BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 73(5), 2025, Article number: e154731
DOI: 10.24425/bpasts.2025.154731

Multimodal drone swarm for search and rescue mission

Antoni KOPYT* **and Anna CZAPLIŃSKA**

Warsaw University of Technology, Institute of Aeronautics and Applied Mechanics, Poland

Abstract. A drone swarm is a large group of cooperating unmanned aerial vehicles that exhibit some form of autonomy. This structure allows for an increase in the complexity of performed tasks while limiting the mental load put on the operator. One field where drone swarms can prove especially useful is search and rescue – they can reduce the mission time and improve personnel safety. This paper presents an offline mission planning module composed of a set of mission scenarios designed to aid rescuers in a disaster area: internet provision, area search, and patrol. The internet connection is provided by a hexagonal mesh of drones spanning the area. A method for continuous drone replacement and charging is presented. The area search scenarios are based on a cooperative subarea search by groups of drones using a lawnmower pattern. The patrol scenarios feature continuous area patrol by an unstructured flock of drones and formation boundary patrol (circle formation and V-shape). The developed solutions are an easy-to-implement base for multipurpose search and rescue drone swarm solutions. Their functionality can be expanded by the ground control station as desired.

Keywords: aerospace control; autonomous aerial vehicles; drones; path planning; swarm robotics; search and rescue.

1. INTRODUCTION

A robot swarm can be defined as "a group of three or more robots that perform tasks cooperatively while receiving limited or no control from human operators" [1]. This technology proved popular in the aerospace sector, namely, unmanned aerial vehicles (UAVs). Applying swarm control to large numbers of easy-to-produce, small drones combines the advantages of both components. Swarm operation allows drones to take on complex tasks and enhances their autonomy, thus unburdening pilots and broadening the range of applications. Using drones means that the units are small, able to operate in various environments, and are largely not constrained by terrain. This makes drone swarms an asset in many applications – mapping, farming, defense, search and rescue, and entertainment. Another important aspect of swarm mission planning is efficient and optimized routing and path planning. The use of evolutionary algorithms for this purpose is presented in [2]. Particle swarm optimization (PSO) was used in [3,4]. A comparison between genetic algorithms and PSO was discussed in [5]. A convex optimization approach was discussed in [6]. Drone swarms are a well-researched subject in terms of formation flight [7], network architecture [8–10], collision avoidance [11], and sensor integration [12]. Dedicated simulation software was created to facilitate platform development [13]. Direct and indirect control methods of swarm control were analyzed [14]. As the principles of drone swarm operation were established, the focus shifted to more specific applications. From being a minimally researched subject in 2019 [1], the field of UAV applications in search and rescue has expanded significantly, with surveys published on the topic [15, 16]. The former breaks down the subject by SAR mission stages (analyzing the disaster area, localizing targets, swarm navigation, and collision avoidance), and the latter categorizes research papers regarding the environment, equipment, sensors, optimization methods, and uncertainty. The core of search and rescue operations is fast target localization. This is reflected by a wide variety of approaches to this task [17, 18]. For drone swarms specifically, bio-mimetic solutions based on stigmergy and flocking have been popular [19–21]. A pigeon-inspired mission planning method for search-attack missions was developed [22]. A variety of online control algorithms were proposed, such as the mixed-integer linear programming [23], the moving peak drone search problem [24], and the continuous factored coordinated Monte Carlo tree search algorithm [25]. In terms of offline algorithms, the topic of coverage path planning is well-researched [26, 27]. However, the majority of existing solutions are either generic or developed for single-drone operations [28–33]. An offline path planning solution was developed, where the area is partitioned for a desired number of UAVs, and each subarea is searched by a single drone [34]. It should be noted, however, that target localization is not the only task performed in search and rescue operations, and once the swarm is deployed in the disaster area, it could perform other functions as well and become a useful tool. One such assignment would be providing network coverage over an area. A method for deploying a mobile sensor network was presented in [35]. Another task would be surveillance/patrol, which is a field where UAV applications are more common, for example: drone systems were developed for wildfire detection [36, 37], a platform for autonomous patrol with a manual override option was created [38], and a border patrol

*e-mail: antoni.kopyt@pw.edu.pl

Manuscript submitted 2025-03-10, revised 2025-05-07, initially accepted for publication 2025-05-28, published in August 2025.

solution for a group of drones was proposed. However, all the abovementioned are isolated solutions. The multi-functionality of the drone swarm is only possible if the platform for managing SAR swarm operations includes a variety of missions. To the best of the authors' knowledge, no such platform has been developed so far. This paper aims to design a set of mission scenarios for a drone swarm, crafted to aid personnel during search and rescue operations over a disaster area. Three mission categories were developed: internet provision, area search, and patrolling. Furthermore, the scope of this paper was limited to offline mission planning. The main advantages of this approach are easy implementation, few onboard hardware requirements, and easy supervision. Since the mission planning stage takes place on the ground control station, onboard computational power can be fully allocated for swarm management, i.e., collision avoidance, network connectivity, sensor data analysis, etc. Furthermore, the operator can verify drone routes during the mission planning stage and, later, monitor the progress of the mission. The following research was the first phase of the project called "Starling". The project aimed to develop an algorithm that would allow a set of swarm drones to fly and perform a predefined set of missions. This approach significantly differed when compared to drone shows, which involve no logic and cooperation between the units, and they are just a set of individuals flying in parallel. The presented approach is the first in the development and implementation of algorithms for drone swarms.

2. MATERIALS AND METHODS

The mission scenarios were designed for a swarm of several dozen drones developed as a part of the same research project. The swarm is comprised of "leaders" and "starlings". The leader drones can carry a payload and various sensors, whereas the starling drones are followers that function as relays for providing the internet connection. The mission scenarios are partially optimized in terms of swarm network architecture since they comply with the requirements of the internet provision module, but no constraints are placed on the drone-GCS connection (at the time when the scenarios were developed, no such requirements were known). It is assumed that the drones operate within the GCS connection range, and a continuous data exchange between the units is possible. The main goal for the mission planning module is to be versatile and require minimal user input. The mission scenarios must be highly automated and parameterized regarding drone specifications.

2.1. Internet provision

The first mission scenario aims to provide an internet connection by spreading a hexagonal mesh of drones over a specified area. It is inspired by the distribution of relay towers [39] and the node placement for ad-hoc networks for optimal planar area coverage [40]. The main constraint imposed by the swarm internet provision module was that the distance between every two neighboring drones must be equal to d. The structure of the mesh is shown in Fig. 1. To comply with the constraint, the

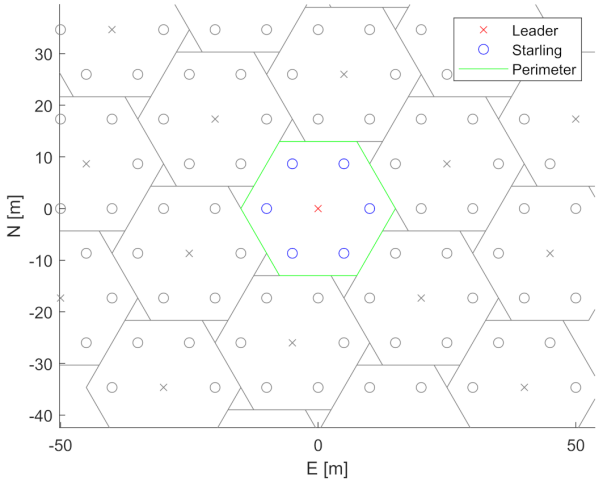


Fig. 1. The structure of the hexagonal mesh for internet provision (orientation $\alpha = 0°$)

drones are placed in the vertices of a grid of equilateral triangles [40]. Each cell is comprised of one leader (in the center) and six starlings (spaced radially). It is assumed that the distortion of the mesh caused by differences in flight altitude of drones is negligible. As such, the mesh calculation can be simplified to two dimensions. The coordinates of the starlings regarding the leader are defined in (1)

$$\begin{cases} N_i = N_l + d\sin\left(60°i + \alpha\right), \\ E_i = E_l + d\cos\left(60°i + \alpha\right), \end{cases} \tag{1}$$

where E_l and N_l are the leader coordinates, i is the starling ordinal number, d is the distance between neighboring drones, and α is the orientation of the cell (measured from the flat-top position). All mesh cells have the same orientation. The coordinates of neighboring cell leaders (N_n, E_n) regarding the specific leader are defined as presented in (2)

$$\begin{cases} N_n = N_l + d\sqrt{7}\sin\left(60°i + \alpha + \tan^{-1}\left(\dfrac{\sqrt{3}}{5}\right)\right), \\ E_n = E_l + d\sqrt{7}\cos\left(60°i + \alpha + \tan^{-1}\left(\dfrac{\sqrt{3}}{5}\right)\right). \end{cases} \tag{2}$$

There is a set of functions that covers the subject of the simulation research part, divided into 12 main sections. The structure of pseudocode and description of algorithms are presented in the Appendices (Algorithms 1–12), where the basic logic of operation is presented. The algorithms consist of: BSF mesh generation, Cell append condition set, generating the cells, connecting the ground control station to the mesh, automated replacement path, area patrol, circular formation 1, 2, and V-shaped formation 1 and 2, and finally boundary patrol. All those algorithms are part of a simulation. A breadth-first search-based (BSF) algorithm was developed for grid generation (Algorithm 1). Starting from a specified location (the base of operations), the cells are appended iteratively within the area of operations (AOO). This algorithm is presented in more detail in [41]. Three cells appending the

condition options were implemented (Algorithm 2). Starlings that do not contribute to area coverage can be removed from the grid to further reduce the number of required drones. Redundancy occurs when the unit is further than $d/2$ from the AOO boundary. Algorithm 3 presents the cell generation procedure with the redundant starling removal. Since the mission scenarios are to be used in disaster areas, it should be assumed that the base of operations is set up outside the AOO. This entails the necessity to connect the base of operations to the AOO via the mesh to maintain network connectivity. This is realized by appending a rectangular path to the AOO (Algorithm 4). This generates additional connections made by the drones, so the connection is led from the ground control station to the AOO. It must connect the base of operations to the AOO via the shortest possible line segment. The width of the path is user-defined to accommodate various circumstances – a wider path offers a more stable connection, but a narrower path uses fewer drones. The orientation of cells is chosen exhaustively by iterating through $\alpha \in (0°, 60°)$ with a fixed interval and choosing the best orientation by the following criteria:

1. The best area coverage
2. The fewest leader drones
3. A mixed condition (a weighted mean of the two former conditions, where w_1 is the area coverage weight and w_2 is the leader number weight)

The scenario handles a drone limit, which can be enforced either within the scenario (the BFS stops when there are either no leaders or no starlings available) or by the ground control station (a full mesh is generated, but the bottom-most entries are discarded to comply with the drone limit). Algorithm 5 summarizes the full mesh generation procedure and sets the proper positions for the drones to generate the mesh. Since the anticipated duration of the internet provision mission exceeds the typical battery capacity of the drone, a method for continuous drone replacement was developed. A Hamiltonian path (or paths) is found for the leaders, which is the replacement path. For simplification purposes, it is assumed that the removal of redundant starlings does not occur. Full cells move along the leader replacement path. If the redundant starling removal were in place, a separate replacement path for starlings could be calculated to accommodate partial cells (with fewer than six starlings). The cells move synchronously by one position along the replacement path with a fixed time interval. The hover time over each position should be longer than the traverse time between positions. Simultaneously, the drones must complete the whole replacement path within a single battery charge. Therefore, automatic mesh division and generation of multiple paths were implemented. The maximum length of the replacement path is defined as in (3)

$$n_{dp} = \varepsilon \frac{t_x v - l_{c2}}{d}, \tag{3}$$

where n_{dp} is the number of cells in the replacement path, t_x is the single charge flight duration, v is the average flight velocity, l_{c2} is the doubled distance between the base of operations and the mesh centroid (to account for a potential commute between the charging station and the path endpoints), and ε is a corrective coefficient. Based on this relation, the required number of replacement paths is calculated in (4)

$$n_{rp} = \left\lceil \frac{N}{n_{dp}} \right\rceil, \tag{4}$$

where N is the total number of cells in the mesh. The mesh is then divided into n_{rp} submeshes using the k-means algorithm. For each submesh, a Hamiltonian path is found with endpoints that are closest to the base of operations. The graph for which the Hamiltonian path is found is defined as follows: the leaders are nodes, and the neighboring leaders are connected by edges. If a leader only has one neighbor, additional edges are added connecting the leader to its neighbor's neighbors. Only the neighbors that are on the edge of the mesh (have fewer than six neighbors) are added. By limiting the number of edges added, the distortion of the replacement path shape is minimized, while increasing the probability of finding the Hamiltonian path. To maximize the mesh connectivity as it is rolled out and rolled in, these maneuvers should be performed along the replacement paths. The automatic generation of replacement paths is shown in Algorithm 6, which finds the best path using Hamiltonian.

2.2. Area search

The area search scenarios deal with the key part of search and rescue operations – target localization. The developed scenarios were described separately in detail in [26]. However, they will be summarized here for clarity. The area search scenarios are based on the cooperative parallel track (lawnmower) search of subareas by groups of drones. In the first scenario, the AOO is divided into subareas of equal size using Voronoi tessellation. The subareas are searched simultaneously by groups of drones, whose size is proportional to subarea priority. The lawnmower pattern is oriented as per the rotating calipers algorithm. This method significantly reduces the search mission time. In the second scenario, a group of drones searches the subareas of equal size consecutively, while another group of drones patrols the boundary between the searched and unsearched areas. As before, the subareas are generated via the Voronoi tessellation, and the search pattern orientation is based on the rotating calipers algorithm. This solution improves the precision of the search by accounting for targets leaving the AOO. The third scenario is an application of the simultaneous subarea search to a mountainous terrain, inspired by the contour search pattern. The AOO is divided into subareas based on the shape of the mountain range. The size of search groups is proportional to the subarea priority and the subarea size. The parallel track pattern is perpendicular to the mean subarea altitude gradient.

2.3. Patrolling

The patrolling scenarios provide basic surveillance functionalities – an area patrol and a boundary patrol. The area patrol scenario draws upon two major sources; both designed for forest fire detection. The first source [32] uses fixed-wing and rotary-wing UAVs to patrol the AOO from a high altitude. The fixed-wing drone patrols the AOO continuously and, if a fire is

spotted, the rotary-wing drone is dispatched to the point of interest. Upon closer inspection, the fire alarm is either confirmed or canceled. This method decreases the number of false-positive alarms. The second source [36] detects and predicts forest fires using a fleet of drones. A digital map of the AOO is built using satellite data. Then, a fire risk distribution and a short-term prediction of the fire propagation are calculated for every cell of the AOO. Based on the results, a Voronoi tessellation of the AOO is performed, where Voronoi centroids are waypoints. The fleet performs continuous patrol over the AOO using the waypoints. In the developed method, a fleet of drones patrols the AOO continuously via waypoints. Whenever an anomaly is detected, a single drone is dispatched from the main group to investigate – either confirm or cancel the alarm. The method for generating waypoints is analogous to [31], i.e., based on an approximate cellular decomposition of the AOO. The cells are square-shaped, their edges have a length of parameter a defined in (5)

$$a = \frac{d_s}{\sqrt{2}}, \tag{5}$$

where a is the square edge length and d_s is the cumulative fleet sensor range. The cell size determines the waypoint density and is parametrized to provide full area coverage. The value of d_s should be approximated based on the behavior and shape of the fleet resulting from the online control algorithms and the drone sensor range. For each cell, a center of priority (center of mass) is found based on the user-defined priority map. The calculated points are the fleet waypoints. They are located near high-priority areas while being constrained by cell boundaries. Unlike [36], to form a patrol path, the waypoints are ordered by finding a Hamiltonian cycle in a graph where the waypoints are nodes and edges connect the priority centers of neighboring cells. Additional edges are added for single-neighbor nodes (as in the replacement path generation for internet provision). Algorithm 7 presents the course of area patrol mission planning.

The boundary patrol scenario utilizes basic formation shapes to increase the combined field of view of drone sensors. The formations are defined in a local 2D coordinate system xy, whose center coincides with the formation center of mass, and the y-axis is oriented along the formation flight direction. The drone positions and sensor orientations in the global coordinate system (WGS-84 and NED) are obtained by translating and rotating the formation frame of reference. The sensors used in the algorithms may be represented as cameras looking forward to covering the biggest area. However, they are typically not directed towards the ground but are instead tilted at an angle relative to it [42,43]. Thus, the cover area may vary depending on the formation. Four formation shapes are defined, each with an automated generation method presented as a separate algorithm:

1. Circular, drones facing outward (Algorithm 8)
2. Circular, drones facing outward and inward (Algorithm 9)
3. V-shape (Algorithm 10)
4. V-shape with a rear guard (Algorithm 11)

Examples of formations are presented in Fig. 2. Drone positions are marked with circles, and sensor orientations with arrows. The patrol takes place along a user-defined route (a polyline) in a loop. The formation flies in straight lines and turns in waypoints. The boundary patrol mission scenario is described in Algorithm 12.

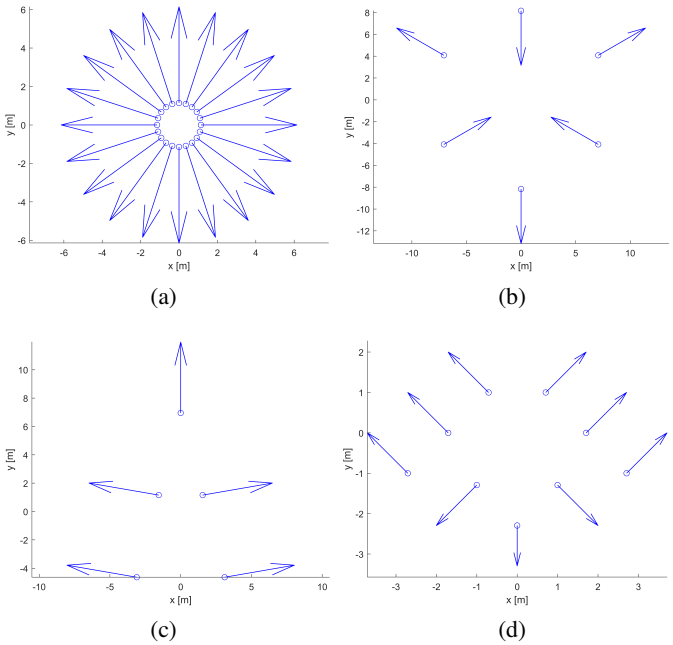
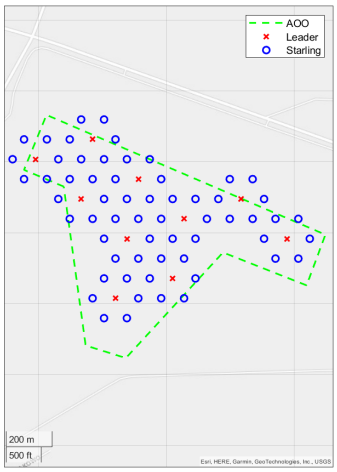


Fig. 2. The four formation shapes: (a) The circular formation with drones facing outward; (b) The circular formation with drones facing outward and inward; (c) The V-shape formation; (d) The V-formation with a rear guard

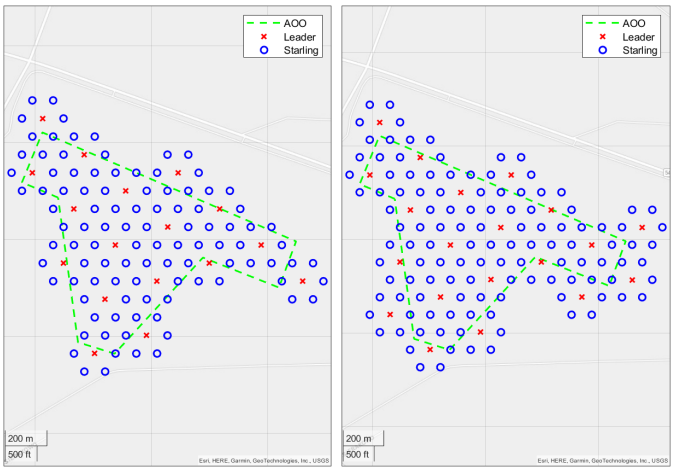
3. SIMULATIONS

3.1. Mission planning

The missions were divided into two types: internet provision and patrol. Figure 3 shows an example result of mesh generation for the three cell append conditions. The conditions are ordered from providing the smallest to the greatest area coverage, and simultaneously from the least to the most cells in the mesh. The second cell append condition is recommended, as it offers exceptionally good area coverage while allowing a reduction in the number of drones in the mesh. An example of a mesh after the redundant starling removal is presented in Fig. 4. Figure 5 depicts a mesh generated for a base of operations outside the area of operations. The rectangular path connects the base to the AOO. The location of the first leader in the mesh coincides with the base of operations. The orientation interval for iteration was set at 5°, the mixed condition weights at $w_1 = 0.7$ and $w_2 = 0.3$. Figure 6 shows an example of drone limit enforcement. There were not enough starlings, which resulted in an incomplete mesh, which could be a case for a significantly large area to be covered. The rightmost cell has a reduced number of drones. Additionally, the full cycle of the movement of cells along the replacement path was simulated. During the operational phase (**f**), the leading drone moved from the last path position to the charging station, while a new drone moved from the charging station to the first path position.



(a)



(b) (c)

Fig. 3. The cell appends conditions for an example mesh ($\alpha = 0°$): (a) leader in the AOO, (b) any starling in the AOO, (c) any boundary vertex in the AOO

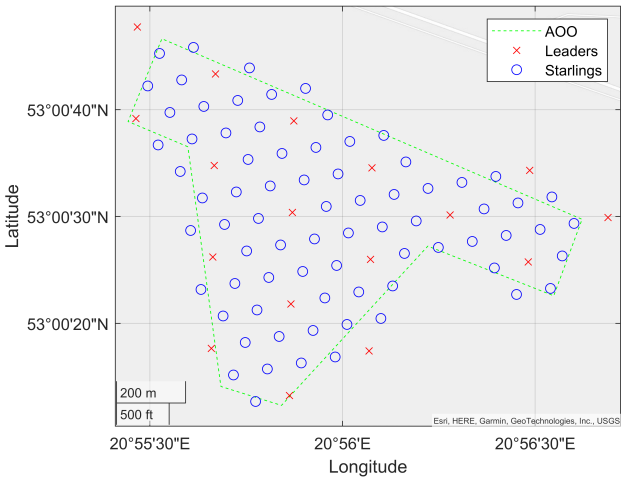


Fig. 4. A mesh after the redundant starling removal ($\alpha = 10°$)

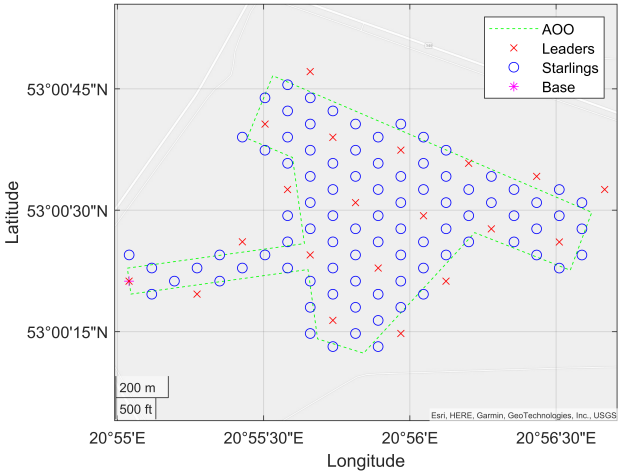


Fig. 5. A mesh with the operation start outside the AOO ($\alpha = 30°$)

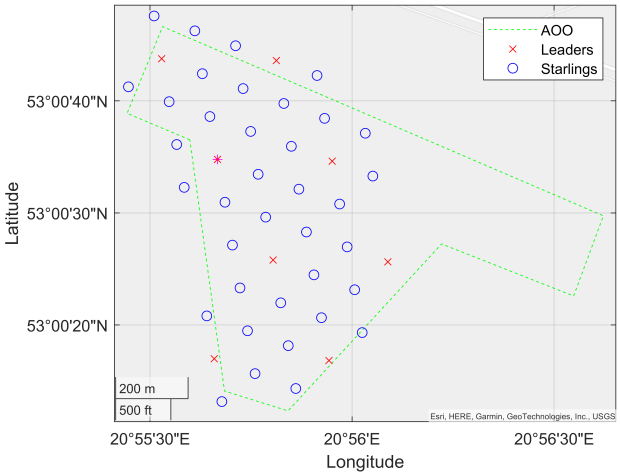


Fig. 6. An example of drone limit enforcement

3.1.1. Patrolling

Figure 7 shows the result of the automatic replacement path generation for three paths. This functionality is crucial, especially

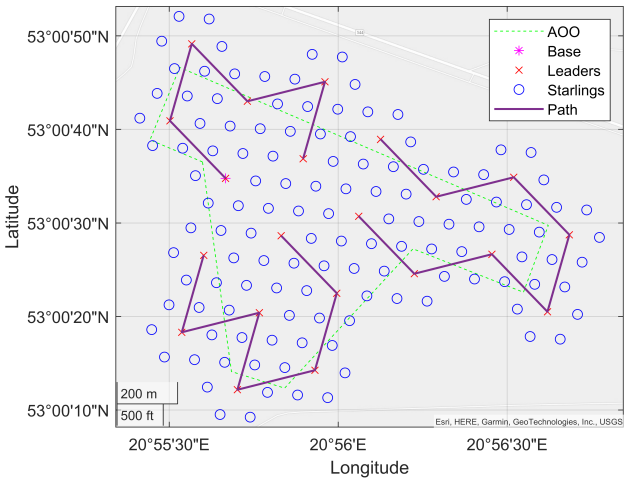


Fig. 7. An example of the automatic replacement path generation (3 paths)

for events where drones must be operated longer than the battery allows. Then the replacement algorithm is useful. Figure 8 presents an example of a V-shaped formation boundary patrol. Drone positions are marked with circles, sensor orientations with solid lines, and the AOO with dashed lines. The positions and orientation colors alternate for consecutive waypoints. The formation flies in straight lines, only turning at the patrol route polyline vertices.

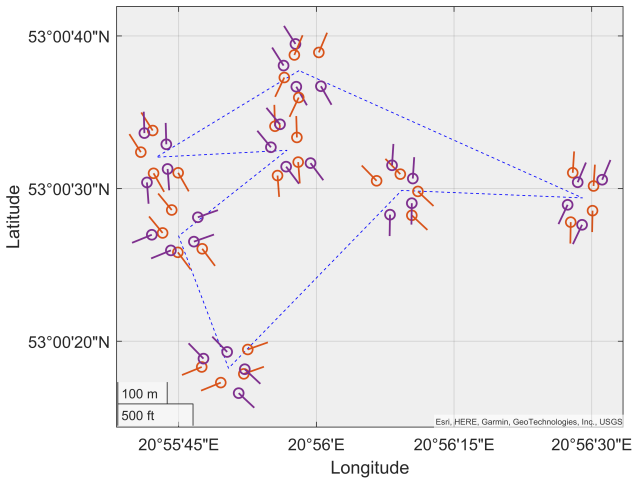


Fig. 8. An example of a boundary patrol path

4. FLIGHT TESTS

To assess if the system and algorithms developed during the research, test flights were conducted at the Warsaw University of Technology Aviation Center test airfield, as presented in Fig. 9. This is a unique infrastructure that allows us to evaluate and verify the algorithm safely. The leader drone was launched using an application operated on a mobile phone and flew according to commands issued by the operator. The followers conducted an autonomous mission of following the leader. After takeoff, the formation took the form of a hexagon as defined at the beginning of the research paper. The drones followed the leader to its position, and after the leader took off, the rest of the six drones took off as well and followed the Leader, as presented in Fig. 10. The series of drones performing the flights in groups

Fig. 9. Set of drones before the flight at Aviation Center Laboratory

to verify the implemented algorithms and communication was done. The scenario with four sets of drones (24 drones) was performed to cover the area for internet delivery. The positions of the platforms are presented in Fig. 11 – a screenshot from the Ground Control Station, from which the drones were controlled. The tests proved that the algorithms operated properly and were dependable, as well as the communication between drones and the Ground Control Station.

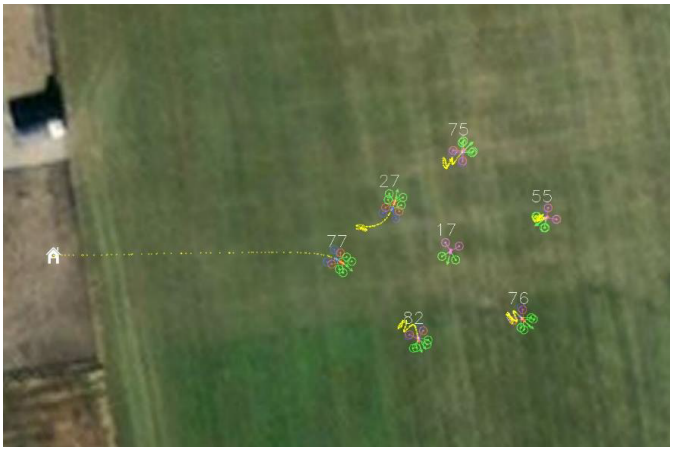


Fig. 10. Set of drones in-flight

Fig. 11. Set of four groups of drones performing area coverage scenario

5. DISCCUSION

The mission planning example results show that the mission scenarios perform as expected. The developed solutions can form the basis of mission planning for the platform. However, at the time the algorithms were developed, no data concerning the performance of online algorithms, network architecture, or GCS functionality were available. This is both a convenience and a hindrance. To accommodate the unknown functionality of the swarm components, mission planning was prepared to be as flexible as possible. Since the algorithms are offline and only generate basic output, the GCS can introduce additional functionalities to build upon the mission planning base. For example, mission reconfiguration is possible whenever a drone is lost, and no replacement is available – the GCS must only run the mission

scenario for modified parameters and send new data to the available drones. On the other hand, the limitations put on mission planning make it impossible to incorporate any functionality besides path planning directly into the scenarios. For the developed solutions, the following improvements can be made. In the internet provision mission, the replacement path generation could be expanded by new methods. Currently, there is no guarantee of the existence of a Hamiltonian path for the mesh (or its fragment). A functionality could be added in the GCS, where a user could manually modify graph edges to ensure the proper generation of the replacement path. Otherwise, a greater selection of submesh division methods could increase the chance of finding all Hamiltonian paths. The new methods could be developed to better account for path endpoint locations, with endpoints as close to the charging station as possible. Furthermore, the downside of the current k-means-based mesh division method is that the number of cells assigned to each centroid is not limited. Even though the number of centroids is chosen to fit the maximum replacement path length, the calculated paths could be longer. Currently, this can only be controlled by the corrective coefficient. A modification could be introduced to the k-means algorithm to equalize the number of submesh cells or to limit the maximum number of cells assigned to the centroid. Furthermore, a replacement path for starlings should be generated and executed following the network and online control algorithm specifications. In the patrol missions, the route-finding method for area patrol could be improved. Since the current solution is based on finding the Hamiltonian cycle in the graph, which is not guaranteed, the method should be either modified or expanded, as in the replacement path generation. A new method for ordering waypoints into a route could be proposed, where the frequency of waypoint appearance in the route is dependent on, e.g., the mean cell priority. Furthermore, the formation shape for area patrol should be defined more precisely whenever more data are available. The formation boundary patrol functionality could be extended to allow multi-formation patrol and formation reconfiguration. This would be provided by the GCS based on the repeated mission scenario calculation for different parameters.

6. CONCLUSIONS

Within the scope of this paper, a multi-function mission planning module was developed for a swarm of drones. The module consists of a set of mission scenarios from three categories: internet provision, area search, and patrolling. The internet provision scenario spans a hexagonal mesh over a specified area to provide an internet connection. Within this scenario, a method for the continuous replacement of drones for charging was developed. The area search scenarios are based on the cooperative subarea search by groups of drones and feature simultaneous search, consecutive search with patrolling, and mountain search. The patrol scenarios are made up of area patrol by an unstructured flock of drones and boundary search by rigid formations.

The developed solutions form a firm base for the development of search and rescue drone swarm solutions that can aid personnel in more than target localization. The methods can be built upon and their functionality expanded. They are easy to implement, as they are ground-station-based. While they require tweaking to match the requirements of the specific swarm structure, they show potential in disaster area management.

APPENDIX

The detailed algorithms explanation in the following tables. There is a logic behind every algorithm presented in the paper. The detailed information is also in the text. The appendix consists of 12 various algorithms used for research.

Algorithm 1 A BFS for mesh generation

```
procedure BFS(A, B, d, α, v)
input A - AOO, B - base of operations, d - distance between the neighboring drones, α - cell orientation, v - cell append condition variant
    Q.APPEND(B)                                   ▷ FIFO queue
    C ← CELL(A, B, d, α)                          ▷ First cell at the base of operations
    M.APPEND(C)                                   ▷ Add to mesh
    while Q ≠ ∅ do
        c ← Q.POP
        N ← NEIGHBOURS(c)                         ▷ Neighbouring cells
        for all n in N do
            if n ∉ M and CAN-BE-ADDED(A, n, v) = true then ▷ If the cell can be added
                C ← CELL(A, n, d, α)
                M.APPEND(C)
                Q.APPEND(n)
            end if
        end for
    end while
    return M                                      ▷ The generated mesh
end procedure
```

Algorithm 2 Cell append conditions

```
procedure CAN-BE-ADDED(A, c, d, v)                ▷ Cell append conditions
input A - AOO, c - leader, d - distance between the neighboring drones, v - cell append condition variant
    switch v do                                   ▷ Condition variants
        case 1                                    ▷ Leader in AOO
            if c ∈ A then
                return true
            end if
        case 2                                    ▷ Any starling in AOO
            if |c, A| ≤ d then
                return true
            end if
        case 2                                    ▷ Any vertex in AOO
            if |c, A| ≤ 1.5 · d then
                return true
            end if
    return false
end procedure
```

Algorithm 3 Generate the cell

```
procedure CELL(A, L, d, α)
input A - AOO, L - leader, d - distance between neighboring drones, α - cell orientation
    for i ← 1, 6 do                               ▷ Starlings' positions
        S_i ← L + d · [sin 60° · i + α, cos 60° · i + α]
        if S_i ∉ A and |S_i, A| > d/2 then        ▷ Remove redundant starlings
            S_i ← ∅
        end if
    end for
    C.STARLINGS ← S                               ▷ Create the cell
    C.LEADER ← c
    return C
end procedure
```

Algorithm 4 Connect the base and the area of operations

```
procedure ADD-PATH(A, B, d_p)
input A - AOO, B - base of operations, d_p - path width
    if B ∉ A then
        X ← X : |B, X| = |B, A|_min               ▷ Find the closest point on polygon
        v_n ← v_n ⊥ BX                            ▷ Versor normal to the path
        Y_1, Y_2 ← (X+B)/2 ± (d_p/2) · v_n
        p ← RECTANGLE(BX, Y_1Y_2)                 ▷ Rectangle from axes
    else
        p ← ∅
    end if
    A ← UNION(A, p)
    return A                                      ▷ The new AOO
end procedure
```

Algorithm 5 Mesh generation

Require: A, B, d, v, d_p, w
A - AOO, B - base of operations, d - distance between neighbouring drones, v - cell append condition, d_p - path width, w - weights
$A \leftarrow$ ADD-PATH(A, B, d_p)
for $\alpha \leftarrow 0°, 5°, 60°$ **do**
  $M \leftarrow$ BFS(A, B, d, α, v)
  if $|A - M.\text{COVERED-AREA}| < C_1$ **then**
    $C_1 \leftarrow |A - M.\text{COVERED-AREA}|$
    $R_1 \leftarrow M$
  end if
  if $|M.\text{LEADERS}| < C_2$ **then**
    $C_2 \leftarrow |M.\text{LEADERS}|$
    $R_2 \leftarrow M$
  end if
  if $w_1 \cdot |A - M.\text{COVERED-AREA}| + w_2 \cdot |M.\text{LEADERS}| < C_3$ **then**
    $C_3 \leftarrow w_1 \cdot |A - M.\text{COVERED-AREA}| + w_2 \cdot |M.\text{LEADERS}|$
    $R_3 \leftarrow M$
  end if
end for
return R_1, R_2, R_3

Algorithm 6 Automatic replacement path generation

Require: L, n, H
L - leaders, n - no. replacement paths, H - charging station
$a \leftarrow$ KMEANS(L, n) ▷ Mesh division
for $i \leftarrow 1, n$ **do**
  $m_i \leftarrow L : a_L = i$ ▷ Submesh
  $e \leftarrow C(m, 2)$ ▷ Leader pair combinations
  $e \leftarrow$ SORT$(e, \sum_{j=1,2} |H, e_{x,j}|)$ ▷ Sort pairs by their combined distance from H
  for $x \leftarrow 1, |e|$ **do**
    $p_i \leftarrow$ HAMILTONIAN(m_i, e_x) ▷ Get Hamiltonian path
    if $p_i \notin \emptyset$ **then**
      break ▷ Path found for the submesh
    end if
  end for
end for
return p ▷ Replacement paths

Algorithm 7 Area patrol

Require: A - AOO, P - priority map, a - cell size
$C \leftarrow$ CELLULAR-DECOMPOSITION(A, a) ▷ Approximate cellular decomposition
for all c in C **do**
  $X_c, p_c \leftarrow P \in c$ ▷ Priority map points in the cell
  $W_c \leftarrow \frac{\sum_i X_{c_i} \cdot p_{c_i}}{\sum_i p_{c_i}}$ ▷ Waypoint location
end for
$H \leftarrow$ HAMILTONIAN(W) ▷ Patrol path
return H

Algorithm 8 Circular formation no. 1

procedure CIRCLE1(n, d)
input n - no. drones, d - distance between neighbouring drones
  $\alpha \leftarrow \frac{2\pi}{n}$
  $D \leftarrow \frac{d}{\sqrt{(2)}\sin\alpha}$
  for $i \leftarrow 1, n$ **do**
    $x_i \leftarrow D \cdot [\sin((i-1)\cdot\alpha), \cos((i-1)\cdot\alpha)]$ ▷ Drone position
    $\theta_i \leftarrow$ NORM(x_i) ▷ Sensor orientation versor
  end for
  return x, θ
end procedure

Algorithm 9 Circular formation no. 2

procedure CIRCLE2(n, r, d)
input n - no. drones, r - no. sensors facing inward, d - distance between neighbouring drones
  $x, \theta \leftarrow$ CIRCLE1(n, d)
  $j \leftarrow$ ROUND(n/r)
  for $i \leftarrow 1, j, n$ **do**
    $\theta_i \leftarrow -\theta_i$ ▷ Reverse every j^{th} sensor's orientation
  end for
  return x, θ
end procedure

Algorithm 10 V-shape formation no. 1

procedure V1(n, θ, d)
input n - no. drones, θ - angle between formation branches, d - distance between neighbouring drones
  $D \leftarrow d \cdot [\sin\theta/2, \cos\theta/2]$
  $x, \theta \leftarrow \emptyset$
  if $n\%2 = 1$ **then** ▷ Odd no. drones
    $x_1 \leftarrow [0, 0]$ ▷ Central (first) drone
    $\theta_1 \leftarrow [0, 1]$ ▷ First sensor faces forward
    for $s \leftarrow [1, -1]$ **do** ▷ For each formation branch
      for $i \leftarrow 1, \lfloor n/2 \rfloor$ **do**
        x.APPEND$(x_1 + i \cdot D \cdot [s, -1])$
        θ.APPEND$([sD_2, 1])$
      end for
    end for
  else ▷ Even no. drones
    for $s \leftarrow [1, -1]$ **do**
      x.APPEND$(\frac{d}{2} \cdot [s, -\cot\theta/2])$ ▷ No central drone
      θ.APPEND$([sD_2, 1])$
      $j \leftarrow |x|$
      for $i \leftarrow 1, \lfloor n/2 \rfloor - 1$ **do** ▷ For each formation branch
        x.APPEND$(x_j + i \cdot D \cdot [s, -1])$
        θ.APPEND$([sD_2, 1])$
      end for
    end for
  end if
  $C \leftarrow$ CENTROID(x) ▷ Formation's centroid
  $x \leftarrow$ TRANSLATE(x, C) ▷ Translate the formation frame of reference to the centroid
  return x, θ
end procedure

Algorithm 11 V-shape formation no.2 (with a rear guard)

procedure V2(n, θ, d, r)
input n - no. drones, θ - angle between formation branches, d - distance between neighbouring drones, r - no. drones in the rear guard
  $x, \theta \leftarrow$ V1$(n - r, \theta, d)$ ▷ Main formation
  $x_G, \theta_G \leftarrow$ V1(r, θ, d) ▷ Rear guard
  $x_G, \theta_G \leftarrow$ ROTATE$(x_G, \theta_G, [0, -1])$ ▷ Rotate the read guard by 180°
  $x_x \leftarrow x_x :$ CENTROID$(x_G + x_x) \in OY$ and
    $x_{G_{1,2}} + x_x < x_{1,2}$ and
    DISTANCE$(x, x_G)_{min} = d$ ▷ Find the rear guard translation vector
  $x_G \leftarrow x_G + x_x$
  return $[x, x_G], [\theta, \theta_G]$
end procedure

Algorithm 12 Boundary patrol

Require: T, f, p
T - patrol route (polyline), f - formation type, p - formation parameters
(The formation parameters must match the formation type)
$X \leftarrow \emptyset$ ▷ Drone waypoints and sensor orientations
$F \leftarrow$ FORMATION(f, p) ▷ Create and instance of the formation
for $i \leftarrow 1, |T|$ **do**
  $\vec{v_p} \leftarrow \vec{v_p} \parallel \overrightarrow{T_i T_{i+1}}$
  X.APPEND$(F.\text{AT}(T_i, \vec{v_p}))$
  $\gg F.\text{AT}(T_i, \vec{v_p})$ - drone locations and sensor orientation of the formation moved to the point T_i and oriented along the vector $\vec{v_p}$
  X.APPEND$(F.\text{AT}(T_{i+1}, \vec{v_p}))$
end for
$\vec{v_p} \leftarrow \vec{v_p} \parallel \overrightarrow{T_{end} T_1}$
X.APPEND$(F.\text{AT}(T_{end}, \vec{v_p}))$
X.APPEND$(F.\text{AT}(T_1, \vec{v_p}))$
return X

ACKNOWLEDGEMENTS

The work was supported by the National Centre of Research and Development as part of the POIR program under research project agreement POIR.04.01.04-00-0078/20.

REFERENCES

[1] M.P.R. Arnold, K. Carey, B. Abruzzo, and C. Korpela, "What is a robot swarm: A definition for swarming robotics," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019, pp. 74–81, doi: 10.1109/UEMCON47517.2019.8993024.

[2] G.B. Lamont, J.N. Slear, and K. Melendez, "UAV Swarm Mission Planning and Routing using Multi-Objective Evolutionary

Algorithms," in *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, vol. 1, 2007, pp. 10–20, doi: 10.1109/MCDM.2007.369410.

[3] X.L. Pyke and C. Stark, "Dynamic Pathfinding for a Swarm Intelligence Based UAV Control Model Using Particle Swarm Optimization," *Front. Appl. Math. Stat.*, vol. 7, 2021, doi: 10.3389/fams.2021.744955.

[4] Y. Fu, M. Ding, and C. Zhou, "Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAV," *IEEE Trans. Syst. Man Cybern. Paart A-Syst. Hum.*, vol. 42, no. 2, pp. 511–526, 2012, doi: 10.1109/TSMCA.2011.2159586.

[5] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning," *IEEE Trans. Ind. Inform.*, vol. 9, no. 1, pp. 132–141, 2013, doi: 10.1109/TII.2012.2198665.

[6] F. Augugliaro, A.P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a Quadro copter fleet: A sequential convex programming approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2012, pp. 1917–1922, doi: 10.1109/IROS.2012.6385823.

[7] W. Chen, J. Liu, and H. Guo, "Achieving Robust and Efficient Consensus for Large-Scale Drone Swarm," *IEEE Trans. Veh. Technol.*, vol. 69, pp. 15867–15879, 2020, doi: 10.1109/TVT.2020.3036833.

[8] A. Kopyt, M. Sochacki, K. Kaczmarek, "Tool for Analysis of Traffic of Vertical Take-off and Landing Aircraft in Urban Agglomerations," *J. Theor. Appl. Mech.*, vol. 63, no. 1, pp. 103-113, 2025, doi: 10.15632/jtam-pl/195717.

[9] O. Shrit, S. Martin, K. Alagha, and G. Pujolle, "A new approach to realize drone swarm using ad-hoc network," in *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2017, pp. 1–5, doi: 10.1109/MedHocNet.2017.8001645.

[10] G. Asaamoning, P. Mendes, D. Rosário, and E. Cerqueira, "Drone Swarms as Networked Control Systems by Integration of Networking and Computing," *Sensors*, vol. 21, p. 2642, 2021, doi: 10.3390/s21082642.

[11] Q. Cui, P. Liu, J. Wang, and J. Yu, "Brief Analysis of Drone Swarms' Communication," in *2017 IEEE International Conference on Unmanned Systems (ICUS)*, 2017, pp. 463–466, doi: 10.1109/ICUS.2017.8278390.

[12] E. Soria, F. Schiano, and D. Floreano, "Distributed Predictive Drone Swarms in Cluttered Environments," *IEEE Robot. Autom. Lett.*, vol. 7, pp. 73–80, 2022, doi: 10.1109/LRA.2021.3118091.

[13] D. Wu *et al.*, "ADDSEN: Adaptive Data Processing and Dissemination for Drone Swarms in Urban Sensing," *IEEE Trans. Comput.*, vol. 66, pp. 183–198, 2017, doi: 10.1109/TC.2016.2584061.

[14] E. Soria, F. Schiano, and D. Floreano, "SwarmLab: a Matlab Drone Swarm Simulator," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8005–8011, doi: 10.1109/IROS45743.2020.9340854.

[15] F. Saffre, H. Hildmann, and H. Karvonen, "The Design Challenges of Drone Swarm Control," in *Engineering Psychology and Cognitive Ergonomics*, D. Harris and W.C. Li, Eds., Cham: Springer, 2021, pp. 408–426, doi: 10.1007/978-3-030-77932-0_32.

[16] M. Lyu, Y. Zhao, C. Huang, and H. Huang, "Unmanned Aerial Vehicles for Search and Rescue: A Survey," *Remote Sens.*, vol. 15, p. 3266, 2023. doi: 10.3390/rs15133266.

[17] Z. Goraj *et al.*, "High Altitude Long Endurance Unmanned Aerial Vehicle of a New Generation – a Design Challenge for a Low Cost, Reliable and High-Performance Aircraft," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 52, no. 3, p. e153827, 2004; doi: 10.24425/bpasts.2025.153827.

[18] M. Krawczyk, "Conditions for Unmanned Aircraft Reliability Determination," *Eksploat. Niezawodn.-Mainten. Reliab.*, vol. 15, no. 1, pp. 31–36, 2013.

[19] S. Grogan, R. Pellerin, and M. Gamache, "The Use of Unmanned Aerial Vehicles and Drones in Search and Rescue Operations – a survey," in *Proc. PROLOG*, 2018, pp. 1–13.

[20] M.G. Cimino, A. Lazzeri, and G. Vaglini, "Combining stigmergic and flocking behaviors to coordinate swarms of drones performing target search," in 6^{th} *International Conference on Information, Intelligence, Systems and Applications (IISA)*, IEEE, 2015, pp. 1–6, doi: 10.3233/idt-170317.

[21] K.B. Lee, Y.J. Kim, and Y.D. Hong, "Real-time swarm search method for real-world quadcopter drones," *Appl. Sci.*, vol. 8, p. 1169, 2018, doi: 10.1007/978-3-031-75107-3_1.

[22] A.L. Alfeo, M.G. Cimino, N. De Francesco, M. Lega, and G. Vaglini, "Design and simulation of the emergent behavior of small drones swarming for distributed target localization," *J. Comput. Sci.*, vol. 29, pp. 19–33, 2018, doi: 10.1038/s41598-023-39926-5.

[23] H. Duan, J. Zhao, Y. Deng, Y. Shi, and X. Ding, "Dynamic discrete pigeon-inspired optimization for multi-UAV cooperative search-attack mission planning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, pp. 706–720, 2020, doi: 10.1155/2024/5876393.

[24] J. Berger and N. Lo, "An innovative multi-agent search-and-rescue path planning approach," *Comput. Oper. Res.*, vol. 53, pp. 24–31, 2015, doi: 10.1016/j.cor.2014.06.016.

[25] N.A. Kyriakakis, M. Marinaki, N. Matsatsinis, and Y. Marinakis, "Moving peak drone search problem: An online multi-swarm intelligence approach for UAV search operations," *Swarm Evolution. Comput.*, vol. 66, p. 100956, 2021, doi: 10.1016/j.cor.2014.06.016.

[26] C.A. Baker, S. Ramchurn, W.L. Teacy, and N.R. Jennings, "Planning search and rescue missions for UAV teams," in *Proc. 22nd European Conference on Artificial Intelligence*, 2016, pp. 1777–1778, doi: 10.3390/rs15133266.

[27] T.M. Cabreira, L.B. Brisolara, and P.R. Ferreira Jr., "Survey on Coverage Path Planning with Unmanned Aerial Vehicles," *Drones*, vol. 3, p. 4, 2019, doi: 10.3390/drones3010004.

[28] E. Galceran and M. Carreras, "A Survey on Coverage Path Planning for Robotics," *Robot. Auton. Syst.*, vol. 61, pp. 1258–1276, 2013, doi: 10.1016/j.robot.2013.09.004.

[29] L. Lin and M.A. Goodrich, "UAV intelligent path planning for wilderness search and rescue," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 709–714, doi: 10.1016/j.robot.2013.09.004.

[30] A. Ghaddar and A. Merei, "Energy-aware grid-based coverage path planning for UAVs," in *Proc. SENSORCOMM 2019*, 2019, pp. 34–45, doi: 10.3390/s20133742.

[31] J. Huang, W. Fu, S. Luo, C. Wang, B. Zhang, and Y. Bai, "A Practical Interlacing-Based Coverage Path Planning Method for Fixed-Wing UAV Photogrammetry in Convex Polygon Regions," *Aerospace*, vol. 9, p. 521, 2022, doi: 10.3390/aerospace9090521.

[32] M. Torres, D.A. Pelta, J.L. Verdegay, and J.C. Torres, "Coverage path planning with unmanned aerial vehicles for 3D terrain

reconstruction," *Expert Syst. Appl.*, vol. 55, pp. 441–451, 2016, doi: 10.3390/drones3010004.

[33] L.D. Nielsen, I. Sung, and P. Nielsen, "Convex Decomposition for a Coverage Path Planning for Autonomous Vehicles: Interior extension of edges," *Sensors*, vol. 19, p. 4165, 2019, doi: 10.3390/s19194165.

[34] Y. Li, H. Chen, M.J. Er, and X. Wang, "Coverage Path Planning for UAVs Based on Enhanced Exact Cellular Decomposition Method," textitMechatronics, vol. 21, pp. 876–885, 2011, doi: 10.1016/j.phycom.2023.102073.

[35] G. Skorobogatov, C. Barrado, E. Salamí, and E. Pastor, "Flight planning in multi-unmanned aerial vehicle systems: Nonconvex polygon area decomposition and trajectory assignment," *Int. J. Adv. Robot. Syst.*, vol. 18, 2021, doi: 10.1177/1729881421989551.

[36] A. Howard, M.J. Matarić, and G.S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. Distributed Autonomous Robotic Systems 5*, Springer, 2002, pp. 299–308, doi: 10.1023/A:1019625207705.

[37] A. Giuseppi, R. Germanà, F. Fiorini, F. Delli Priscoli, and A. Pietrabissa, "UAV Patrolling for Wildfire Monitoring by a Dynamic Voronoi Tessellation on Satellite Data," *Drones*, vol. 5, p. 130, 2021, doi: 10.3390/drones5040130.

[38] D. Kinaneva, G. Hristov, J. Raychev, and P. Zahariev, "Early Forest fire detection using drones and artificial intelligence," in *2019 42nd International Convention*, 2019, pp. 1–6, doi: 10.1007/978-3-030-68080-0_8.

[39] N.Okati, B. Razeghi, M.R. Mosavi, "On relay selection to maximize coverage region for cooperative cellular networks with multiple fixed and unfixed relays" in *Proc. 2015 6th International Conference on Computing, Communication, and Networking Technologies (ICCCNT), IEEE*, 2015, pp. 1–6.

[40] V. Paruchuri, A. Durresi, R. Jain, "Optimized Flooding Protocol for ad hoc Networks," arXiv preprint cs/0311013 2003.

[41] T.H. Cormen *et al.*, *Introduction to Algorithms*. MIT Press, 2022.

[42] K. Gromada, B. Siemiątkowska, W. Stecz, K. Płochocki, and K. Woźniak, "Real-Time Object Detection and Classification by UAV Equipped With SAR," *Sensors*, vol. 22, no. 5, p. 2068, 2022, doi: 10.3390/s22052068.

[43] K. Gromada and P. Nalewajko, "Recognition analysis issues for tactical unmanned aerial vehicles based on optical photographs and SAR scans," in *Proc SPIE – The International Society for Optics and Photonics, Radioelectronic Systems Conference 2019*, P. Kaniewski and J. Matuszewski, Eds. vol. 11442, 2019, pp. 1–10, doi: 10.1117/12.2564987.