Finally, the path of each UAV is formulated as a Traveling Salesman Problem (TSP) and solved using an enhanced Genetic Algorithm (GA). Specifically, this GA is tailored to accelerate convergence and yield optimized paths. The principal advantages of the proposed method, as demonstrated through simulation experiments, are its optimization capabilities, flexibility and reduction in computational time.

1 INTRODUCTION

Over the past two decades, the rapid advancement in sensor technology and embedded systems for Unmanned Aerial Vehicles (UAVs) has markedly expanded their applicability across a range of domains, both civilian and military. These applications include surveillance, photogrammetry, the Internet of Things (IoT), search and rescue operations, and agricultural tasks, etc (see Fig. 1a and Fig. 1b). Additionally, the deployment of UAV fleets mitigates the limitations inherent in single UAV operations, enhancing overall efficiency. This has numerous advantages, particularly in scenarios that pose significant risks to human operators (refer to Fig. 1c), such as reducing execution time and minimizing human resource involvement. Nonetheless, each application presents its technical challenges, with control, optimization and path planning being critical areas requiring advancement.

This paper focuses on coverage path planning (CPP) (Choset, 2001) using a UAV group (Cabreira et al., 2019) (Bouras et al., 2022) (Kumar and Kumar, 2023) (Chen et al., 2024). Each UAV must systematically measure a predefined set of points to gather information essential for achieving comprehensive coverage while circumventing obstacles. This necessitates the employment of spatial division techniques, the generation of Points of Interest (PoI), and the optimization of planned routes. Moreover, optimizing these procedures in terms of energy and time (Yu and Lee, 2024) is crucial.

Typically, research addressing the Multi-Coverage Path Planning (mCPP) problem within a Region of Interest (RoI) follows a four-step process (Bouras et al., 2019): 1) discretizing the spatial domain; 2) distributing the workload among UAVs based on their flight autonomy; 3) employing a planning algorithm to define the paths; and 4) smoothing the resultant routes.

In reference (Kapoutsis et al., 2017), the authors tackle the coverage problem by discretizing the space into small square cells and establishing rules to ensure equitable distribution of the RoI, thereby avoiding redundancy and ensuring comprehensive coverage. The paths, determined by the Spanning Tree Coverage (STC) algorithm, navigate around obstacles such as trees. While effective, the STC algorithm's speed decreases with more UAVs, increasing memory demands and causing suboptimal paths due to frequent turns. This issue was addressed in (Gao et al., 2018) through optimization via Ant Colony Optimization (ACO), which improved path efficiency.

[a] https://orcid.org/0000-0003-3518-6821
[b] https://orcid.org/0000-0002-8400-9912
[c] https://orcid.org/0000-0002-5899-6862

Abdelwahhab, B., Yasser, B., Youcef, C. and Mohamed, G.
Coverage Path Planning Using a Group of UAVs.
DOI: 10.5220/0013421200003970
In *Proceedings of the 15th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2025)*, pages 159-165
ISBN: 978-989-758-759-7; ISSN: 2184-2841

(a) (b) (c)

Figure 1: Examples of UAV fleet applications: (a) Agricultural uses, (b) Internet of Things (IoT), (c) Fire forest uses.

In the work by H. Azpúrua et al. (Azpúrua et al., 2018), the RoI is partitioned into hexagonal cells, offering greater connectivity to adjacent cells and minimizing inter-cell distances, thus reducing the Total Path Length (TPL). The planning is modeled as a Traveling Salesman Problem (TSP), the results illustrate the impact of UAV involved and coverage line distance on mission execution time. However, the optimization remains requires improvements due to the non-exact discretization of the RoI.

Another study (Balampanis et al., 2017) employs triangulation to mesh the RoI, aligning the triangles with the sensor's footprint. UAVs are assigned subregions based on their flight autonomy using two proposed algorithms: Wave Front Propagation and Reverse Watershed Scheme (AWP & RWS). This approach also incorporates a Dead Lock Handling (DLH) algorithm to prevent no-fly zones. However, the analysis does not address the number of turns or the energy consumption associated with these paths.

This work introduces a novel three-phase methodology that enhances the efficiency of multi-UAV path planning while addressing the limitations of existing scenario-specific approaches. Unlike traditional methods with restricted applicability, our approach offers greater flexibility in discretization, sensor selection, and path optimization. A key contribution is the precise discretization of the RoI, overcoming the approximations common in prior techniques. This structured representation enables optimized task allocation via K-means clustering. Additionally, an enhanced Genetic Algorithm (GA) with problem-specific mutation operators accelerates convergence and improves solution quality. These advancements collectively lead to faster convergence, greater scalability, superior path optimization, and increased computational efficiency compared to standard GA-based methods.

The structure of this article is organized as follows: Section II and Section III delineate the methodology and provide a detailed description of the algorithms employed. Section IV is devoted to presenting the results obtained from the simulations. Finally, Section V concludes the paper with a discussion of the results and outlines potential avenues for future research.

2 METHODOLOGY

2.1 Preliminary

Coverage of the RoI using UAVs involves the generation of PoIs distributed over the surface of this RoI. The quantity and spatial distribution of these points are primarily influenced by the desired extent of coverage (either full or partial) and the range of the onboard sensors. Each UAV is allocated a finite set of PoIs based on its flight capabilities. The objective is to take measurements at each PoI by tracing an optimal path that connects these points. The mission begins at an initial point, denoted as "***Start***", and concludes at a "***Goal***" point, with the caveat that each UAV ultimately returns to the starting point, effectively considering them as confused, in other words, each UAV ends the coverage mission by returning to the starting point.

To achieve this, we model the problem representing our RoI as a closed region delineated by polygonal segments, which may be either convex or nonconvex. Obstacles are present only in the area outside the RoI but within the operational workspace. The configuration of the RoI's shape and the number of UAVs are user-defined parameters, and each UAV's flight autonomy must be sufficient to accomplish the assigned mission.

The UAVs employed are considered rotary-wing aircraft, specifically quadcopters. Their inherent maneuverability, hovering capability, and ease of control make them well-suited for such missions, particularly when equipped with various types of embedded sensors (as illustrated in Fig. 2).

2.2 Mathematical Formulation

Let $\mathcal{P} \subseteq \mathbb{R}^2$ represent the workspace designated for coverage, with dimensions $[l \times w]$, where l and w denote the length and width of $\mathcal{P}$, respectively.

Define $\mathcal{A} \subseteq \mathbb{R}^2$ as the RoI characterized by a list of n segments $\mathcal{S} = \{s_1, s_2, ..., s_n\}$, and $\mathcal{A} \subseteq \mathcal{P}$, $\mathcal{O} \subseteq \mathbb{R}^2$ and $\mathcal{O} \subseteq \mathcal{P}/\mathcal{A}$ is the non-fly zone or $\mathcal{O} = \sum_{i=0}^{z} o_i$, such that o_i represents the i^{th} from z small obstacles located between $\mathcal{A}$ and $\mathcal{P}$.

Let $c_i \in \mathcal{A}$ denote the set of PoIs, where $c_i = (x_i, y_i)$ for $i \in \{0, \ldots, r\}$ and r is the number of the PoIs. Specifically, $c_0 = (x_0, y_0)$ represents the initial position of the UAVs' parking platform, with coordinates x_0, y_0. Let $u \subseteq \mathbb{N}^*$ denote the number of UAVs,

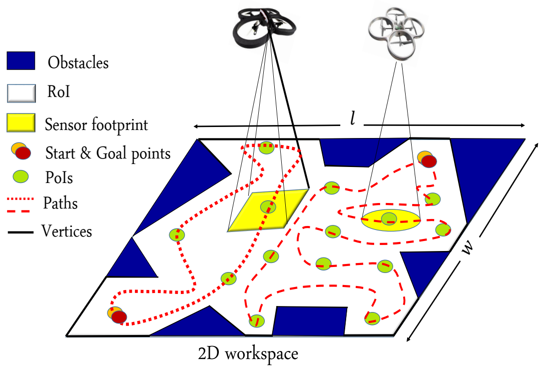

Figure 2: Coverage using a group of UAVs.

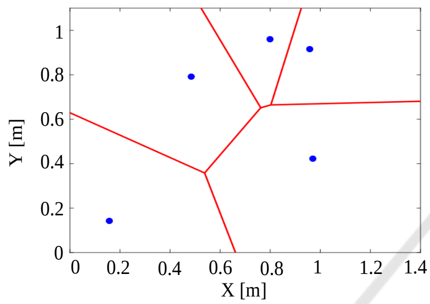

Figure 3: Voronoi diagram.

and let R_{sf} signify the radius of the sensor footprint, which may take the form of either a square $[l_{sf} \times l_{sf}]$ or a rectangle $[l_{sf} \times w_{sf}]$, where l_{sf} and w_{sf} represent the length and width of the sensor footprint, respectively.

In the initial phase, following the selection of $\mathcal{P}$, $\mathcal{A}$ and r, we proceed to generate the PoIs within the RoI $\mathcal{A}$. This generation process utilizes the Voronoi diagram, defined as follows:

$$\mathcal{V} = \left\{ y \in \Omega \mid d(y,\, x_i) \leq d(y,\, x_j),\, for\ j \neq i \right\} \quad (1)$$

This function partitions the space into sub-regions (referred to as cells when dealing with a large number of r), such that all points $p_i : p_i \in \Omega$ within each cell are closer to x_i than to x_j for $j \neq i$. Additionally, the grid that delineates the r cells consists of the bisectors of the line segments $[x_i, x_j]$ (Fig. 3).

The computation of the Voronoi diagram for an arbitrary set $\mathcal{X} = \{x_i \mid i = 0, \ldots, r\}$, followed by the integration of each resulting cell and the determination of its centers, constitutes a version of Voronoi Tessellation known as the ***Lloyd*** algorithm (Du et al., 1999). These centers represent our PoIs, with their quantity being chosen according to the coverage requirements and the type of sensors employed. A comprehensive description of this algorithm is provided in the subsequent section (Algorithm 1).

The second step involves distributing these PoIs among UAVs. It is crucial to achieve a homogeneous distribution that considers the UAVs' autonomies and ensures an appropriate grouping of these points to prevent collisions and balance the loads. This step is facilitated by the ***K-means*** algorithm.

In the third step, we aim to determine the optimal sequence of PoIs $\mathcal{C} = \left\{c_1, c_2, \ldots, c_i, \ldots, c_{r_{u_i}}\right\}$ for each UAV, which minimizes the TPL. Here, r_{u_i} represents the number of points assigned to the i^{th} UAV u_i. Given the flexibility in selecting the number and distribution of PoIs, it is assumed that the TPLs are within the operational range of the UAVs. Therefore, the objective is to minimize the following function:

$$TPL_{min} = \sum_{i=1}^{r_{u_i}-1} d(N_i,\, N_{i+1}) + d\left(N_{r_{u_i}},\, N_1\right) \quad (2)$$

where $d(N_i,\, N_{i+1})$ is the distance between points N_i and N_{i+1}. The optimization of these paths is ensured by adapting it as a TSP and solving it using a GA, whose details are provided in Algorithm 2.

3 ALGORITHMS DESCRIPTION

The primary focus of this section is to develop the previously outlined steps in the form of algorithms.

To generate PoIs, the RoI is partitioned into r cells using an adapted Meshing algorithm (Algorithm 1). Unlike traditional meshing techniques, this approach iteratively refines the spatial subdivision to enhance uniformity and adaptability. After initializing the input data, the algorithm constructs an initial Voronoi tessellation and determines the set $x_i \mid i = 0, \ldots, r$. At each iteration, the centroids c_j of the Voronoi cells are computed and reallocated as new distribution points for x_i, triggering a re-computation of the tessellation. This process continues until a predefined convergence criterion is met or the maximum number of iterations is reached. As a result, the algorithm progressively produces more uniform cells, ultimately approximating a structured grid while preserving adaptability to the underlying spatial constraints.

The second stage consists of assigning the final cell centers $c_j \mid j = 0, \ldots, r$ to the UAVs by clustering them into m_u subgroups (Clusters), each containing r_{u_i} points, while minimizing the intra-cluster distances. This ensures an efficient workload distribution among UAVs. To achieve this, we employ ***K-means*** clustering algorithm, which optimally partitions the PoIs by minimizing the distance between each cluster

Let Ω be the working field, σ a density function on Ω, and r the number of generators. The initial set of generators is denoted as $x_i \mid i = 0, \dots, r$. The Meshing algorithm is defined as follows:

Input: $l, w, \mathcal{S}, r, \mathcal{X}, numIter, x_0, y_0$.

Function: $[v_r$, order of v_r, x_{c_j}, $y_{c_j}]$ = ***MeshingAlgorithm*** $(x_i, y_i, \mathcal{S}, numIter, r)$ *% v_r and its order represent the Voronoi cell borders and their arrangement, while x_{c_j}, y_{c_j} denote cell centroids.*

Initialization: Generate an initial set of r points $x_i \mid i = 0, \dots, r$.

while *the iteration count numIter is not reached* **do**

- **Voronoi Tessellation:** Compute the Voronoi diagram of Ω using the generator set $x_i \mid i = 0, \dots, r$.
- **Centroid Computation:** Determine the centroids $c_j \mid j = 0, \dots, r$ of the Voronoi cells.
- **Update:** Set $x_i \mid i = 0, \dots, r \leftarrow c_j \mid j = 0, \dots, r$ *% Replace the set x_i with the computed centroids c_j.*

end

Output: $[v_r$, order of $v_r, x_{c_j}, y_{c_j}]$

Algorithm 1: Meshing Algorithm.

center and its assigned points, as expressed by:

$$argmin_{\mathcal{M}} = \sum_{i=1}^{m_u} \sum_{c_j \in \mathcal{M}_i} \left|\left|c_j - \mu_i\right|\right|^2 \tag{3}$$

where μ_i represents the centroid of cluster $\mathcal{M}_i$.

The clustering process follows an iterative two-step refinement:

1. Cluster assignment: Each point c_j is assigned to the closest cluster center μ_i based on a Voronoi partitioning:

$$\mathcal{M}_i^{(t)} = \{c_j : \| c_j - \mu_i^{(t)} \| \leq \| c_j - \mu_{i^*}^{(t)} \| \forall i^* = 1, \dots, m_u\} \tag{4}$$

2. Centroid update: The cluster centers are recomputed as the mean of the assigned points:

$$\mu_i^{(t+1)} = (\mathcal{M}_i^{(t)})^{-1} \sum_{c_j \in \mathcal{M}_i^t} c_j \tag{5}$$

This iterative process ensures convergence by progressively reducing the cost function. Given the finite number of possible partitions, the algorithm is guaranteed to reach a stable solution, making it well-suited for UAV task allocation in large-scale environments.

The final stage involves determining the optimal sequence of points within each set $\mathcal{M}_i$ to minimize the total travel distance (Equation (2)) and establish the optimal paths $\mathcal{R}_i = \left\{c_0, \dots, c_{r_{u_i}}\right\}$. Given the combinatorial nature of this NP-hard problem, it is formulated as a TSP and efficiently solved using an enhanced GA. For instance, even a modest set of 10 points leads to 181 440 and 362 880 possible arrangements in symmetric and asymmetric TSPs, respectively, highlighting the necessity of an adapted optimization approach.

In this context, GA optimization involves evolving a population of chromosomes $\mathcal{C}h_i = \left\{H_1, \dots, H_{r_{u_i}}\right\}$, where each chromosome represents a candidate route $\mathcal{R}_i$. The fitness function evaluates each solution based on the total traveled path length (TPL):

$$f(C) = \sum_{i=1}^{r_{u_i}-1} d(H_i,\, H_{i+1}) + d\left(H_{r_{u_i}},\, H_1\right) \tag{6}$$

where r_{u_i} denotes the number of waypoints assigned to a single UAV.

To efficiently explore the $(n-1)!$ possible permutations, we introduce swap operations (flip, slide, and swap) (Pham et al., 2017), which significantly accelerate convergence, reduce computational complexity, and ensure feasible solutions even for large populations. Without these adaptive operators, the algorithm struggles to handle populations exceeding 16 elements, resulting in computation times exceeding $(> 2000\ sec)$ seconds.

Algorithm 2 summarizes the third step of our solution. Generating an initial population is necessary, where each chromosome is assigned a value f_i calculated by the fitness function. We then perform a selection within this population. Subsequently, new individuals in the population are created using crossover and mutation. Finally, we repeat the process.

The entire proposed technique and its algorithm sequence are summarized in Fig. 4. A detailed discussion of each step, based on the simulation results, is provided in the next section.

4 SIMULATION RESULTS AND DISCUSSION

In this section, we tested the effectiveness of the algorithms presented in the previous section using two scenarios. We selected a study region, which is polygon-shaped and located within a square space of dimensions $[3 \times 2]$ Km. In this region, we generated the PoIs, distributed evenly across the surface to be studied. The base station is positioned at $p_0(x_0, y_0) =$

Data: population size (r_{u_i}), Number of generations (n_g), Number of cities = size (N_p), Global min = Inf, $i \leftarrow 1$ *(i is the current iterations)*, $Pop(t) \leftarrow (r_{u_i})$ chromosomes $Pop_i(t)$.

while *not terminating condition* **do**
  while $i \leq r_{u_i}$ **do**
    $f_i \leftarrow f(Pop_i(t))$ *% f is the fitness function*
    while $i \leq r_{u_i}$ **do**
      $NewPop_i(t+1) \leftarrow$rand choose $Pop_i(t)$ from $Pop(t)$
      $CrossPop(t+1) \leftarrow$recombine $(NewPop(t+1))$ with P_c *% P_c is the crossover probability*
      $MutPop(t+1) \leftarrow$mutate $(CrossPop(t+1))$ with P_m *% P_m is the mutation probability*
      Switch
      case 1 mutate by flipping,
      case 2 mutate by swapping,
      case 3 mutate by sliding.
      $Pop(t+1) \leftarrow MutPop(t+1)$
      $i \leftarrow i+1$
    end
  end
end

Algorithm 2: Solving TSP-GA.

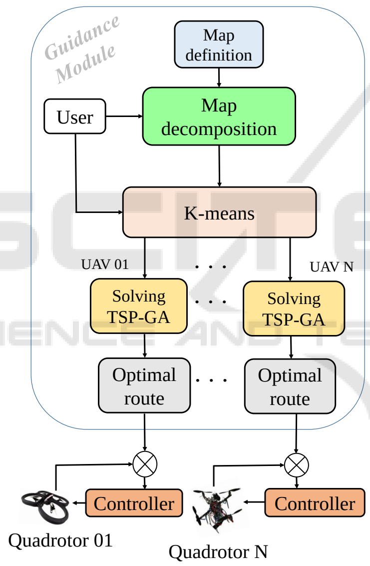

Figure 4: Summary of the technique proposed in this work.

$(1\ Km, 1\ Km)$. The parameters of the simulation tests are illustrated in Table 1.

Table 1: Simulation data.

Parametres	Scenario 1	Scenario 2
Number of UAVs (u)	01	04
Number of iterations $(numIter)$	80	180
Number of cities (N_p)	20	80
Population size (r_{u_i})	20	80
Number of generations (n_g)	100	150

In the first scenario, we generated 20 measurement PoIs. The Meshing algorithm required 80 iterations to establish the distribution of PoIs. Figure 5 (a) illustrates the outcome of this step, where lines extend from the base station to each cell center, representing the path during the execution of the Meshing algorithm. The final coordinates of the cell's centers obtained from this process are used in the subsequent step. However, their deployment paths are not considered in this study.

A single UAV is utilized to provide coverage, equipped with a circular sensor of radius $R_s = 200\ m$. The path optimized by the TSP-GA (blue line) and the resulting footprint coverage (red circles) are presented in Figure 5 (b). The optimization of the TPL with TSP-GA across generations is shown in Figure 5 (c).

From this first test, it appears that:

- The number of iterations is directly related to the number of PoIs generated. The number of PoIs depends on the type of application, the onboard sensor used, and the desired coverage quality (with or without overlaps).
- The TSP-GA is a crucial step for optimizing the TPL. Theoretically, there are 19! possible ways to connect the 20 PoIs and return to the initial point. Using TSP-GA, this is efficiently calculated, resulting in an overall distance of 7.7236 Km.

In the second scenario, 80 measurement points were generated for four UAVs over the same region. This setup provides a concrete demonstration of the advantages of using a group of UAVs in terms of coverage quality. Additionally, it allows us to test the effectiveness of the proposed algorithms in more complex situations compared to a single UAV.

Such a distribution of the PoIs, as shown in Figure

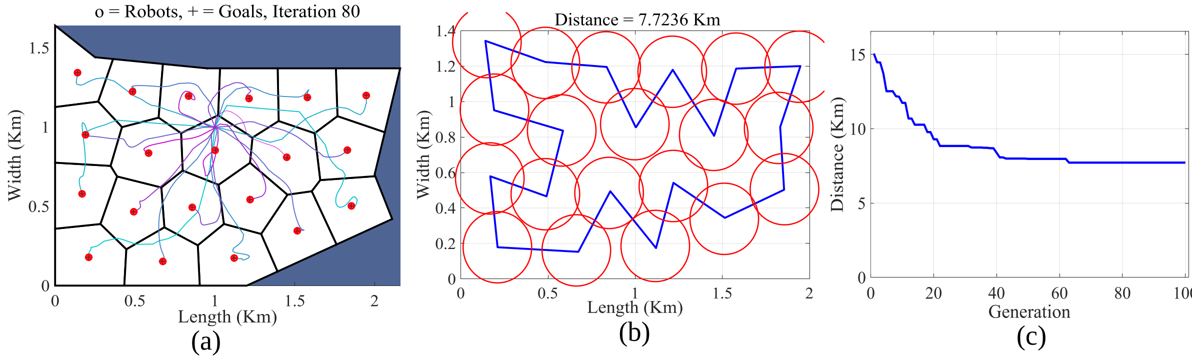

Figure 5: Area division and CPP (Scenario 1): (a) Area discretization, (b) Planned path and coverage results with circular sensor footprint, (c) TPL according to generation.

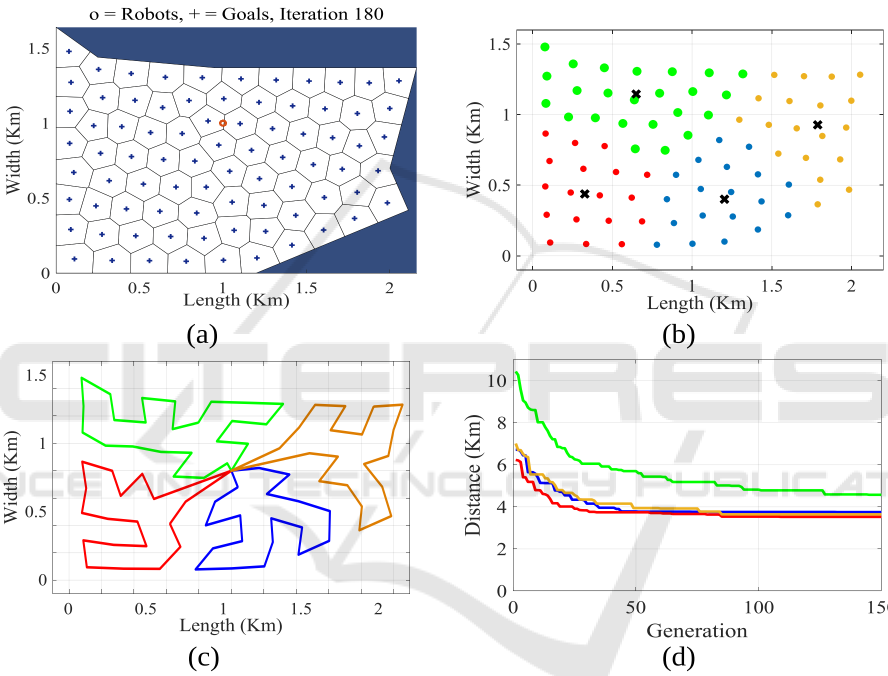

Figure 6: Area division and CPP (Scenario 2): (a) Area discretization with Meshing algorithm, (b) K-means clusters, (c) Planned paths solving TSP-GA, (d) TPLs according to generation.

6 (a), required 180 iterations. The result of grouping the PoIs for each UAV using the K-means algorithm is illustrated in Figure 6 (b), where the black crosses represent the centroids of each cluster. Figure 6 (c) presents the final planned paths for the UAVs, generated from the optimization step using TSP-GA. The TLPs according to generation during optimization are shown in Figure 6 (d). The results of this scenario lead us to summarize the following conclusions:

- ***C1**:* The Meshing algorithm requires more iterations as the number of PoIs increases, which is manageable with the number of PoIs.
- ***C2**:* A homogeneous distribution of clustered PoIs by the K-means algorithm has a beneficial effect on TPLs (Table 2). This clustering also results in isolated paths relative to each other, which helps to avoid collisions between UAVs flying at the same altitude.
- ***C3**:* The set of planned paths are optimal in terms of TPL (and/or energy consumption), feasible, and ensure passage through all PoIs, thereby achieving overall coverage of the RoI.

Moreover, from ***C1*** and ***C2***, the choice of the number of PoIs and the number of UAVs provides another degree of freedom to accommodate the flight capabilities of UAVs. Additionally, there is the possibility of implementing sensors of various shapes (circular, square, rectangular, etc.), which offers a generic solu-

Table 2: PoIs and TPLs.

Path	**Number of PoIs**	**TPL** (Km)
Green path ($\mathcal{R}_1$)	24	4.5744
Red path ($\mathcal{R}_2$)	18	3.5241
Blue path ($\mathcal{R}_3$)	19	3.7513
Brown path ($\mathcal{R}_4$)	19	3.6438

tion for a wide range of applications.

5 CONCLUSION

Our work addresses UAV group path planning for aerial detection applications (coverage). The primary objectives are the automation of setup for crossing points and planning optimized paths for UAVs. The global approach is structured into three steps: the generation of PoIs, the clustering of these PoIs, and the optimal connection of all these points to ensure comprehensive coverage of the studied map. Addresses the limitations of scenario-specific approaches by proposing a more flexible methodology that integrates diverse discretization techniques, sensor types, and path optimization strategies.

The precise spatial discretization using the Meshing algorithm ensures comprehensive coverage of the global RoI, while the K-means clustering method enables balanced task allocation, contributing to collision avoidance and optimized path planning. The final optimization phase formulates the problem as a TSP, solved using an enhanced GA with modifications that significantly accelerate convergence. These improvements lead to more efficient path planning, reduced energy consumption, and overall enhanced UAV performance, as demonstrated by the simulation results.

While the proposed method improves UAV path planning, several limitations remain. Scalability issues may arise with larger UAV fleets due to the computational cost of GA optimization. The approach also assumes a static environment, lacking adaptability to dynamic obstacles, and does not explicitly account for UAV constraints, communication limits, or collision avoidance. To overcome these limitations, future work will focus on adaptive clustering for improved task allocation and coordination, real-time obstacle avoidance using Rapidly-exploring Random Tree (RRT) and RRT*, and trajectory generation to refine UAV motion planning before real-world deployment.

REFERENCES

Azpúrua, H., Freitas, G. M., Macharet, D. G., and Campos, M. F. (2018). Multi-robot coverage path planning using hexagonal segmentation for geophysical surveys. *Robotica*, 36(8):1144–1166.

Balampanis, F., Maza, I., and Ollero, A. (2017). Coastal areas division and coverage with multiple uavs for remote sensing. *Sensors*, 17(4):808.

Bouras, A., Bouzid, Y., and Guiatni, M. (2019). Multi-uavs coverage path planning. In *International Conference on Electrical Engineering and Control Applications*, pages 23–36. Springer.

Bouras, A., Bouzid, Y., and Guiatni, M. (2022). Multi-uav coverage path planning for gas distribution map applications. *Unmanned Systems*, 10(03):289–306.

Cabreira, T. M., Brisolara, L. B., and Paulo R, F. J. (2019). Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1):4.

Chen, Y., Mou, Z., Lin, B., Zhang, T., and Gao, F. (2024). Complete coverage path planning for data collection with multiple uavs. In *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 01–06. IEEE.

Choset, H. (2001). Coverage for robotics–a survey of recent results. *Annals of mathematics and artificial intelligence*, 31:113–126.

Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal voronoi tessellations: Applications and algorithms. *SIAM review*, 41(4):637–676.

Gao, C., Kou, Y., Li, Z., Xu, A., Li, Y., and Chang, Y. (2018). Optimal multirobot coverage path planning: Ideal-shaped spanning tree. *Mathematical Problems in Engineering*, 2018(1):3436429.

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

Kumar, K. and Kumar, N. (2023). Region coverage-aware path planning for unmanned aerial vehicles: A systematic review. *Physical Communication*, 59:102073.

Pham, T. H., Bestaoui, Y., and Mammar, S. (2017). Aerial robot coverage path planning approach with concave obstacles in precision agriculture. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 43–48. IEEE.

Yu, Y. and Lee, S. (2024). Multi-uav coverage path assignment algorithm considering flight time and energy consumption. *IEEE Access*, 12:26150–26162.