PANTHEON

Community-Based Smart City Digital Twin Platform for Optimised DRM operations and Enhanced Community Disaster Resilience

D6.1

IN SITU COLLABORATIVE SWARM OF DRONES

The project has received funding from the European Union's Horizon Europe programme under Grant Agreement N°101074008.

DOCUMENT INFO

Deliverable Number	D6.1
Work Package Number and Title	WP6 - PANTHEON Remote sensing for multi-hazards and Data Delivery Scheme Implementation
Lead Beneficiary	13 - EPSILON
Due date of deliverable	30/04/2025 (M28)
Deliverable type[1]	DEM
Dissemination level[2]	PU
Author(s)	Lampis Papakostas (THL), Ilham Zerrouk (ENAC), Esther Salami (UPC)
Internal reviewer(s)	Fanourios Fakoukakis (FINT), Ana-Maria Dumitrescu (SIMAVI)
Version - Status	1.5

TASK ABSTRACT

This task focuses on analysing and evaluating different swarming algorithms to optimize the movement patterns of unmanned vehicles (UxVs). The selection and refinement of these swarming schemes will be carried out in close collaboration with end-users to ensure alignment with real-world operational needs. Also, software modules will be designed for the real-time orchestration of unmanned vehicle (UxV) swarms to autonomously control multiple UxVs, ensuring optimized coordination based on fused and raw real-time data.

[1] **Please indicate the type of the deliverable using one of the following codes**:
R = Document, report
DEM = Demonstrator, pilot, prototype, plan designs
DEC = Websites, patents filing, press & media actions, videos
DATA = data sets, microdata
DMP = Data Management Plan
ETHICS: Deliverables related to ethics issues.
OTHER: Software, technical diagram, algorithms, models, etc.

[2] **Please indicate the dissemination level using one of the following codes**:
PU = Public
SEN = Sensitive

REVIEW HISTORY

Version	**Date**	**Modifications**	**Editor(s)**
1.0	03/04/2025	First draft	Lampis Papakostas (THL)
1.1	24/04/2025	ENAC Input	Ilham Zerrouk (ENAC)
1.2	27/04/2025	UPC Input	Esther Salami (UPC)
1.3	28/04/2025	Merge inputs and finalize	Lampis Papakostas (THL)
1.4	29/04/2025	Update figures	Ilham Zerrouk (ENAC)
1.5	07/05/2025	Apply Reviewers Comments	Lampis Papakostas (THL)

DISCLAIMER

Funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The present document serves as supplementary material, offering in-depth technical details on the multi-Unmanned Autonomous Vehicles (UAVs) swarm collaboration strategies developed and implemented throughout the work in T6.1 “Swarming schemes for optimised collaboration of drones” & T6.2 “Resource controller for autonomous drones”.

It focuses on creating new methods for swarm navigation, energy-efficient path planning, and area coverage, making sure the drones can work together safely and effectively in real-world disaster situations. The solutions were used in different disaster scenarios like wildfires, earthquakes, and cyber-attacks. Feedback from first responders helped to adjust the methods to real needs. The results show that using UAV swarms can greatly improve the speed, safety, and quality of information gathered during emergencies, and can also support realistic training.

1. Introduction

UAV swarms are emerging as a transformative technology in disaster management, offering rapid, flexible, and scalable solutions for a range of critical tasks. By coordinating multiple UAVs to operate collaboratively, swarm systems can achieve extensive area coverage, resilient communication networks, and adaptive mission execution, even in complex and dynamic environments. In disaster scenarios such as earthquakes, floods, wildfires, and industrial accidents, UAV swarms can be deployed for search and rescue, damage assessment, situational awareness, and supply delivery. Their distributed nature enhances system robustness, reduces single points of failure, and enables real-time data collection and decision-making. The combination of advanced autonomy, decentralized control, and cooperative behaviours makes UAV swarms a promising and increasingly vital tool for improving the efficiency and effectiveness of disaster response operations.

Integrating UAV swarm technologies into PANTHEON Smart City Digital Twin (SCDT) offers a powerful approach to training first responders by creating highly realistic, dynamic simulations of disaster scenarios. By leveraging the coordinated actions of multiple UAVs, the digital twin can simulate large-scale emergencies with detailed real-time data on environmental conditions, structural damage, and population movement. This enables first responders to engage in immersive, high-fidelity training exercises that mirror the complexity and unpredictability of real-world events. UAV swarms contribute to enhancing situational awareness within the digital twin, allowing for the testing of different response strategies, resource allocations, and communication protocols under varied and evolving conditions. Thus, this integration can be a critical tool for improving preparedness, decision-making skills, and the overall resilience of urban emergency response systems.

In this work, UAVs are integrated into three disaster scenarios: an *earthquake* and a *wildfire,* both located in Athens, and a *cyber-attack* scenario in Vienna. These scenarios present diverse conditions in which a swarm of heterogeneous drones is deployed to perform critical assessment tasks. In the *earthquake* scenario, the drones assess damage to buildings and infrastructure. In the *wildfire* scenario, they monitor fire propagation and provide aerial imaging. In the *cyber-attack* scenario, the drones assess structural damage, identify casualties resulting from an explosion, and measure gas concentrations in the air caused by the ignition of hazardous products or structures.

The current document chapters can be summarized as follows: **Chapter 2** provides a literature review, examining current approaches and methodologies for swarm and multi-UAV systems, with a particular focus on their application to disaster management scenarios. **Chapter 3** offers a description of the swarming schemes, presenting the algorithmic foundations and technical implementations of the swarming strategies developed in this work. **Chapter 4** focuses on the alignment with Pantheon use cases, detailing the application and integration of UAV swarm technologies into the Pantheon system's operational scenarios. Finally, **Chapter 5** presents the conclusion, summarizing the main contributions of the work, discussing key insights, and outlining directions for future research and development.

2. LITERATURE REVIEW

The use of UAVs in disaster management has greatly enhanced the efficiency of emergency response, enabling rapid assessment, communication support, and supply delivery. This literature review highlights key research on UAV swarms in disaster scenarios, with a focus on critical challenges such as traversal of designated waypoints and effective area coverage. Efficient waypoint planning ensures UAVs reach vital locations—such as survivor sites or damaged infrastructure—while maximizing area coverage is essential for thorough situational awareness. Recent studies propose various optimization strategies to address these challenges under constraints like battery life, terrain, and communication limits.

Waheed et al. [1] propose a reinforcement learning-based approach to optimize UAV placement for covering critical nodes in emergency networks, focusing on maximizing capacity and minimizing information delay.

Malandrino et al. [2] propose an optimization framework for joint planning of multitask missions using a fleet of UAVs equipped with standardized accessories, enabling them to perform heterogeneous tasks such as surveillance, communication support, and parcel delivery in post-disaster scenarios. Their approach demonstrates that fully equipping UAVs enhances operational flexibility and efficiency, reducing the number of UAVs needed without compromising task quality.

Bailon-Ruiz et al. [3] present a trajectory planning approach for fixed-wing UAV fleets to monitor dynamic wildfires, utilizing realistic models of terrain, fire propagation, wind, and UAV dynamics. By tailoring a Variable Neighbourhood Search algorithm to these models, their method enables adaptive observation planning that update as new fire data becomes available, enhancing situational awareness for firefighting operation.

Malandrino et al. [4] propose an optimization framework for deploying UAVs to provide wireless coverage in disaster-stricken areas, focusing on maximizing user throughput while ensuring fairness across affected regions. Their simulation of a flooding scenario in San Francisco demonstrates the effectiveness of UAV-based networks in maintaining communication services when traditional infrastructure is compromised.

Maza and Ollero [5] present a cooperative search strategy for multiple heterogeneous UAVs, utilizing polygonal area decomposition and efficient coverage algorithms. Their approach divides the search area based on UAV capabilities and initial positions, assigning regions to each UAV for coverage using a zigzag pattern that minimizes turns.

This paper [6] introduces a multicircuit route planning method for UAVs to optimize terrain coverage, focusing on minimizing operational time while considering constraints like battery life and communication range. This approach enhances the efficiency of UAV deployments in disaster management by ensuring comprehensive area coverage with reduced mission durations.

In [7], the authors propose a cooperative multitask assignment based on area segmentation. They use a Genetic Algorithm (GA) that considers area mapping task requirements, UAV kinematics constraints, resource constraints (maximum endurance and sensor capabilities), and task number ceiling constraints; and optimizes the task sequence based on the objective function that considers UAVs coverage path planning characteristics. An improved double-chromosome encoding GA with a conflict-mediation mechanism is proposed to perform genetic operations on the population while meeting these constraints.

3. Swarming Schemes Description

Swarm Navigation

After conducting an exhausting literature review and determining the limitations of existing solutions, a swarm navigation framework that enables coordinated UAV movement between Points of Interest (POIs) was proposed to satisfy the needs of the project. This approach ensured efficient area coverage and real-time collision avoidance through inter-UAV communication.

The proposed system comprises four key components which are determined by the disaster mechanisms. First, the area of interest must be defined, and critical Points of Interest (POIs) must be identified and prioritized. Next, appropriate UAVs and sensors are selected to form a robust and efficient swarm formation. Finally, the communication framework governing data exchange between the UAV agents and the base station must be established to ensure seamless information flow.

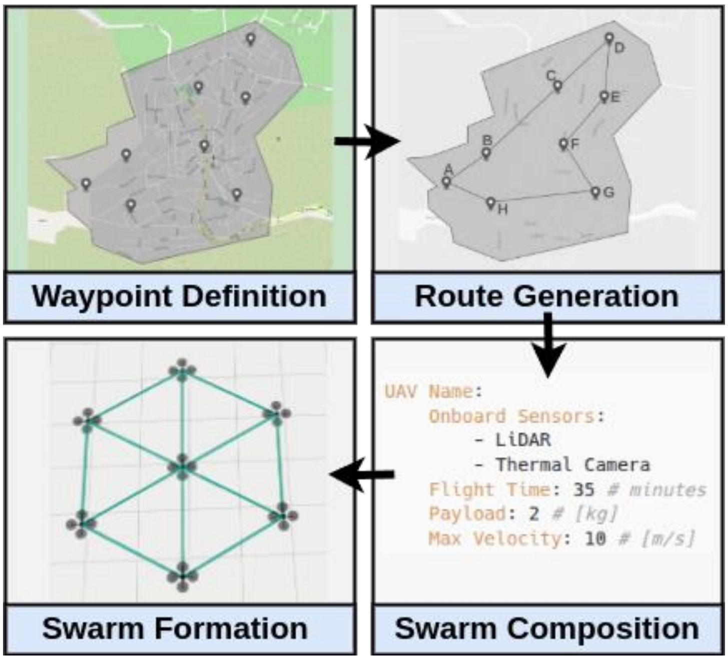

Figure 1: Swarm Navigation System Architecture

Certain POIs are prioritized based on their significance in assessing the impact and guiding emergency response. Additionally, the dynamic nature of a post-disaster environment imposes time constraints on the availability of this information. To compensate these factors, the proposed system accounts:

(a) the coordinates of POIs

(b) a priority-based prize value that quantifies the significance of the acquired information, and

(c) the opening and closing times of each POI, which define the time window when the surrounding area is deemed a safe flight zone, ensuring that all available information is collected while minimizing the risk to the UAV swarm.

In cases where visiting all POIs is infeasible, the algorithm prioritizes those that provide the most critical information, ensuring an optimal balance between data acquisition and operational safety.

The UAV swarm composition, including the number of agents and onboard specialized sensors (thermal camera, gas sensor, humidity sensor, LiDAR, etc), is determined using a capabilities database that catalogues UAV characteristics. Each UAV can carry different sensors, with flight duration and velocity dependent on its payload and its inherent specifications. The swarm configuration is selected as a subset of this database using an empirical rule that accounts for sensor heterogeneity, minimum flight duration, and minimum velocity, both constrained by the weakest agent. Once the UAV count is set, the swarm is organized into a spatial formation—structured as a polygonal shape in 2D or 3D space depending on the area of interest—to ensure maximum coverage with available sensors. A designated UAV at the geometric centre, chosen for its extended flight duration and high velocity with minimal payload, serves as a reference for trajectory generation and formation maintenance without carrying specialized sensors.

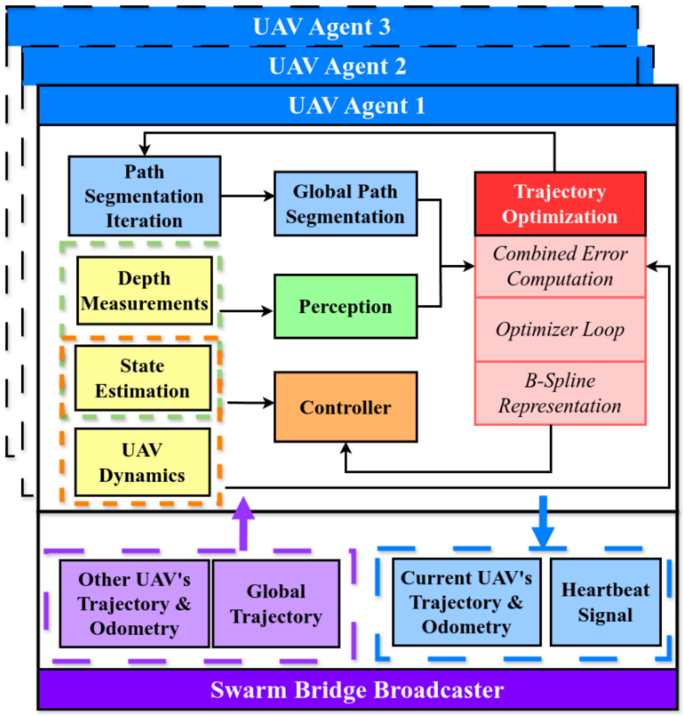

Figure 2: UAV Agent Software Architecture

The central UAV acts as a communication hub, maintaining strong signal links within the swarm to ensure reliable data transmission by minimizing the distance between agents. Implementing a communication bridge at central UAV, all agents can broadcast and transmit odometry, local trajectory, and heartbeat signals between them, each identified by a unique identify number. The ground station only transmits waypoints for global trajectory planning and an execution signal, with the central UAV serving as the reference.

To maintain swarm formation, a formation similarity metric has been adapted based on the representation of the swarm as an undirected graph. Each vertex i of the graph, represents a UAV with position vector $p_i = [x_i\,, y_i\,, z_i]$. Each edge e_{ij} connects UAV Agent i and UAV Agent j using the Euclidean distance between agents. A formation similarity metric has been adapted [8] (Eq.1) as

$$f = ||\,\hat{L} - \hat{L}_{des}||_F^2 = tr(\,(\,\hat{L} - \hat{L}_{des})^T(\,\hat{L} - \hat{L}_{des})) \quad (1)$$

where $tr\{\}$ denotes the trace of a matrix, $\hat{L}$ is the symmetric normalized Laplacian of the current formation, $\hat{L}_{des}$ the counterpart of the desired formation and F denotes the Frobenius norm. This metric is invariant to geometric transformations since the corresponding graph is weighted by the (absolute) Euclidean distance between UAVs and scaling invariance is achieved with the use of normalized graph Laplacian.

To achieve robust collision avoidance, the proposed methodology is based on the development of a local submap for each UAV agent in the swarm, focused on the UAV's immediate surroundings, significantly reducing computational overhead. This incrementally built Euclidean Signed Distance Field (ESDF) map [9] is used to compute the distance and gradient relative to nearby obstacles and is created by utilizing depth sensor data and GPS-based odometry. The swarm's collective perception enhances local mapping by covering a broader area, while the compact map size enables fast and accurate environmental representation, even in dynamic conditions [10].

Using distance and gradient information from the ESDF map, an optimization-based local trajectory replanning algorithm is deployed, and the essential terms are integrated into the cost function L (Eq. 2). The proposed cost function also includes agent-specific terms for formation maintenance and intra-UAV collision avoidance based on the swarm similarity metrics. The algorithm generates a smooth B-Spline trajectory across sequential 3D points, leveraging the local control properties of B-Splines for real-time replanning [11].

$$L_i = L_{c,i} + L_{ep,i} + L_{s,i} + L_{fs,i} + L_{fa,i} \qquad (2)$$

The overall cost function is a sum of individual agent terms, comprising: a collision term ($L_{c,i}$) to penalize paths that intersect with obstacles, an endpoint error term ($L_{e,i}$) to maintain replanned and initial trajectory consistency, a soft limit ($L_{s,i}$) that constrains velocity, acceleration, and jerk to allow the use of an unconstrained optimization algorithm, a swarm formation similarity term ($L_{fs,i}$) to preserve formation integrity, and a reciprocal avoidance term ($L_{fa,i}$) to prevent UAV-to-UAV collisions.

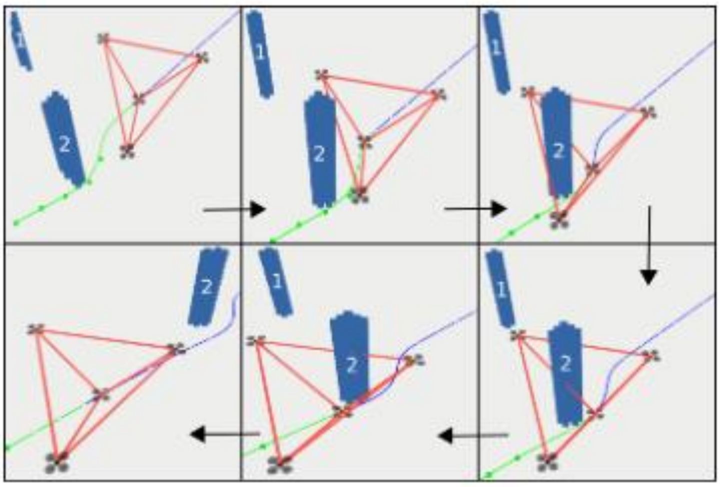

Figure 3: Obstacle Avoidance and Local Re-planning

Finally, to minimize travel costs a priority-based variant of the Traveling Salesman Problem (TSP) is implemented. In this approach, each POI is assigned a prize value, and the swarm must visit a subset of points to maximize profit under constraints. This method allows for skipping less critical nodes, balancing the trade-off between travel costs and the rewards associated with visited points. The TSP used in this work is specifically adapted to incorporate constraints related to infrastructure importance, the necessity of the required information, available flight time, and the capabilities of the onboard sensors. Furthermore, Time Windows (TW) are introduced, requiring each point to be visited within a specific time interval to ensure the feasibility of visits.

The proposed methodology was tested in simulation experiments with different swarm sizes and was validated using metrics for area coverage, formation maintenance and trajectory deviation. This work also led to a publication in the MED 2025 conference titled: "*Optimized Area Coverage in Disaster Response Utilizing Autonomous UAV Swarm Formations*" [12] accepted on 08/04/2025.

ENERGY AWARE COVERAGE PATH PLANNING

To minimize the transportation costs while maintaining efficient path planning, a Multi-UAV Coverage Path Planning (mCPP) approach was introduced [13]. An energy consumption estimation based on trajectory generation and the novel fast energy consumption estimation algorithms are described, both leveraging the pen-and-paper algorithm for calculating the optimal speed per travelled distance [14]. Each part of the path planning algorithm is described such as the area decomposition based on Boustrophedon Cellular Decomposition (BCD) [15], the novel problem conversion to Multiple Set Traveling Salesman Problem (MS-TSP) instance, and the MS-TSP solver extending the approach [16].

The proposed CPP algorithm is designed to plan paths for a specified number of UAVs as well as to minimize the number of paths (needed UAV flights) while the maximum path energy stays below a user-defined limit. A single planning step for a specified number of paths starts with a greedy calculation of N_{angles} best initial rotation angles for the input polygon fly-area. For each of those angles, an initial Area of Interest (AOI) is rotated by that angle and decomposed using BCD into a set of non-overlapping polygon. If the number of polygons is smaller than the user-defined limit ($Nmin \cdot Npaths$), polygons with the largest area are divided. For each of the resulting polygons, different back-and-forth coverage paths with distinct sweeping angles are generated.

Each of these paths is then represented by a node in a weighted graph with a weight equal to the estimated energy consumption of a UAV following that path. By grouping all the graph nodes corresponding to the same sub-polygon into a set, we can generate directed edges between each pair of nodes from different sets. The weight of such edges is equal to the energy needed to move from the last point of the source node path to the first point of the destination node path. Finally, initial and end nodes that represent UAVs' initial and end positions are added.

After the transformation to MS-TSP, the instance is solved on the created graph representation. The solver uses Greedy Randomized Adaptive Search Procedure (GRASP) [17] with Greedy Random Search Procedure (GRP) for initial solution generation followed by Tabu Search (TS) [18] for searching for a better solution. UAV paths are then recovered from the solution by replacing solution nodes with corresponding coverage paths and edges with a path connecting two adjacent paths.

The estimation of path energy consumption is based on the pen-and-paper algorithm [14]. Based on the UAV's physical parameters, it can estimate the optimal speed for maximizing the flight distance and power consumption during either flying with such speed or in hover conditions [13]. The optimal speed then minimizes the energy consumption of traveling along a straight segment with a fixed distance.

The algorithm decomposes the AOI by traversing points from left to right and creating vertical lines, with the rotation of the AOI affecting the decomposition result. The best initial rotation is estimated using the method in [18], where the area is decomposed for each boundary segment's rotation, and the decomposition is evaluated with the cost function

$$w = \sum_{i=1}^{m} y_{max,i} - y_{min,i} \qquad (3)$$

where m is the number of sub-polygons, and $y_{max,i}$ and $y_{min,i}$ are the uppermost and lowermost y-coordinates of the i-th segment. The best N_{angles} initial rotations are selected, and the path planning algorithm is run for each. After fixing the initial rotation, the AOI is decomposed again using Boundary Constrained Decomposition (BCD), with the condition $m \geq N_{min} \cdot N_{UAV}$, where N_{UAV} is the number of UAVs and N_{min} the minimum number of sub-polygons per UAV. If this condition isn't met, the largest sub-polygons are divided further.

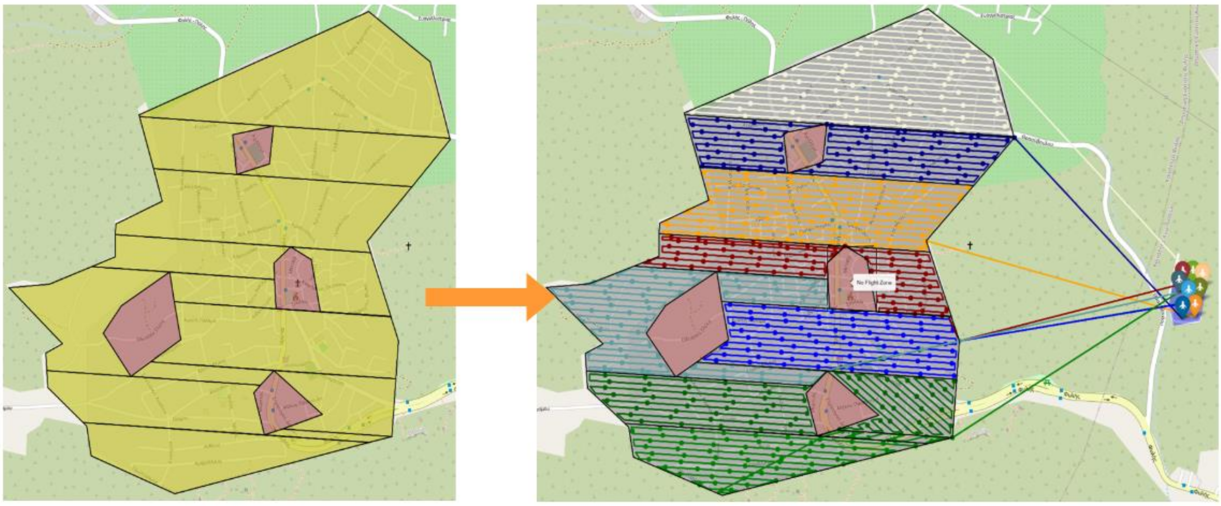

Figure 4: Energy Aware Area Decomposition and Path Generation

Having the AOI decomposed into sub-polygons, multiple sweeping coverage paths are generated for each of them. This algorithm step is based on a statistical analysis showing that the most energy-efficient paths usually have sweeping lines along the longest edge of the polygon. Therefore, for the i-th sub-polygon $4 \cdot min\,(N_{i,feas},\ N_e)$ sweeping patterns are generated along $4 \cdot min\,(N_{i,feas},\ N_e)$ longest edges in feasible sweeping directions. The $N_{i,feas}$ is the number of i-th feasible polygon sweeping edges, and the N_e is the maximum number of sweeping rotations in each sub-polygon.

When creating sweeping paths along the chosen edge, the two possible start points are located at each end of that edge at the distance $\frac{s}{2}$ from it, where s is the sweeping step. This leads to two corresponding end locations that can also serve as start locations by changing each path's direction, leading to four possible sweeping paths.

After having multiple sweeping coverage paths for sub-polygons, each is assigned one graph node, where the weight of the node is equal to the energy needed to fly along it. All of the nodes associated with same sub-polygon are grouped into the same set. Edges connecting each pair of nodes from different sets are added with the weight of an edge equal to the energy spent to travel from the last point of the sweeping pattern associated with the source node to the first point of the sweeping pattern associated with the destination node.

Considering that the number of nodes in the generated graph may be large, and that the algorithm should obtain a solution in a short time just before the flight, a meta-heuristic algorithm is utilized to quickly find possibly only a sub-optimal solution. Both the total energy consumption and the maximum energy spent by any of the UAVs are considered as a combination of cost functions for MS-TSP. The maximum energy consumption is used for initial cost comparison, and the average energy consumption is used in the case of two solutions having the same cost.

The Greedy Random Search Procedure (GRP) is used as the initial solution generator where the solution is represented as a list of paths with one path for every UAV. Each path is a list of graph nodes representing the sweeping patterns and their ordering. Adaptive Tabu Search (TS) is then used to improve the initial solution using four moves that search through the solution neighbourhood. The solution neighbourhood generation uses randomly one of the following four moves:

1) *Random shift*: randomly choose a node and move it into a random position among all paths.
2) *Best shift*: Randomly choose a node. Iteratively find the best position for it and move it there.
3) *Best swap*: Randomly choose a node. Iteratively find the best node to swap it with and swap them.
4) *Change direction*: Randomly choose a node and replace it with a random node from the same set.

After applying move procedures 1-3, the moved nodes are replaced with the best (minimizing the cost function) nodes from the same set.

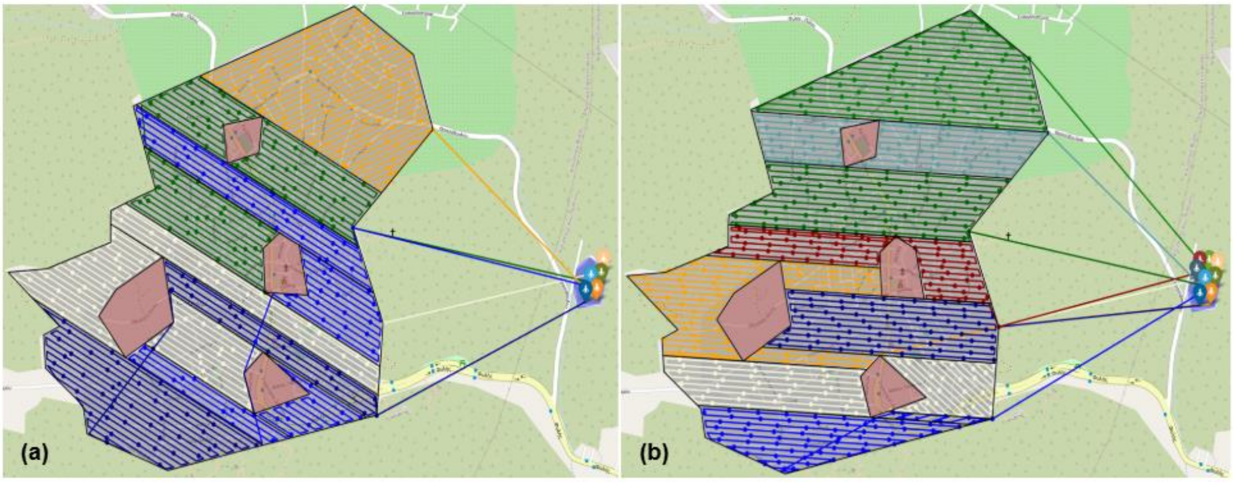

Figure 5: Algorithm Results for (a) 5 UAVs with 13 Convex Polygons and (b) 8 UAVs with 14 Convex Polygons for the same fly area

OPTIMIZED COVERAGE PATH PLANNING (CPP) LEVERAGING DRONE CAPABILITIES

The objective of our proposed method is to find a path for drones that cover the whole area of interest efficiently taking into consideration the drone capabilities. The algorithms that make up our method take as input the area of interest which will be surveyed. This area must be known and limited by a polygon.

Other areas can be defined in the mission like areas of no interest inside the polygon, and restricted areas due to the presence of other airspace users, such as evacuation helicopters or water bombers. The latter can be outside or intersecting with the main polygon. Indeed, these kinds of areas are considered in the proposed method.

To effectively cover the polygon of interest, we employ a two-step method: first, we decompose the area into sub-zones with defined proportions, and then we plan drone trajectories for non-convex regions.

The decomposition process utilizes the number of available drones and their maximum coverage area/surface to create partitions of suitable sizes and account for any existing holes within the area of interest. These partitions are uniquely non-convex.

Trajectory planning considers the non-convex nature of the partitions and aims to propose short paths from numerous possibilities. It ensures complete coverage of the area while avoiding obstacles, prohibited zones, and partitions assigned to other drones.

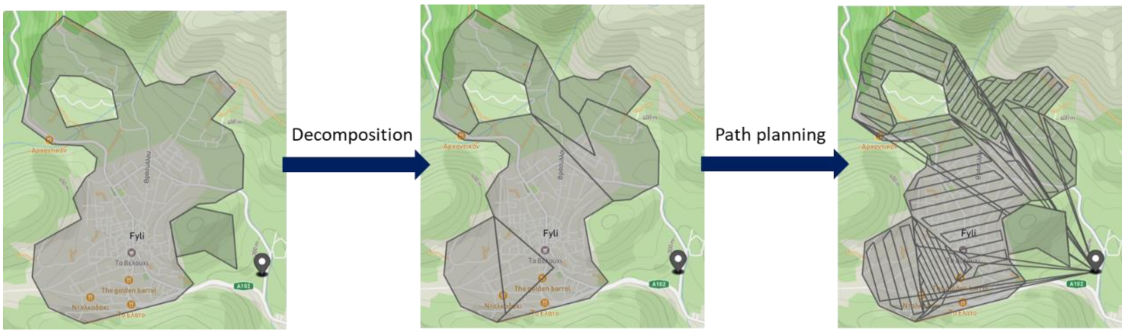

Figure 6: Area Decomposition and Path Planning Showcase

For the functioning of the algorithms, we consider the maximum flight altitude, the sensor characteristics (Field of View (FoV), resolution), the needs of mapping and detection tasks (overlapping, ground sampling distance, the target object size), the drone endurance, and the initial position from which it takes off (generally far from the target area). Thanks to this data, we calculate many parameters needed by both steps:

- The drone capability in terms of the area it can cover, which is used by the Area decomposition step to partition the area to subareas.
- The drone flight altitudes, drone sweep separation distance, and distance between camera shots are used for flight planification.
- Camera pan angle for each drone.

A decomposition algorithm is developed to address the problem of dividing the target area among multiple UAVs. It takes as input the perimeter of a polygon that defines the area of interest, which may include holes and no-fly zones, and a list of available drones, each with the surface area it can cover. The output is a list of perimeters corresponding to the resulting partitions.

The algorithm extends the bottom-up approach detailed in [19]. Its objective is to generate partitions that best match the UAVs' capabilities while maximizing compactness. The UAV capabilities are represented as area requirements, that is the proportion of the total area to be allocated to each drone. The metric of compactness for a polygon P, denoted as $C(P)$, is defined as the ratio of the square root of its area to its perimeter:

$$C(P) = \frac{\sqrt{Area(P)}}{Perimeter(P)} \qquad (4)$$

The original bottom-up algorithm begins by generating a triangular mesh using constrained Delaunay triangulation with additional Steiner points. This mesh is transformed into a hierarchical region-adjacency graph, where nodes represent simple polygons or subgraphs. The algorithm starts with the smallest area requirement and iteratively merges neighbouring chunks, prioritizing compactness, until the desired area is reached. Any remaining small chunks are merged, and a final adjustment phase ensures compliance with area constraints. This process is repeated for each remaining requirement using the existing graph until all the requirements are met. Finally, a post-processing step smooths boundary lines, while maintaining area constraints.

However, we observed that processing area requirements in ascending order can negatively impact the compactness of later partitions. To address this, the implemented version adopts a balanced splitting strategy. The list of requirements is initially divided into two subsets such that the difference in their total areas is minimized. The polygon is then split into the two corresponding subregions. This process is applied recursively: each subregion is further divided based on a balanced split of its associated requirements. This strategy results in more uniformly compact partitions.

In addition, the initial triangulation is performed only on the border points of the polygon. We observed that this significantly reduces computation time, without compromising the quality of the final results.

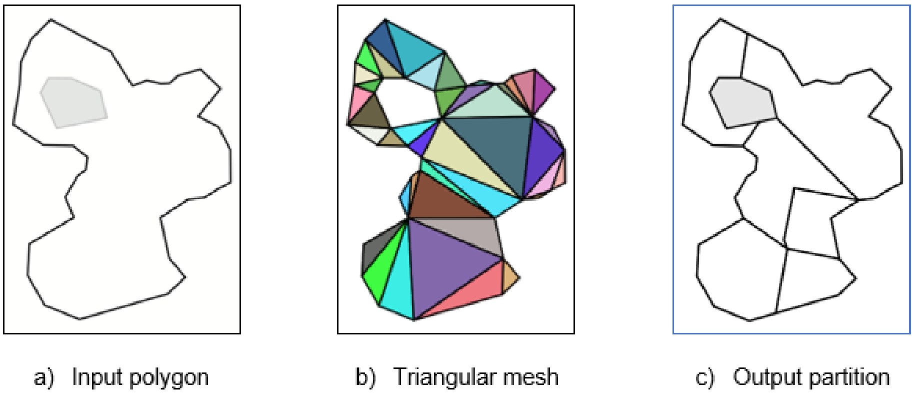

Figure 7: Decomposition of a polygon with one hole in 6 partitions with area requirements 18%, 14%,9%, 18%, 8%, and 33%.

The bottom-up algorithm for compact workspace decomposition [19] was compared against two approaches: an improved version of the Hert and Lumelsky algorithm for non-convex polygon decomposition (referred to as IHL) [20] and the DARP algorithm [21]. DARP, notable for providing an open-source implementation, is based on cellular decomposition and requires representing polygons as a collection of grid cells. Since DARP also requires UAV initial positions without a provided selection method, we randomly generated these positions within the polygon and chose the best result based on compactness. The evaluation criteria included normalized compactness of the partitions, the number of tracks in the generated flight plans, and algorithm execution time. All methods were tested on the same set of 100 randomly generated polygons with 4 to 50 vertices, divided among 2 to 10 UAVs with equal area assignments.

Maximizing the compactness of partitions is a standard quality measure in polygon decomposition. To ensure fair comparison across polygons of different sizes but similar shapes, compactness was normalized to a range from 0 to 1, with 1 representing maximum compactness. This normalization simplifies result interpretation, with higher values indicating better performance. Figure 8 shows that the Bottom-up algorithm consistently outperforms both IHL and DARP in terms of compactness, and it also scales better with the number of partitions compared to IHL, highlighting its efficiency and robustness for multi-UAV workspace division.

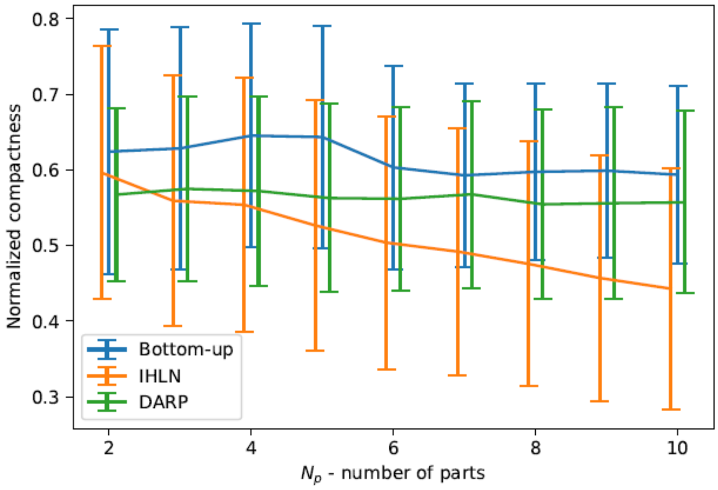

Figure 8: Normalized compactness vs number of parts for the bottom-up, IHL and DARP algorithms

For each partition created by the decomposition step, the planning algorithm transforms the partition into a grid of points, spaced by fixed horizontal and vertical distances. These distances are determined by the camera's field of view, the flight altitude, and the overlap between consecutive photo shots. The points indicate the paths the drones will follow and specifically mark the locations where the onboard camera captures photos. A grid is generated for a specific orientation and the algorithm calculates for each partition different orientations. Each point in the grid is associated with a Boolean value indicating whether the drone can pass through that point a "pass point" or not. A "no-pass point" is one that lies outside the polygon or within a fixed obstacle inside it. For each orientation, some extra points are calculated for small zones that are not covered by shots and associated with their nearest point in the corresponding grid. These points will be added to the trajectory during the next stage "Path calculation". A grid and its extra points help the algorithm to systematically plan paths, ensuring that the entire area is covered efficiently.

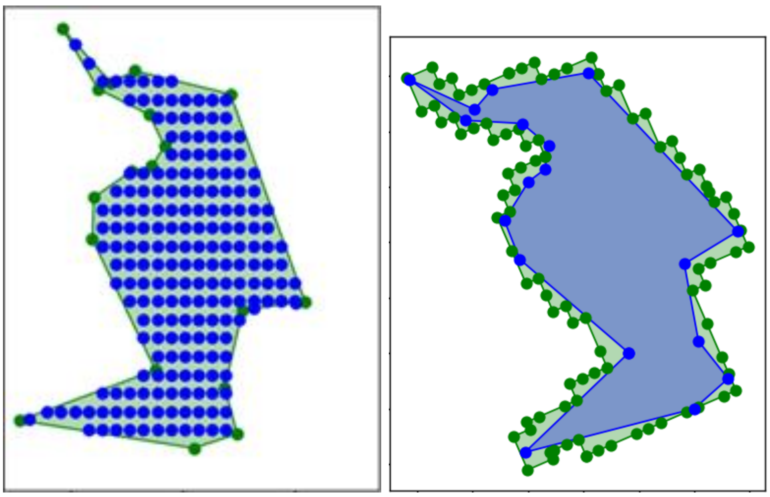

Figure 9: Point grid generated for a partition with the corresponding coverage

After constructing the grids for various orientations and their corresponding extra points, the next stage is to create the trajectories. A trajectory starts at the drone's initial position and continues to the partition entrance. If any no-fly-zone is found between these two points, the algorithm generates a path that avoids that zone by flying over its outer contour following the shortest side of this area. The entry point for the partition is situated on the first line of the grid. The selection of this point, whether at the beginning or end of the line, is deferred until the final trajectory is calculated, ensuring the shortest path. The trajectory is constructed using a back-and-forth pattern. When the algorithm encounters one or more no-pass points, it makes a detour to reach the next "pass point". Once all pass points and extra points have been visited, the point corresponding to the drone's initial position is added as the end point. To reach the end point, the external no-fly zone, if it exists, is avoided in the same way as before.

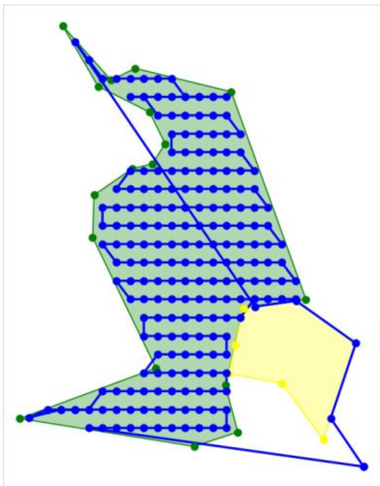

Figure 10 Generated trajectory for a partition avoiding a no-fly zone

This approach helps avoid fixed obstacles and no fly zones but does not prevent flying over neighbouring partitions when accessing its own zone or when moving from one grid point to another that crosses segments of an adjacent partition. The solution for the first problem is to use an altitude different from those of the scan when trying to reach its partition. For the second problem, the algorithm implements a mechanism that calculates an alternative route when the initial route crosses segments of neighbouring zones. By precisely following the outline of the prohibited area, the drone can continue its flight without violating any restrictions, ensuring both safety and efficiency.

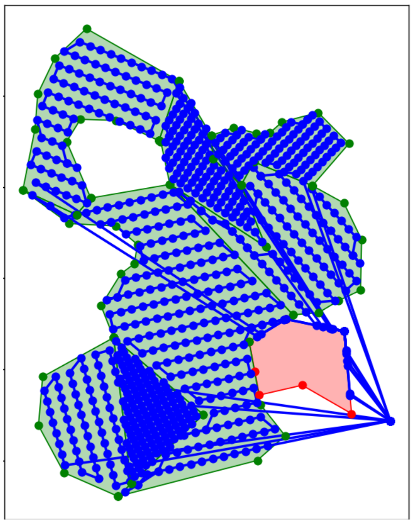

Figure 11 Final drone trajectories for scanning the entire area

To compare our path planning algorithm with existing methods, Fields2Cover [22], an open-source CPP library originally developed for agricultural vehicles (with adaptations made for UAV applications) is employed. Fields2Cover offers a comprehensive framework for generating coverage paths, developing new techniques, and benchmarking state-of-the-art algorithms. It supports path generation for non-convex areas with obstacles, utilizing various sweep patterns and turn models.

For the comparison, a polygon including an internal hole, is used as a test environment. The same UAV model was employed across all experiments, using the boustrophedon pattern for sweeps, with simple straight-line segments connecting the ends and starts of sweep lines. Notably, some differences exist between the methods compared. One key difference concerns the starting point of the sweep lines: in all methods except ours, each sweep line starts at a distance equal to half the separation distance between two consecutive shots. In contrast, in our algorithm, this distance is variable and can range between zero and the full shot separation distance. Additionally, because boustrophedon decomposition splits the area into multiple cells, UAVs following such paths may temporarily exit the main coverage area when transitioning between adjacent cells.

Metrics	**Our proposed CPP**	**Brute Force**	**Single orientation (π/2)**	**Boustrophedon decomposition + Brute Force**
Flight time (s)	2721	3098	3020	3363
Flight distance (m)	40820	46482	45309	50445
Number of turns	--	54	56	58

Table 1: Comparison of the path planning methods

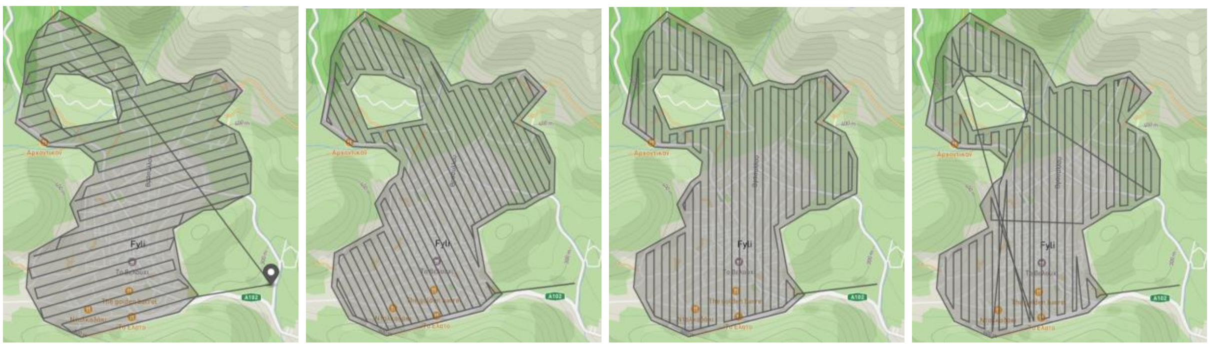

Figure 12 Compared path planning methods - from left to right: Our method, Brute force, Single orientation, Boustrophedon decomposition + Brute force

4. PANTHEON USE CASES ALIGNMENT

Based on the end-user requirements of the PANTHEON platform and the feedback provided by first responders during the defined use case scenarios, the developed methods are carefully modified or, in some cases, rejected to better align with the practical needs and constraints of real-world disaster response. First responders' input is critical in ensuring that the algorithms and swarm strategies are not only technically sound but also operationally viable in high-pressure, dynamic environments. For instance, factors such as real-time decision-making, ease of use, adaptability to various disaster scenarios, and the ability to handle large-scale emergencies are incorporated into the refinement process. In Table 2, the mapping of UAV Swarm methods along the Use Cases will be displayed.

Method	Use Case
Swarm Navigation	---------
Energy Aware Path Planning	*ATTICA WILDFIRE*
Area Decomposition and Path Planning	*ATTICA EARTHQUAKE, VIENNA CYBER-ATTACK*

Table 2: Swarming Methods and Use Case Alignment

Since the current deliverable is a demonstrator, the implementations, architectures, and UAV usage within the PANTHEON project can be explored through the following links.

SWARM NAVIGATION DEMO

Swarm Navigation algorithm implementation can be described in PANTHEON_D6_1_SWARM_NAVIGATION on PANTHEON Zenodo.

ATTICA WILDFIRE SCENARIO DEMO

The Attica wildfire Scenario can be described in PANTHEON_D6_1_UAV_WILDFIRE_MONITORING video on PANTHEON Zenodo.

ATTICA EARTHQUAKE AND VIENNA CYBER-ATTACK SCENARIO DEMOS

The Attica earthquake and the Vienna cyber-attack Scenarios can be described in PANTHEON_D6_1_UAV_EARTHQUAKE_CYBER_ATTACK video on PANTHEON Zenodo.

5. Conclusion

This deliverable has presented the design, implementation, and evaluation of collaborative swarming schemes for UAVs to enhance disaster management operations within the PANTHEON platform. Through an in-depth exploration of swarm navigation, energy-aware coverage path planning, and advanced area decomposition techniques, the work demonstrates significant advancements in operational efficiency, adaptability, and resilience. The alignment of these methods with real-world disaster scenarios, shaped by end-user feedback, highlights their practical relevance and readiness for deployment.

6. LIST OF ABBREVIATIONS

Abbreviation	Meaning
UAV	Unmanned Autonomous Vehicles
SCDT	Smart City Digital Twin
GA	Genetic Algorithm
POI	Points of Interest
ESDF	Euclidean Signed Distance Field
TSP	Traveling Salesman Problem
TW	Time Windows
CPP	Coverage Path Planning
BCD	Boustrophedon Cellular Decomposition
MSTSP	Multiple Set Traveling Salesman Problem
AOI	Area of Interest
GRASP	Greedy Randomized Adaptive Search Procedure
GRP	Greedy Random search Procedure
TS	Tabu Search
FOV	Field of View
IHL	improved Hert – Lumelsky Algorithm

7. References

[1] Waheed, Maham & Ahmad, Rizwan & Ahmed, Waqas & Alam, Muhammad & Magarini, Maurizio. (2023). On Coverage of Critical Nodes in UAV-Assisted Emergency Networks. Sensors. 23. 1586. 10.3390/s23031586.

[2] Francesco Malandrino, Cristina Rottondi, Carla-Fabiana Chiasserini, Andrea Bianco, & Ioannis Stavrakakis (2019). Multiservice UAVs for Emergency Tasks in Post-disaster Scenarios. *CoRR, abs/1905.05453.*

[3] R. Bailon-Ruiz, S. Lacroix and A. Bit-Monnot, "Planning to Monitor Wildfires with a Fleet of UAVs," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018, pp. 4729-4734, doi: 10.1109/IROS.2018.8593859.

[4] Malandrino, F., Chiasserini, C.F., Casetti, C., Chiaraviglio, L., & Senacheribbe, A. (2019). Planning UAV activities for efficient user coverage in disaster areas. *Ad Hoc Networks, 89, 177–185.*

[5] Maza, I., Ollero, A. (2007). Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In: Alami, R., Chatila, R., Asama, H. (eds) Distributed Autonomous Robotic Systems 6. Springer, Tokyo. https://doi.org/10.1007/978-4-431-35873-2_22

[6] Y. Wu, H. Yin, X. Chen and M. Zhang, "Multicircuit Route Planning for UAVs Performing the Terrain Coverage Task," in *IEEE Internet of Things Journal*, vol. 11, no. 13, pp. 23765-23779, 1 July1, 2024, doi: 10.1109/JIOT.2024.3386125.

[7] Jiaxuan Li, Xuerong Yang, Yajun Yang, Xianglin Liu, Cooperative mapping task assignment of heterogeneous multi-UAV using an improved genetic algorithm, Knowledge-Based Systems, Volume 296, 2024, 111830, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2024.111830.

[8] L. Quan et al., “Distributed swarm trajectory optimization for formation flight in dense environments,” in 2022 IEEE Int. Conf. Robot. Autom. (ICRA), 2022, pp. 4979–498

[9] A. Geladaris et al., “Real-time local map generation and collision-free trajectory planning for autonomous vehicles in dynamic environments,” in 2023 Int. Conf. On Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), 2023, pp. 1–6

[10] L. Han et al., “FIESTA: Fast incremental Euclidean distance fields for online motion planning of aerial robots,” in 2019 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2019, pp. 4423–4430

[11] V. Usenko et al., “Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer,” in 2017 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2017, pp. 215–222.

[12] Papakostas, L., Geladaris, A., Mastrogeorgiou, A., Sharples, J., Hattenberger, G., Chatzakos, P., Polygerinos, P., “Optimized Area Coverage in Disaster Response Utilizing Autonomous UAV Swarm Formations,” 33rd Mediterranean Conf. on Control & Automation (MED ‘25), June 10-13, Tangier, Morocco.

[13] Datsko, D., Nekovar, F., Penicka, R., & Saska, M. (2024). Energy-Aware Multi-UAV Coverage Mission Planning With Optimal Speed of Flight. *IEEE Robotics and Automation Letters, 9(3), 2893–2900.*

[14] 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.

[15] H. Choset and P. Pignon, “Coverage path planning: The boustrophedon cellular decomposition,” in Field and Service Robotics, A. Zelinsky, Ed. London: Springer London, 1998, pp. 203–20

[16] F. Nekov´aˇr, 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

[17] O. Guemri, A. Bekrar, B. Beldjilali et al., “Grasp-based heuristic algorithm for the multi-product multi-vehicle inventory routing problem,” 4OR, vol. 14, no. 4, pp. 377–404, 2016

[18] Paulo M. França, Nelida M. Sosa, Vitoria Pureza, An adaptive tabu search algorithm for the capacitated clustering problem, International Transactions in Operational Research, Volume 6, Issue 6,1999, Pages 665-678, ISSN 0969-6016, https://doi.org/10.1016/S0969-6016(99)00017-9.

[19] Skorobogatov, G., Calvo, T., Barrado, C., & Salamí, E. (2025). Compact workspace decomposition based on a bottom-up approach. *IEEE Access*.

[20] Hert, S., & Lumelsky, V. (1998). Polygon area decomposition for multiple-robot workspace division. *International Journal of Computational Geometry & Applications*, *8*(04), 437-466.

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

[22] Mier, G., Valente, J., & Bruin, S. (2023). Fields2Cover: An Open-Source Coverage Path Planning Library for Unmanned Agricultural Vehicles. *IEEE Robotics and Automation Letters, 8(4), 2166-2172.*