In: 42nd Digital Avionics Systems Conference (DASC), 1-5 October 2023, Barcelona, Spain
DOI: 10.1109/DASC58513.2023.10311218

Mission Planning for a Multiple-UAV Patrol System in an Obstructed Airport Environment

Ruifan Liu
School of Aerospace, Transport and Manufacturing
Cranfield University
Bedfordshire, United Kingdom
ruifan.liu@cranfield.ac.uk

Hyo-Sang Shin
School of Aerospace, Transport and Manufacturing
Cranfield University
Bedfordshire, United Kingdom
h.shin@cranfield.ac.uk

Antonios Tsourdos
School of Aerospace, Transport and Manufacturing
Cranfield University
Bedfordshire, United Kingdom
a.tsourdos@cranfield.ac.uk

***Abstract*—This paper investigates using multiple unmanned aerial vehicles (UAVs) to carry out routine patrolling at an airport to enhance its perimeter security. It specifically focuses on mission planning of the system to facilitate efficient patrolling with consideration of local buildings and restricted airspace. The proposed methodology includes three aspects: 1) a vision-based set cover algorithm to construct the patrolling network, 2) an obstructed partitioning-based clustering algorithm for recharging station placement, and 3) a mixture integer quadratic programming (MIQP) algorithm to plan routes for UAVs minimizing the maximum idle time through out all patrolling waypoints. The main contribution of this work is that it provides a comprehensive mission planning solution for UAVs persistently patrolling in a complex environment characterized by blocked vision and restricted airspace. The proposed methodology is evaluated through intensive simulations in the context of the Cranfield Airport scenario.**

***Index Terms*—unmanned aerial vehicle, patrolling, station deployment, mission planning, obstructed environment**

I. Introduction

Airport perimeter security has focused attention on the critical importance of securing the first line of defence. The primary principle of Perimeter security involves deferring, detecting, and delaying intrusions, as well as intercepting potential intruders to prevent harm to people or property [1]. To ensure effectiveness, regular patrols by security personnel are advised for condition inspection, maintenance problem identification (e.g., damaged fences, tampering with security devices), and detection of individuals hiding in remote areas. However, staff resources are constrained at airports. In this context, autonomous aerial vehicles have emerged as a promising solution [2], which has the advantage of working round the clock and a bird's-eye view of the airport premises, thereby motivating research into multi-UAV airport patrol systems.

Drones have great characteristics such as easy deployment and cost efficiency, while their lightweight design also denotes a very limited battery endurance. To facilitate persistent patrolling, the deployment of recharging stations is motivated and investigated on the subject of deciding the station placement to maximize the coverage demand with the least amount of infrastructure investment [3], [4]. Similar topics have also been explored in terms of locating vertiports for urban air mobility (UAM). German et al. [5] addressed the optimization problem of identifying suitable vertiport locations for package delivery in the San Francisco Bay area. Vertiports in terms of air mobility transport in Florida Country is resolved utilizing a network model [6]. Jeong et al. [7] employed the K-means clustering algorithm considering commuters as the main customers to decide vertiport locations for UAM operations in Seoul city, leveraging the power of artificial intelligence (AI). Junghyun et al. [8] further proposed a partitioning-based K-means clustering algorithm to account for administrative partitions. Although the literature has demonstrated the selection of vertiport locations for UAM using optimization or AI methods, they mostly construct the optimization metric upon geometric distances. For routine patrolling in a busy airport, placing stations needs the extra consideration of restricted airspace. Thus, this paper constructs a patrolling network that incorporates airspace constraints and proposes an obstructed clustering algorithm to identify the optimal recharging station sites.

Routine patrol in airports requires covering a changing environment, where all areas of the patrol space must be visited infinitely often. It is recognized as a persistent monitoring (PM) mission which is repetitive in nature and last for prolonged periods of time. A route optimization problem arises in this context that seeks a set of routes for a group of robots to repeatedly visit targets of interest. In terms of the performance index, it is desirable to minimize the idleness time, which refers to the maximum duration between consecutive visits to any target. Hari et al. [9] discussed the PM problem with a single drone, where the drone's operational range between two sequential service stops is described as a visit number k. The optimal solution is then constructed by finding the walk of k visits with the least revisit time and concatenating it. Rezazadeh et al. [10] investigated the multi-agent PM problem, with a focus on maximizing a utility function tied to the probability of event detection. In this work, a receding horizon sequential greedy algorithm is proposed to generate sub-optimal solutions with a polynomial computation cost and guaranteed bound on optimality. It shows that finding the optimal monitoring policy that defines the sequence of the geographical nodes for each agent is an NP-hard problem. Another theoretical analysis for multi-agent patrolling in [11] has indicated that if agents take a

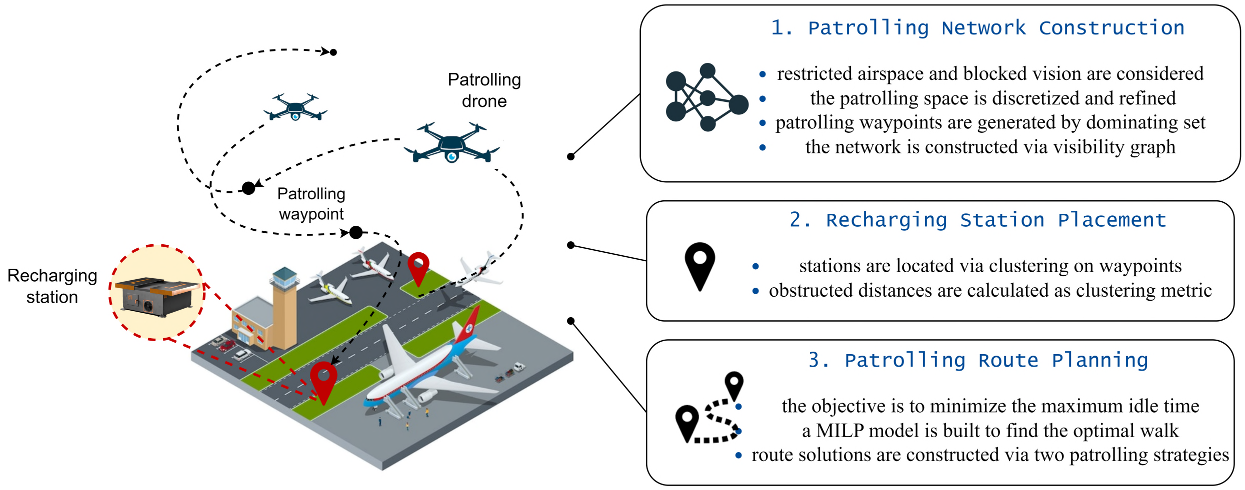


Fig. 1. Overview of the three aspects of mission planning for multi-UAV routine patrolling system.

cyclic walk over the whole graph, the optimal strategy in terms of the shortest idleness is the solution based on solving the corresponding Travel Salesman Problem (TSP). More recently, AI techniques are used to solve a visibility-based multi-agent PM problem [12]. They present a Multi-agent Graph Attention Proximal Policy Optimization (MA-G-PPO) algorithm to learn a policy for each agent. Information sharing between agents is achieved via graph attention, which facilitates an effective joint policy. Despite extensive research on multi-agent patrolling, the patrolling problem inclusive of frequent visits to recharging facilities has received limited attention. This is vital in the context of multi-UAV patrolling, where the charging time of the UAV's battery is not negligible compared to its flight time.

Our work focuses on mission planning for routine patrolling using multiple drones at airports. To achieve intensive, continuous and efficient patrolling, we consider the mission planning as threefold (shown in Figure 1): patrolling network construction incorporated with local buildings and airspace restrictions, recharging station placement, and route optimization for drones to minimize the maximum idle time over the graph. Correspondingly, the three main contributions of this paper are:

1) A vision-based patrolling network is constructed considering the obstructed airspace as a consequence of local buildings and airspace regulations. Patrolling waypoints are iteratively generated using the dominating set of a discretized patrolling space. A patrol network is then generated via the visibility graph algorithm, connecting each pair of nodes with accessible paths.
2) Locations of recharging stations are optimized via an obstructed partitioning-based clustering algorithm, to minimize the total flight distances between station sites and waypoints while considering the blocked airspace.
3) A MIQP model is built to generate optimal patrol walks with the minimum travel time. Based on the MIQP model, two routing methods to construct the patrolling route solutions for multi-UAV systems.

The rest of the paper is organized as follows. Section II introduces how the patrolling network is constructed, including the selection of patrolling waypoints and the suggestion for recharging station placing. This is followed by patrolling route planning in Section III, where two routing methods and a MIQP formulation to find the optimal walk are both introduced. Section IV concludes this study and outlines a brief plan for future research.

II. Patrolling Network

A. Patrolling Area Configuration

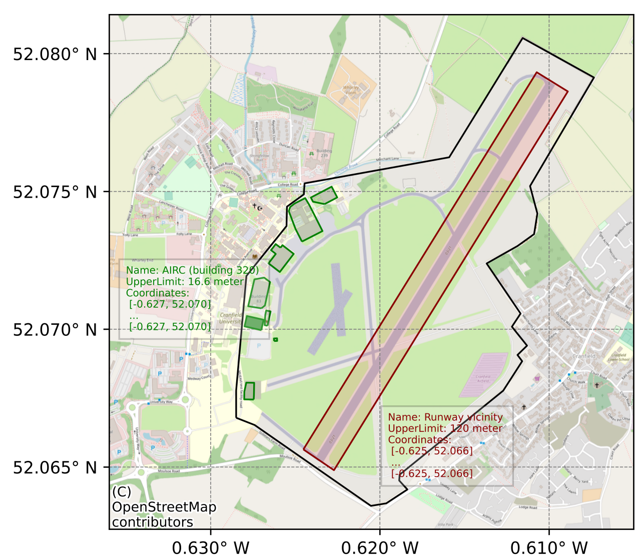


Fig. 2. Patrolling mission area configuration for Cranfield Airport. Green-lined polygons indicate the local buildings and the Red-lined rectangle is the restricted airspace of runaway. Texts next to the polygons give two examples of obstructed airspace information.

The first thing for establishing a patrolling network is to identify the area for drones to patrol over, which is exclusive of restricted airspace and avoids collision with terminals and other infrastructures. The patrol normally covers the entire length of the perimeter fences that surrounds the airport, and the critical areas such as the fuel storage facilities, power substations, and air traffic towers. For Cranfield Airport, we consider patrolling over the whole airfield and vulnerable infrastructures including airplane hangars and communication towers, while excluding the runway and its vicinity.

Figure 2 shows the patrolling area and obstructed airspace in the Cranfield Airport case. The patrolling area is outlined via black lines. The green polygons indicate local buildings and the red area is restricted airspace for not disrupting normal air operations. Those forbidden airspace is presented as three-dimensional polyhedrons with coordinates and altitude limits. Geographic features of local buildings are collected from Open Street Map [13]. The vicinity of 50 meters close to the runway is identified as restricted airspace.

B. Patrolling Waypoint Selection

To guard the identified region, we propose a new concept called *patrolling waypoint*. These waypoints are used as control points along the drone's path, where they can hover and conduct thorough inspections of the adjacent area. The key point of this formulation is positioning those *patrolling waypoints*, minimizing the number of points while ensuring the complete coverage of the entire area within a limited visible range of UAVs.

The above problem can be cast into a computational geometry problem, namely the *art gallery problem*, which is originally studied by [14] and the aim is to find the smallest number of guards necessary to lookout every point in the interior of an n-wall art gallery room. The computational complexity of the art gallery problem has been proved to be NP-hard, either for inspection areas without holes or with holes. Holes refer to obstacles that can not be set up with guards or obstruct the line of sight.

In this work, we adopt an approximation algorithm inspired by the work in [15] to address it. The proposed algorithm consists of discretizing the coverage region into a finite set of points and then solving the dominating set problem over those points. For those shaded regions that are not covered, the area discretization is iteratively refined until a solution is found that achieves complete coverage from the selected observation nodes.

Specifically, the proposed algorithm is implemented by following four steps:

- Step 1: the patrolling area is divided into grids. Centroids of these grids are the first instance of discretization. Centroids within restricted airspace or obstacles are removed.
- Step 2: a visibility graph is established over those discretization nodes. Vertexes are connected if the straight line connecting them does not intersect with any obstacles and is within a certain visible range.
- Step 3: observation points (e.g., patrolling waypoints in the context) are selected by solving the dominating set problem of the visibility graph. A dominating set in a graph is a subset of vertices such that every vertex in the graph is either in the dominating set or adjacent to a vertex in the dominating set.
- Step 4: the union of coverage areas is obtained based on the current observation points. If the set of observation points completely covers the mission area, a viable solution is found, and the algorithm terminates. Otherwise, the area discretization is refined by adding one more point for each uncovered region (e.g., its centroid), and the algorithm proceeds from Step 2 with the updated discretization.

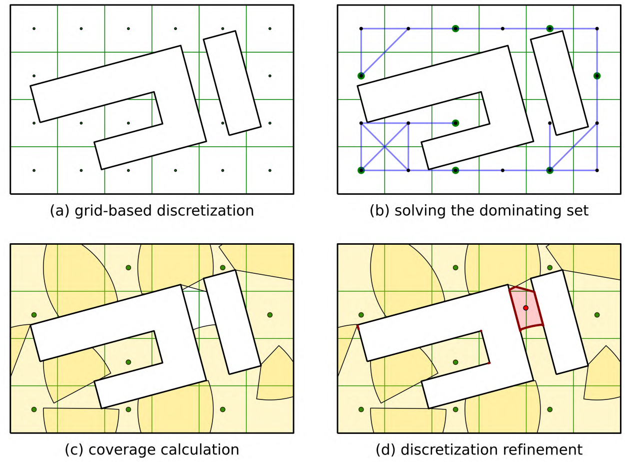


Fig. 3. Four steps to generate patrolling waypoints. If there is an uncovered area in the last coverage, shown as the red area in plot (d), one corresponding node is added to the discretized space, and the algorithm repeatedly proceeds from the plot (a).

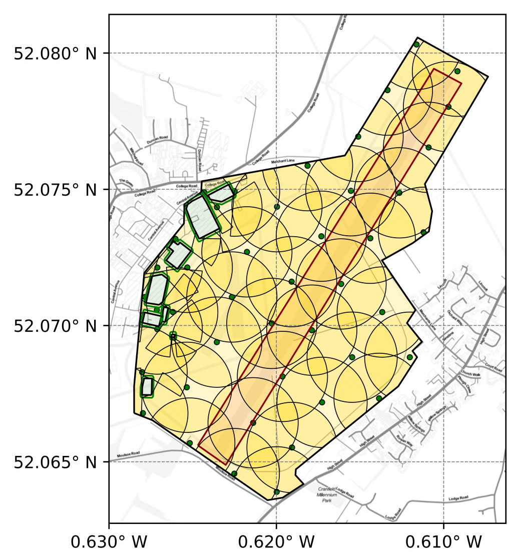


Fig. 4. The patrolling waypoints generated in Cranfield Airport scenario. Grid size in discretization is 100 meters, and the visibility range of drones is assumed to be 300 meters

The four plots in Figure 3 illustrates one iteration of the proposed method. For the first solution, there is an uncovered blank area between two buildings (shown in Figure 3 (c)). This means the current selection of patrolling points is not feasible. Therefore, one more discretization point, i.e. the centroid of the uncovered area, represented by the red dot in the figure 3 (d), is added to the discretization space. The process is then repeated with the updated instance of discretization until the patrolling area is completely covered. Patrolling waypoints for Cranfield Airport are presented in Figure 4, where the yellow circular areas indicate the limited vision from each patrolling point.

C. Recharging Station Placement

This work adopts an obstructed clustering algorithm to locate recharging stations, with additional consideration of the restricted airspace and buildings. More specifically, we adapt the K-means clustering algorithm to all the patrolling waypoints generated in the previous steps and locate stations at the centre of clusters to minimize the overall flight distance between waypoints and stations. To find the clusters that minimize the overall internal distances, the K-means clustering method iteratively assigns data points to the cluster with the nearest centroid and updates the centroids based on the new assignment. During this process, three issues arise with obstacles and restricted airspace: 1) the centre of the cluster may turn out to be inaccessible, e.g., in the middle of a building or inside of forbidden airspace; 2) every time new clusters are identified, new centres are generated at their mean coordination to minimize the overall distances. However, it is not applicable for obstructed scenarios; 3) while updating a new centre, the distances between each node and potential cluster centres have to be recomputed. The computational cost can become very high with a large number of vertexes and obstacles [16].

We develop an obstructed discrete K-means clustering algorithm to address the station deployment problem with obstacles. The algorithm is described in Algorithm 1. Accessible space is discretized into grids, each of which holds a candidate location for stations. k cluster centres $c_1, ..., c_k$ are first randomly selected from those candidates, and nodes are assigned to the nearest centroid according to their obstructed distances. Then, the algorithm searches the neighbourhood of the current centres and updates the centre if a smaller within-cluster distance is found. The process is repeated until no new centroid can be found.

Referring to Algorithm 1, a constant operation is to calculate the obstructed distances between objects and centroids to assign the object or to update the centroid. Here we use the visibility graph algorithm to calculate the obstructed distance in K-means clustering to address the station location problem with obstacles. All obstacles are presented as polygons. Computing the obstructed distance follows these two steps:

- Step1: All n object points and vertices of all m polygonal obstacles $v(P), P = \{p_1, ..., p_m\}$ are added into a visibility graph $V(v)$. If the edge between two nodes of the graph, $e(v_a, v_b)$ is visible, i.e., not intersecting any of the obstacles. It is also added to the graph $V(v, e)$.
- Step2: Every time the centroid o_i is updated, an augmented visibility graph V' is constructed with the new node o_i and all visible arcs connected to o_i. The obstructed distance between the point v_j and the centroid o_i is then calculated as the shortest path in the visibility graph $ShortestPathV(v_j, o_i)$.

Algorithm 1 Obstructed Discrete K-means Clustering

Input: A set of n objects, number of cluster k, centroid candidate c and clustering parameter, $maxtry$.
Output: A partition of the n objects into k clusters with cluster centers, $c_1, ..., c_k$.
Method:

randomly select k centroid candidates to be $c_1, ..., c_k$;
for each object $p = 1, ..., n$ **do**
    find the nearest center $o^{(p)}$, and assign p to it.
end for
compute the sum of obstructed distances D;
let $current = c_1, ..., c_k$, $currentD = D$
while true **do**
    for each centre $o_i = o_1, ..., o_k$ **do**
        for each neighbor $o_i^{(n)}$ of o_i **do**
            replace o_i with $o_i^{(n)}$;
            compute the sum of obstructed distances D';
            if $D' < D$ **then**
                centroids change to $o_1, ..., o_i^{(n)}, ..., o_k$;
                distance updates $D \leftarrow D'$
            end if
        end for
    end for
    if $current = o_1, ..., o_k$ **then** /*centroid unchanged*/
        break;
    else
        for each object $p = 1, ..., n$ **do**
            find the nearest center $o^{(p)}$, and assign p to it.
        end for
        compute the sum of obstructed distances D;
        let $current = c_1, ..., c_k$, $currentD = D$
    end if
end while

The results for station placement with varying station numbers are depicted in Figure 5. Collision-free airpath to the nearest cluster centres are also depicted.

D. Patrolling Network Construction

The patrolling network is constructed via the visibility graph algorithms. The network includes all the vertexes (i.e., patrolling waypoints and recharging stations), and air pathways that connect those vertexes. To find the pathways between two nodes which are not visible from each other, we add the vertices of all obstacles including buildings and restricted airspace into the graph and connect every two nodes if their arc does not intersect with obstacles. With the assistance of these

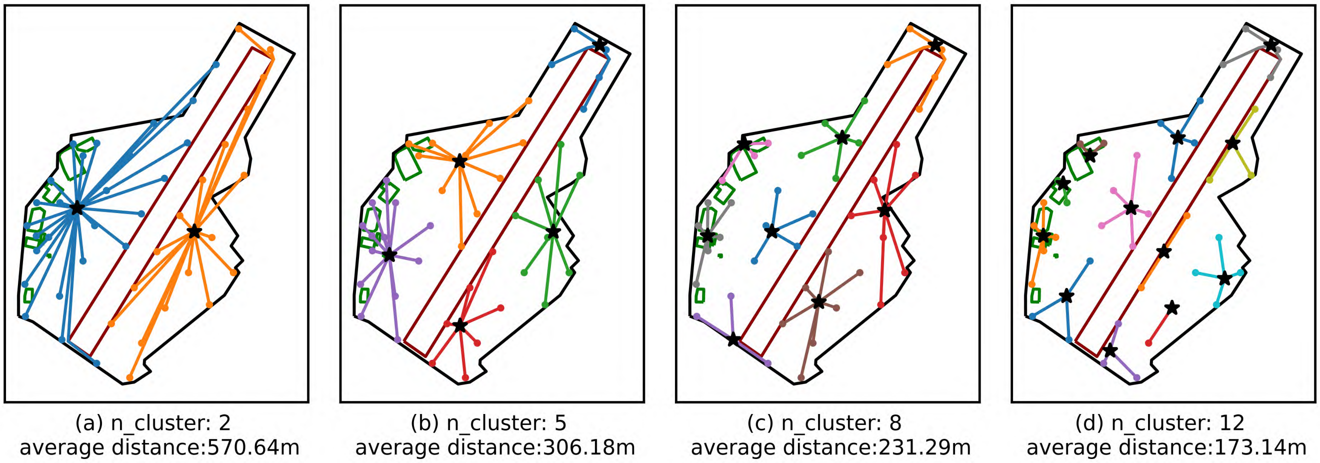


Fig. 5. Station locations for different station numbers.

auxiliary waypoints, the airpath is identified by finding the shortest path using Dijkstra's Algorithm. It ensures the drones will not penetrate no-fly zones or collide with obstacles.

We also compare our proposed graph-based formulation with the commonly used grid-partitioned formulation. The grid-partitioned formulation divides the patrolling space into a grid of equal-sized cells. Table I summarizes the advantages and disadvantages of each approach.

TABLE I
COMPARISONS BETWEEN GRID-BASED FORMULATION AND THE PROPOSED NETWORK FORMULATION

Formulation method	Advantages	Disadvantages
Grid-based formulation	easy to employ	vision can be blocked with complex obstacle boundaries; large state space for route planning especially for large mission areas; intensive decision-making epochs;
Proposed formulation	trade-off between patrolling resolution and computational complexity by adjusting the visibility range; full coverage for complex obstacle shapes; adaptability to unconnected areas; sparse decision making;	time-consuming if discretization is overrefined

III. Patrolling Route Planning

A. Route Optimization Problem

Routine patrolling over an airport requires the drones to repeatedly visit the patrolling waypoints generated in the last section. The performance metric is to minimize the waypoints' idle time, i.e., the maximum of the time elapsed between two successive visits to any waypoint. Due to the limited battery capacities, the drones have to visit the recharging nodes during the mission to avoid battery depletion.

To facilitate the following illustration, we give the definitions of *walk* and *idle time* for a set of agents $k \in \mathcal{K}$ over a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$.

Definition *(Walk)*: A walk is defined as a sequential traversal in which an agent visits the vertices. It is denoted by $W = [v_0, v_1, ..., v_n]$ such that $(v_i, v_{(i+1)}) \in \mathcal{E}$ for each $0 \leq i < n$. An infinite periodic walk can be constructed by repeating a walk infinitely.

Definition *(Maximum Idle Time)*: The maximum idle time is defined as the maximum revisit time over all target areas. Formally, given a set of infinite walks $\mathcal{W} = \{W_k, k \in \mathcal{K}\}$, we define a_i^v as the arrival time of the ith ocurance of the walks at vertex v, and d_i^v as the departure time from vertex v for the ith visit. The idle time of vertex v on walks $\mathcal{W}$ is defined the longest duration that vertex v is left unvisited, i.e.,

$$Idle^{max}(\mathcal{W}, v) = \max_i (a_{(i+1)}^v - d_i^v) \tag{1}$$

The patrolling route planning seeks to find a set of infinite walks for each drone such that the maximum idle time of each vertex is minimised. Formally, it is defined as

Problem *(Minimizing the Maximum Idle Time)*: Given a set of agents $k \in \mathcal{K}$ and a set of target areas $v \in \mathcal{V}$, the optimization problem is to find a set of walks for each agent to minimize the maximum idle time over all target areas:

$$\mathcal{W}^* = \operatorname*{argmin}_{\mathcal{W}} \max_{v \in \mathcal{V}} Idle^{max}(\mathcal{W}, v) \tag{2}$$

B. Methods

We adopt two routing strategies to address the multi-agent patrolling problem: 1) *partitioned patrolling*, the drone only patrols over the waypoints in its own cluster, where separated walks are built within the clustered waypoints; 2) *cyclic patrolling*, a long walk is found to cover all waypoints, and drones are evenly deployed along the walk.

According to [11], the optimal solution of the cyclic-based strategy is based on the shortest path of the corresponding Travel Salesman Problem (TSP). For partition-based strategy,

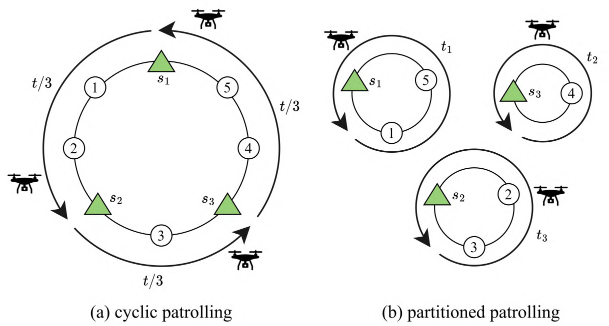


Fig. 6. Two patrolling strategies for a multi-UAV system.

given the sub-graphs induced by the partition, $\{G_1, ...G_\mathcal{K}\}$, the optimal solution in terms of the worst idleness is determined by all single-agent cyclic strategies based on the TSP of G_k. The key operation in both methods is to seek for a walk that visits all points of concern with the minimum travel time. For each walk, the drone travels through the patrolling network established in the last section.

Finding the closed path with minimum travel time is formulated as the Mixed-Integer Quadratic Programming (MIQP) problem. The objective function $\mathcal{J}$ have three terms: 1) the overall travel time through the walk, 2) the overall patrolling time over patrolling waypoints, and 3) the overall recharging time at stations:

$$\begin{aligned}\min \mathcal{J} = \sum_{i,j\in\mathbb{N}} \tau_{ij}x_{ij} + \sum_{i\in\mathbb{N}, j\in\mathbb{P}} h_j x_{ij} \\ + \sum_{i\in\mathbb{N}, j\in\mathbb{G}'} \frac{E_{max}(1-c_j)}{r} x_{ij}\end{aligned} \tag{3}$$

subject to:

$$\begin{gathered}\sum_{i\in\mathbb{N}} x_{ij} = 1, \forall i \in \mathbb{P}\cup\mathbb{G} \\ \sum_{i\in\mathbb{N}} x_{ij} = 1, \forall j \in \mathbb{P}\cup\mathbb{G} \\ \sum_{i\in\mathbb{N}} x_{ij} = \sum_{i\in\mathbb{N}} x_{ji}, \forall j \in \mathbb{G}' \\ -M(1-x_{ij}) \le c_i - c_j - \frac{h_j}{T_{max}} - \frac{\tau_{ij}}{T_{max}} \le M(1-x_{ij}), \\ \forall i \in \mathbb{P}, \forall j \in \mathbb{N} \\ -M(1-x_{ij}) \le 1 - c_j - \frac{\tau_{ij}}{T_{max}} \le M(1-x_{ij}), \\ \forall i \in \mathbb{G}', \forall j \in \mathbb{N} \\ c_{min} \le c_i \le c_{max}, \forall i \in \mathbb{N} \\ -M(1-x_{ij}) \le t_i - t_j + h_j + \tau_{ij} \le M(1-x_{ij}), \\ \forall i \in \mathbb{N}, \forall j \in \mathbb{N}/n_0\end{gathered}$$

The notations of the MIQP problem is presented in Table II.

TABLE II
NOTATIONS IN MIQP MODEL

Graph Notations	
i, j	Indexes of nodes
n_0	Starting vertex of the tour
$\mathbb{P}$	Set of patrolling waypoints
$\mathbb{G}$	Set of recharging stations
$\mathbb{G}'$	Set of recharging stations and it copies
$\mathbb{N}$	Set of all nodes, $\mathbb{N} = \mathbb{P}\cup\mathbb{G}'$
Model Input Parameters	
τ_{ij}, s	Travel time from vertex i to vertex j
h_i, s	Patrolling time at vertex i
T_{max}, s	Maximum travel time
E_{max}, J	Battery capacity
$c_{min}, \%$	The lower bound of state of charge
$c_{max}, \%$	The upper bound of state of charge
r, W	Recharging rate
Decision Parameters	
x_{ij}	Binary variable presenting if vehicle travels from vertex i to vertex j
$c_i, \%$	State of charge when vehicle arrives at vertex i
t_i, s	Time variable to eliminate sub-tours

Note that station nodes are copied in the above formulation, allowing for possible repeated visits in one walk.

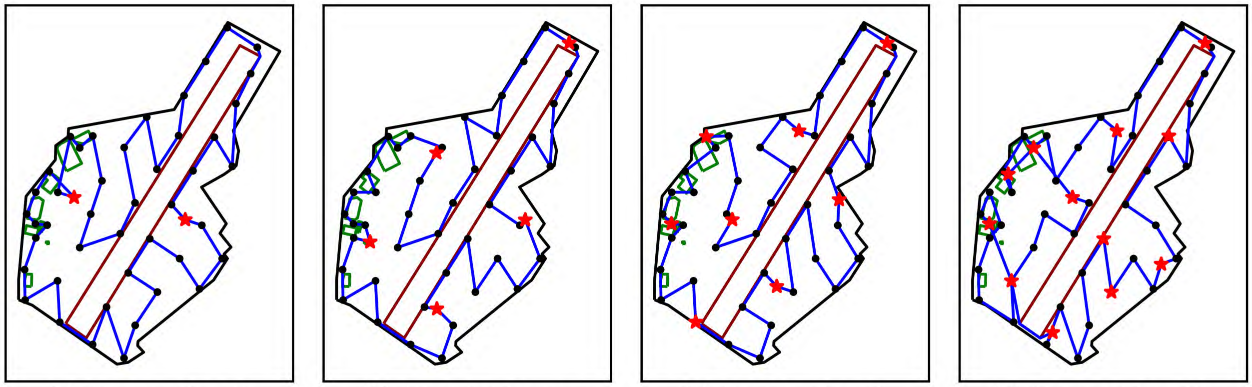
C. Results

Patrolling routes are constructed based upon the patrolling waypoints and recharging station locations generated in previous sections. CPLEX software [17] is used to solve the above MIQP problem to find the walks with the shortest time. A time limit of 500 seconds is imposed, so that a feasible route is provided if the optimal solution is not found. Simulation parameters are listed in Table III, of which drone specifications refer to those of DJI Mavic 3.

TABLE III
SIMULATION PARAMETERS

Parameter	Value (unit)
Maximum flight time, T_{max}	40 min
Battery capacity, E_{max}	277.2 k Joules
Flight speed, v	10 m/s
Flight altitude, h	20m
Patrol time over waypoints, h	1min
Recharging rate, r	65 Watt

Cyclic route solutions are depicted in Figure 7. Four plots present four cases in terms of different drone numbers (same as the station number in the context). The deployment of renewable stations follows the clustering results from the last section. Drones are evenly distributed along the route to minimize the idleness over the patrol network, which leads to an approximately linear reduction in the worst idleness of the cyclic route as the number of drones increases. The separate time cost is presented in Table IV with respect to edge travelling, patrolling, and charging. It can be seen that recharging takes more than half of the total time cost. If the UAV battery recharging rate is doubled, the maximum idle time using 12 drones can be decreased to less than 10 minutes.



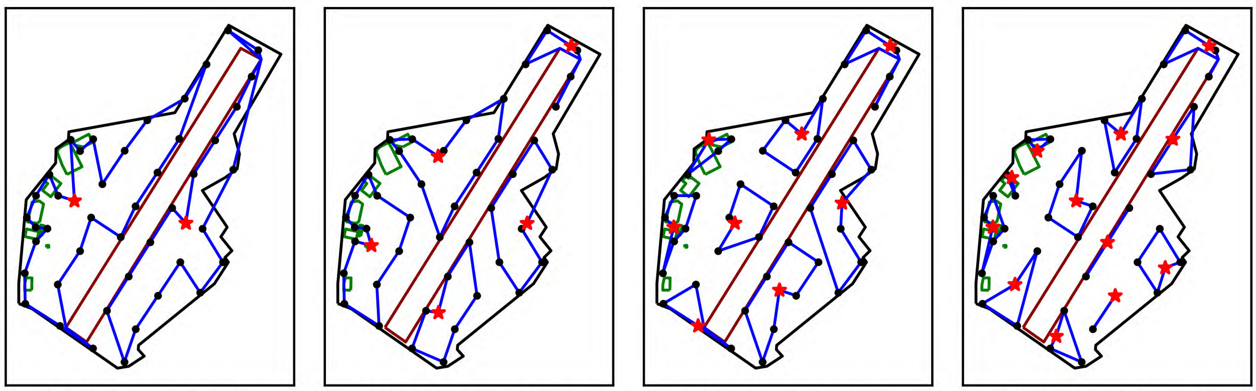
(a) drone number: 2
idle time:86.10min

(b) drone number: 5
idle time:34.35min

(c) drone number: 8
idle time:21.50min

(d) drone number: 12
idle time:14.92min

Fig. 7. Cyclic paths for different station deployments. Drones follow the blue path, and the lag time between every two consecutive drones is kept the same. Red stars indicate the locations of stations, where drones take off and get recharged.



(a) drone number: 2
max idle time:102.16min

(b) drone number: 5
max idle time:48.32min

(c) drone number: 8
max idle time:29.99min

(d) drone number: 12
max idle time:21.51min

Fig. 8. Partition-based paths for different station deployments. Drones patrol over the observation point in their own clusters, operating separately from each other.

TABLE IV
TIME COSTS FOR THE CYCLIC ROUTING METHOD

Station count	Max idleness (min)	Separate time cost (min)		
		travel time	patrol time	charge time
2	86.10	9.01	22	55.09
5	34.35	3.57	8.8	21.98
8	21.50	2.24	5.5	13.76
12	14.92	1.71	3.67	9.54

TABLE V
TIME COSTS FOR THE PARTITION ROUTING METHOD

Station count	Max idleness (min)	Separate time cost (min)		
		travel time	patrol time	charge time
2	102.16	10.17	22	57.17
5	48.32	3.82	8.8	22.43
8	29.99	2.63	5.5	14.45
12	21.51	1.66	3.67	9.47

Partition-based route solutions are depicted in Figure 8, where the drone only patrols on the waypoints of its own cluster. Compared to the cyclic routing strategy, the idle time of waypoints totally depends on how many nodes its cluster has and the distances between them. It causes a discrepancy among clusters and longer maximum idleness than those in cyclic routing solutions. The separate time costs are also listed in Table V. Same as the cyclic case, the recharging time takes the largest portion of the time consumption.

IV. CONCLUSION

This work investigated the mission planning problem for a multi-UAV system conducting routine patrol in an obstructed airport environment. The planning is started by establishing a graph-based patrolling network, from which the patrolling area is fully covered in consideration of the limited-range and blocked vision due to buildings. Locations of recharging facilities are then suggested to minimize the average distance between observing waypoints and stations. Finally, patrolling routes are generated towards the objective function to optimize the worst idleness over all waypoints for an effective persistent patrolling. This research work provides a framework and a promising solution for optimizing the multi-UAV patrol operations in complex airport environments.

For the future work, authors believe that it is desirable to develop a dynamic route optimization algorithm to enhance the robustness of route solutions against unexpected dynamic events that may affect the drone execution, such as human manipulation, the delay of patrolling, or severe weather conditions. Besides, simulations of higher-fidelity and physical flights are expected to validate and improve the proposed algorithms in the future.

ACKNOWLEDGMENT

This work was supported by the Innovate UK funded Future Flight project HADO, reference number 10024815.

REFERENCES

[1] J. Olivier, "Airport perimeter security: Finding the right fit for your airport," *Airport Magazine*, vol. 26, no. 4, 2014.

[2] S. V. Shvetsova and A. V. Shvetsov, "Ensuring safety and security in employing drones at airports," *Journal of Transportation Security*, vol. 14, pp. 41–53, 2021.

[3] J. L. Fendji Kedieng Ebongue, I. Bayaola, C. Thron, and A. Förster, "Charging stations placement in drone path planning for large space surveillance," in *CARI 2020 - Colloque Africain sur la Recherche en Informatique et en Mathématiques Apliquées*, Thies, Senegal, Oct. 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02925708

[4] I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 198–212, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18302304

[5] B. German, M. Daskilewicz, T. K. Hamilton, and M. M. Warren, "Cargo delivery in by passenger evtol aircraft: a case study in the san francisco bay area," in *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 2006.

[6] N. Venkatesh, A. P. Payan, C. Y. Justin, E. Kee, and D. Mavris, "Optimal siting of sub-urban air mobility (suam) ground architectures using network flow formulation," in *AIAA AVIATION 2020 FORUM*, 2020, p. 2921.

[7] J. Jeong, M. So, and H.-Y. Hwang, "Selection of vertiports using k-means algorithm and noise analyses for urban air mobility (uam) in the seoul metropolitan area," *Applied Sciences*, vol. 11, no. 12, p. 5729, 2021.

[8] J. Kim and S. Kim, "Data-driven approach toward airspace design for regional air mobility operations in korea," *Journal of Aerospace Information Systems*, pp. 1–13, 2023.

[9] S. K. K. Hari, S. Rathinam, S. Darbha, K. Kalyanam, S. G. Manyam, and D. Casbeer, "Optimal uav route planning for persistent monitoring missions," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 550–566, 2020.

[10] N. Rezazadeh and S. S. Kia, "A sub-modular receding horizon approach to persistent monitoring for a group of mobile agents over an urban area," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 217–222, 2019.

[11] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004.(IAT 2004).* IEEE, 2004, pp. 302–308.

[12] J. Chen, A. Baskaran, Z. Zhang, and P. Tokekar, "Multi-agent reinforcement learning for visibility-based persistent monitoring," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2021, pp. 2563–2570.

[13] "Openstreetmap," Website, 2022, retrieved from https://www.openstreetmap.org.

[14] V. Chvátal, "A combinatorial theorem in plane geometry," *Journal of Combinatorial Theory, Series B*, vol. 18, no. 1, pp. 39–41, 1975.

[15] M. C. Couto, P. J. de Rezende, and C. C. de Souza, "An exact algorithm for minimizing vertex guards on art galleries," *International Transactions in Operational Research*, vol. 18, no. 4, pp. 425–448, 2011.

[16] A. K. Tung, J. Hou, and J. Han, "Spatial clustering in the presence of obstacles," in *Proceedings 17th International Conference on Data Engineering.* IEEE, 2001, pp. 359–367.

[17] "IBM ILOG CPLEX," IBM ILOG CPLEX Optimizer, 20XX, version X.X. [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio

2023-11-10

Mission planning for a multiple-UAV patrol system in an obstructed airport environment

Liu, Ruifan

IEEE

Liu R, Shin H-S, Tsourdos A. (2023) Mission planning for a multiple-UAV patrol system in an obstructed airport environment. In: 42nd Digital Avionics Systems Conference (DASC), 1-5 October 2023, Barcelona, Spain

https://doi.org/10.1109/DASC58513.2023.10311218