

# A New Graph-Based Flight Planning Algorithm for Unmanned Aircraft System Traffic Management

Sangjun Bae  
SATM  
Cranfield University  
Bedfordshire, United Kingdom  
s.bae@cranfield.ac.uk

Hyo-Sang Shin  
SATM  
Cranfield University  
Bedfordshire, United Kingdom  
h.shin@cranfield.ac.uk

Antonios Tsourdos  
SATM  
Cranfield University  
Bedfordshire, United Kingdom  
a.tsourdos@cranfield.ac.uk

**Abstract**—To efficiently and safely provide various types of services, small Unmanned Aircraft System (sUAS) are envisioned to be integrated with other airspace users. sUAS operation types such as route network, free flight, free routing can be determined depending on services, operating environments, etc. This paper addresses a route network-based flight planning problem that includes separation considered routing and scheduling for multiple sUASs. We propose an algorithm that generates each a route and schedule for each flight from its origin to its destination to minimise each sUAS' flight time while satisfying the minimum separation requirement at all times. The algorithm consists of an inner loop and an outer loop. In the inner loop each sUAS optimises its flight plan by solving its unique shortest path problem in a decentralised way. In the outer loop one of the flights is allocated using a centralised algorithm in each outer loop. The algorithm continues until all flights are allocated. As a preliminary study, we demonstrate the proposed algorithm through case studies for “last-mile delivery”, and “first-mile delivery”. The main contributions of this paper are as follows: increasing a solution search space by solving routing and scheduling problems simultaneously with separation assurance; low computational time. The proposed algorithm can be potentially applied to airspace capacity estimation and throughput of service points.

**Keywords**—Flight planning, Unmanned Aircraft System (UAS) Traffic Management (UTM), Routing and scheduling, Drone delivery service

## I. INTRODUCTION

In the near future, small Unmanned Aircraft Systems (sUASs) are expected to be integrated with other airspace users for different purposes, e.g. shipping with sUASs might be more efficient than shipping with conventional vehicles [1], sUASs might transport medical supplies and necessities over hazardous terrain during a state of emergency [2]. There is a need for the safe operation and the efficient integration with current airspace users. Recently, the Federal Aviation Administration (FAA) published the regulation to allow the operation of sUASs in the National Airspace System [3]. To avoid interference with manned aircraft operations, the operations in the regulation limit altitude of 400 feet above ground level or within 400 feet of a structure, maximum weight to 55 lbs. (25 kilograms), the maximum ground speed of 100 mph (87 knots), etc. More details of the operational limitations are listed in [3]. A variety of international and national organisations have initiated projects on UAS Traffic

Management (UTM) system to establish policies, requirements, frameworks, and infrastructure for the safe operation and the efficient integration [4]–[7]. The projects commonly emphasise that the UTM system requires services such as ge-fencing, airspace design, routing (route planning), scheduling, separation management (spacing and sequencing), contingency management as the current Air Traffic Management (ATM) system. The UTM operations and services might be affected by the current ATM system that is the most relevant and reliable system.

Amongst the services, demands of routing, scheduling, and separation management have been raised because not only its primary aim but also these can be utilised as one of the factors determining airspace capacity and throughput of the service points. Many studies in the literature have focused on the sUAS path planning problem, and solved the problem with free flight or free routing approaches [8]–[13]. In high-density urban airspaces, the free flight-based operation may have difficulty in the safety, flight priority, and contingency operations. To consider potential interactions with the other users, and infrastructure, one possible approach is the route network-based operation. In [7], authors focus on the route networks and its capacity and throughput. The capability and throughput are analysed by using a flight planning algorithm that only considers constant aircraft speed, and is time-independent, i.e., the authors solved the routing problem with an assumption that once a sUAS occupies a route, then any other sUASs cannot traverse the occupied route. However, the approach causes unrealistic urban airspace capacity analysis results because of the conservative assumptions such as constant speed, route-based separation. It might degrade the optimality of solutions, although these have a computational advantage. In the ATM system as well, the routing, scheduling, and separation management are tackled separately or sequentially. In tactical phases, Air Traffic Controllers (ATCOs) make decisions manually based on their experience, intuition and some rules without using formally defined performance indices [14].

In this study, we modify an algorithm, which we proposed in [15] for the multiple aircraft routing and scheduling for commercial aircraft, to solve the route network-based flight planning problems (routing and scheduling) for multiple sUASs

in high-density urban airspaces also called very low level (VLL) airspace. The proposed algorithm generates each sUAS' flight plan iteratively. At each iteration, each sUAS finds its flight plan in a decentralised way by solving a graph-based problem we formulate (inner loop), and one of the sUASs is allocated using centralised algorithms (outer loop). The outer loop continues until all sUASs are planned. For the inner loop, we propose a novel weight scheme to minimise each sUAS' flight time that satisfies the time-based separation requirement. By assigning the weights in the graph, we can consider a few practical factors such as feasible speed ranges of the sUASs, different separation requirements, etc. Then, each sUAS has its unique VLL airspace graph, and its optimal route obtained by using any well-known shortest path algorithm implies each sUAS' flight plan. We focus not only proposing the flight planning algorithm for multiple sUASs over the route networks, but also a comparison between two different outer loop algorithms. The algorithm is demonstrated on the last mile delivery (1-to- $\mathcal{M}$ ) and the first mile delivery ( $\mathcal{N}$ -to-1) cases. Results of the two case studies will demonstrate the multiple flights flight planning and provide insights into establishing detailed requirements for the sUAS operations in the high-density VLL airspace. Throughout this preliminary study, we expect that more efficient route network-based flight plannings, and capacity estimation for the high-density VLL airspace.

The rest of this paper is organised as follows. Section II formulates the problem for a single sUAS flight planning. Section III describes the proposed algorithm for multiple sUASs' flight planning problem. As a preliminary case study, over-road route networks for "last-mile delivery" and "first-mile delivery" are given and solved in Section IV. Section V concludes this study and Section VI discusses the proposed algorithm and further work.

## II. PROBLEM STATEMENT FOR SINGLE sUAS FLIGHT PLANNING

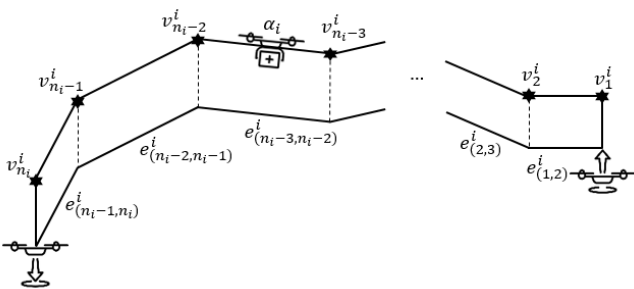


Figure. 1: A series of linear flight segments (sUAS  $\alpha_i$ 's route from its origin  $v_1^i$  to destination  $v_{n_i}^i$ )

Our focus is on flight planning a finite set  $\mathcal{A}^+ = \{\alpha_1, \alpha_2, \dots, \alpha_i\}$  of sUASs in the high-density VLL airspace where each sUAS  $\alpha_i \in \mathcal{A}^+$  traverses from its origin to its destination. The flight planning, in this context, means finding routes, its speed profile (schedule), and departure or an arrival

sequence while satisfying the separation requirement from a finite set  $\mathcal{A}^-$  of planned sUASs.

Main assumptions we make for this study are as follows: each flight path is composed of a series of linear flight segments as shown in Fig.1; each sUAS traverses along each segment at a constant speed; each UASs can accurately follow the route without deviation; a time-based separation concept is utilised; uncertainties produced by external sources are neglected such as adverse weather; flight plans are shared between sUASs; the hovering manoeuvre is not considered.

### A. Time-Based Separation

Conflict detection is activated when separation of two sUASs is less than a minimum separation criterion. In this study, we apply the time-based separation concept instead of the widely used distance-based separation concept to stabilise the spacing between the sUASs. By satisfying the separation assurance flight time for each waypoint rather than adjusting the distance between sUASs, we can obtain results that satisfy the minimum separation at airways as well as at waypoints at all times. The concept allows time-based separation to be satisfied at both merging points and crossing points as described in Section II-B.

For ease of understanding, we describe the concept with an example. At the time  $t_0$  of Fig. 2,  $\alpha_i$  and  $\alpha_{i'}$  fly toward the  $v_C$  through the same merging point,  $v_Y$ , at the speeds  $s_1^\alpha$  and  $s_1^{i'}$ , respectively. The time  $t_1$  when  $\alpha_{i'}$  just passes through  $v_Y$  at the speed of  $s_1^{i'}$  is stored at the merging point  $T(v_Y)$ , and no sUAS can pass through this point for  $t_{SP}$  seconds before and after  $t_1$ . The time  $t_2$  is stored in  $T(v_Y)$  in the same way. Then,  $\alpha_i$  and  $\alpha_{i'}$  traverse toward  $v_C$  at the speeds of  $s_2^i$  and  $s_2^{i'}$ , respectively. If  $\alpha_i$  and  $\alpha_{i'}$  fly from  $v_Y$  to  $v_C$  at the speeds of  $s_2^i$  and  $s_2^{i'}$ , respectively, the separation between two sUASs will always be met the separation requirement or greater than that on the route between  $v_Y$  and  $v_C$ . At the time  $t_3$  of Fig. 2,  $\alpha_i$  and  $\alpha_{i'}$  fly at the same speed while maintaining the minimum separation. The time data at the waypoint  $v_Y$  is as follows:

$$T(v_Y) = \{0s, 120s, 210s\}. \quad (1)$$

The time data at each waypoint obtained such as (1) is used to calculate the optimisation problem that will be described in Section.III.

### B. Mathematical Modelling

For the operational purposes, the route network-based operation is one of the concepts for the UTM system [16]. Thus, we model the high-density VLL airspace for the UTM system as a directed graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ , called an airspace graph. In the airspace graph, each vertex  $v \in \mathcal{V}$  is a waypoint candidate to be traversed through Euclidean space of dimension two or three. Each edge  $e \in \mathcal{E}$ , corresponds a rectifiable curve, is a route between some pair of waypoints in the airspace graph. In this model, a feasible flight path of sUAS  $\alpha_i$  in the airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V})$  is defined as follows:

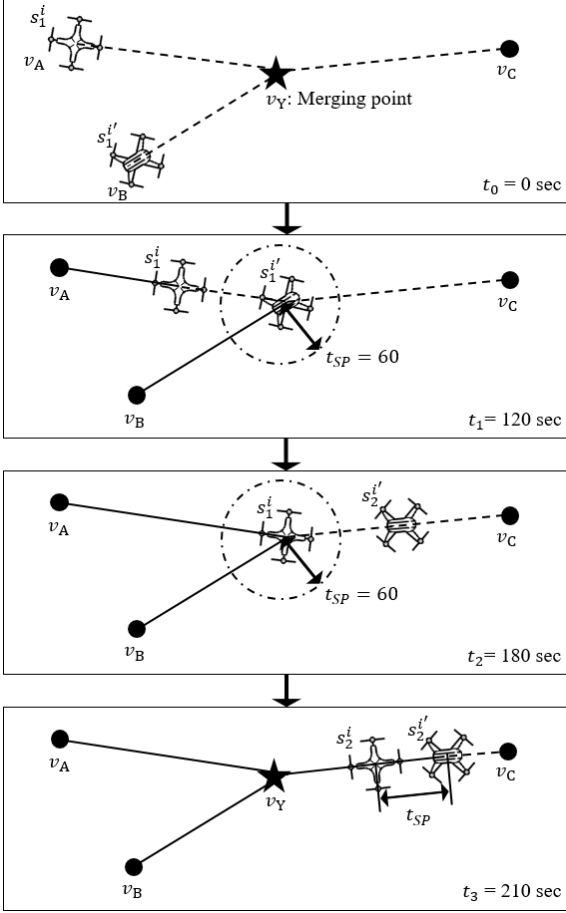


Figure 2: Time-based separation concept for merging points ( $\alpha_i$  and  $\alpha_i'$  fly from its origin  $v_A$  and  $v_B$  to  $v_C$  through a merging point  $v_Y$  at the speed  $\{s_1^i, s_2^i\}$  and  $\{s_1^{i'}, s_2^{i'}\}$ , respectively. The superscript and subscript of  $s$  are the sUAS index and the segment index, respectively)

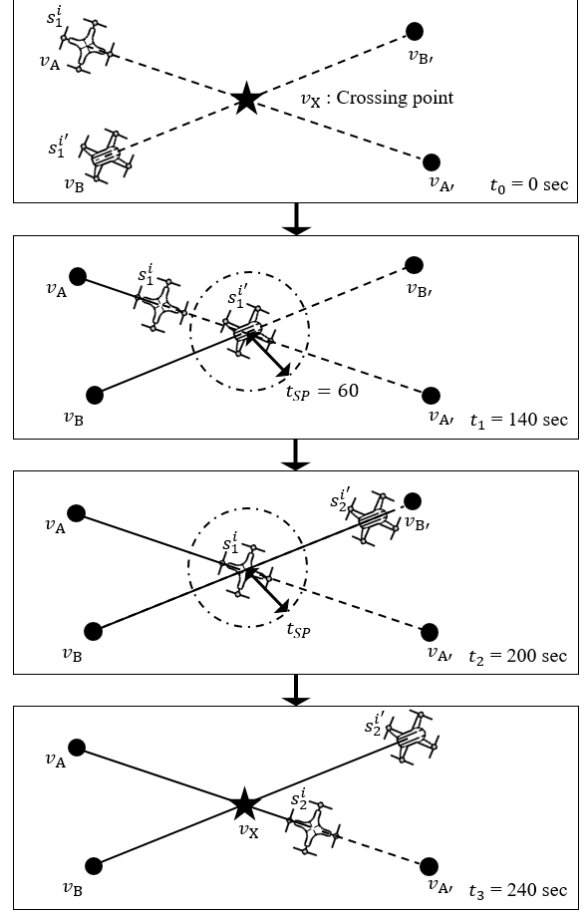


Figure 3: Time-based separation concept for crossing points ( $\alpha_i$  and  $\alpha_i'$  fly from its origin  $v_A$  and  $v_B$  to its final point  $v_{A'}$  and  $v_{B'}$  through a crossing point  $v_X$  at the speed  $\{s_1^i, s_2^i\}$  and  $\{s_1^{i'}, s_2^{i'}\}$ , respectively. The superscript and subscript of  $s$  are the sUAS index and the segment index, respectively)

**Definition 1. (D1)** As given in the airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ , sUAS  $\alpha_i$  to be routed in  $\mathcal{G}$ , and the initial waypoint  $v_1^i \in \mathcal{V}$  and the final waypoint  $v_{n_i}^i \in \mathcal{V}$ , a flight path denoted by  $p(\alpha_i)$  in  $\mathcal{G} = (\mathcal{E}, \mathcal{V})$  is defined by a sequence of waypoints.

Thus, in the airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ , there can be an abundance of flight path candidates denoted by  $\mathcal{C}$  that satisfy the conditions as described in **D1**. Through **D1**, each flight path candidate  $p(\alpha_i) \in \mathcal{C}$  can be given a corresponding flight path (a set of waypoints) as follows:

$$p(\alpha_i) : v_1^i, v_2^i, \dots, v_{n_i}^i. \quad (2)$$

For each sUAS' route as given in (3), there is a set of routes connecting waypoints through this flight path, which is as follows:

$$\mathcal{E}(p(\alpha_i)) : e_{(1,2)}^i, e_{(2,3)}^i, \dots, e_{(n_i-1, n_i)}^i. \quad (3)$$

Through this formulation, for each flight path candidate, it is assumed that a performance index for a set of airways  $\mathcal{E}(p(\alpha_i))$  can be quantified as a set of positive numeric weight values, as follows:

$$\mathcal{W}(p(\alpha_i)) : w_1^i, w_2^i, \dots, w_{n_i-1}^i. \quad (4)$$

Then, the airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V})$  is transformed into a weighted directed graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$  by assigning a weight to each route for all flight routes. In the airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$ , each flight path can be estimated by summing all weights of  $\mathcal{W}(p(\alpha_i))$ , as follows:

$$\mathcal{T}(p(\alpha_i)) = \sum_{j=1}^{n_i-1} w_j^i. \quad (5)$$

Based on the airspace graph,  $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$ , the routing problem that minimises a performance index can be defined as follows:

**Definition 2. (D2)** Given an airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$  and corresponding all flight path candidates  $\mathcal{C}$ , the routing problem is defined as finding a flight path (or a sequence of waypoints) such that

$$p^*(\alpha_i) = \operatorname{argmin}_{p(\alpha_i) \in \mathcal{C}} \mathcal{T}(p(\alpha_i)). \quad (6)$$

The optimal flight path  $p^*(\alpha_i)$  can be found by using the well-known shortest path algorithms such as Dijkstra's algorithm, A\* algorithm, the brute-force search [17]–[19]. Although the optimal route  $p^*(\alpha_i)$  can be obtained according to **D2**, the optimality of the flight path might be disturbed in the following stages: scheduling stage and conflict resolution stage to satisfy the separation requirement. Such a sequential approach can not only cause the optimality issues but also require additional works for scheduling, and sequencing and spacing.

Our objective of solving the flight planning problem is to maximise airspace capacity, which can be achieved by minimising the flight time of each sUAS. We also intend to minimise the disturbances in the sequential approaches by solving the routing, scheduling, and sequencing and spacing problems simultaneously. Our main idea for achieving the objective is to assign a weight to each route (edge) of the airspace graph where the weight is expressed as sUAS' speed or the flight time. Therefore, each sUAS' speed profile also can be obtained by finding the optimal route in the airspace graph. We set the flight time of each sUAS as a weight  $w \in \mathcal{W}$  to each route  $e \in \mathcal{E}$  of the airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$ . Note that, the flight time and speed of each sUAS are mutually interchangeable using the geographic data in the airspace graph. Each sUAS' speed profile is determined within its feasible speed range while satisfying the separation requirement from planned sUASs.

Another issue we have addressed is to satisfy the separation requirement between each pair of sUASs. To assign the flight time that satisfies the separation requirement to each edge of the airspace graph, we need data of the planned sUASs' flight time at each waypoint as discussed in Section.II-A. The time data will be included in the airspace graph  $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W}, T)$  and be used to calculate weights of the airspace graph. By finding the optimal route of the airspace graph  $\mathcal{G}$ , then, we can obtain the schedule of the sUAS while satisfying the separation requirement simultaneously. Each sUAS  $\alpha_i \in \mathcal{A}^+$  might have different weights  $\mathcal{W}_i$  because of different specifications such as each sUAS' feasible speed range or the separation requirement as well as its initial and final positions. Thus, each sUAS might have its unique airspace graph  $\mathcal{G}_i$  as shown in Fig.4.

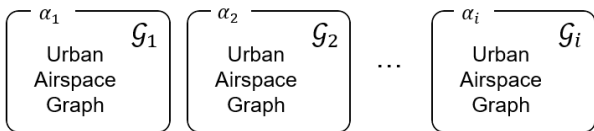


Figure. 4: Unique urban airspace graphs for each sUAS

### C. Calculating weights on $\mathcal{G}_i$ suitable for separation-compliant speed profiles

In this subsection, we propose a weight scheme to be applied to the airspace graphs  $\mathcal{G}_i$  to fulfil our objectives. An optimisation problem for calculating the proposed weights is as follows:

$$\min \sum_{j=1}^{n_i-1} \frac{d_j^i}{s_j^i} \quad (7)$$

$$\text{s.t.} \quad \frac{d_1^i}{s_1^i} + \frac{d_2^i}{s_2^i} + \dots + \frac{d_{n_i-1}^i}{s_{n_i-1}^i} \geq \max T(v_{n_i}^i) + t_{SP} \quad (8)$$

$$\frac{d_1^i}{s_1^i} + \frac{d_2^i}{s_2^i} + \dots + \frac{d_{n_i-2}^i}{s_{n_i-2}^i} \geq \max T(v_{n_i-1}^i) + t_{SP} \quad (9)$$

⋮

$$\frac{d_1^i}{s_1^i} + \frac{d_2^i}{s_2^i} \geq \max T(v_3^i) + t_{SP} \quad (10)$$

$$\frac{d_1^i}{s_1^i} \geq \max T(v_2^i) + t_{SP} \quad (11)$$

$$s_{\min}^i \leq s_1^i, s_2^i, \dots, s_{n_i-1}^i \leq s_{\max}^i \quad (12)$$

where  $s_{\min}^i$  and  $s_{\max}^i$  are minimum and maximum speed of sUAS  $\alpha_i$ , respectively. Time  $t_{SP}$  is the minimum time-based separation requirement between sUASs at waypoints as shown in Fig.2. The objective function (7) is to minimise the flight time. The constraints from (8) to (11) are for satisfying the minimum separation requirement at each waypoint. In Fig.1, for instance, if  $\mathcal{E}(p(\alpha_i))$  consists of  $n_i - 1$  segments for sUAS  $\alpha_i$ , each segment requires the flight time that satisfies the separation requirement. Each of the left hand side of the constraints from (8) to (11) is the flight time to the each segment, which must be greater than or equal to the time that satisfies the separation requirement. For the constraints, flight distances  $d$  between waypoints connected by airways and the time data  $T$  stored in  $\mathcal{G}_i$  are required. The time data  $T(v_{n_i}^i)$  indicates the flight time that sUAS  $\alpha_i$  fly over waypoint  $v_{n_i}^i$ . The time data is stored in  $T$  for each waypoint  $v \in \mathcal{V}$ , and  $T$  is updated every time sUAS is allocated and shared with  $\forall \alpha_i \in \mathcal{A}^+$ . Then, the optimisation problem we formulated can be solved by using any well-known nonlinear programming algorithm. In this study, we utilise an interior point method.

In the optimisation problem, decision variables as shown in (13) are constant speeds for flight segments, which will be transformed into the flight time and assigned into weights  $\mathcal{W}_{i,t}$ . Therefore, a solution of the airspace graph  $\mathcal{G}_i = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}, T)$  can simultaneously provide a route and schedule satisfying the minimum separation at all time. A speed profile shown in (13) can be derived from the solution and used as a separation-compliant speed profile. The permissible speed range of each sUAS is a function of the type of the sUAS.

$$s(i) : s_1^i, s_2^i, \dots, s_{n_i}^i. \quad (13)$$

We also construct flight distance-based weights  $\mathcal{W}_d$  on  $\mathcal{G}_i$  to reflect the flight distance  $d$  as a second criterion. The weights are necessary to prioritise for multiple sUASs' flight planning problem when two or more sUASs arrive at the waypoint at the same time and determining the departure sequence, more details about the priority are in Section.III. The single

sUAS flight planning problem that motivated this study can be formulated as follows:

**Problem 1. (P1)** Given an airspace graph  $\mathcal{G}_i = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}, \mathcal{W}_d, T)$ , sUAS  $\alpha_i$  to be routed and scheduled in  $\mathcal{G}_i$ , and its origin  $v_1^i \in \mathcal{V}$  and destination  $v_{n_i}^i \in \mathcal{V}$  reachable from the origin, construct a route  $p(\alpha_i)$  and a speed profile  $s(\alpha_i)$ ,  $\forall \alpha_i \in \mathcal{A}^+$  such that

- the separation requirement is satisfied from the planned sUASs,
- the speed profile of each sUAS must be within its feasible speed range,
- the airspace graphs are updated every time when sUAS  $\alpha_i \in \mathcal{A}^+$  is planned.

### III. FLIGHT PLANNING ALGORITHM FOR MULTIPLE SUASS

---

**Algorithm 1: (A1)** Flight planning algorithm for multiple sUASs

---

**Input:**  $v_1^i, v_{n_i}^i \in \mathcal{V}, \forall \alpha_i \in \mathcal{A}^+$ , airspace information  
**Output:**  $p(\alpha_i), s(\alpha_i), \forall \alpha_i \in \mathcal{A}^+$

- 1  $k = 1$
- 2 generate  $\mathcal{G}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}^k, \mathcal{W}_d, T^k), \forall \alpha_i \in \mathcal{A}^+$
- 3 **while**  $\mathcal{A}^+ \neq \emptyset$  **do**
- 4     find  $h$ -shortest routes for  $\mathcal{G}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_d, T^k), \forall \alpha_i \in \mathcal{A}^+$  (using Yen's algorithm);
- 5     generate  $\bar{\mathcal{G}}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}^k, \mathcal{W}_d, T^k), \forall \alpha_i \in \mathcal{A}^+$  that only considers the best  $h$  routes,  $\forall \alpha_i \in \mathcal{A}^+$ ;
- 6     find  $p^*(\alpha_i)$  of  $\bar{\mathcal{G}}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}^k, T^k), \forall \alpha_i \in \mathcal{A}^+$ ,  $\forall \alpha_i \in \mathcal{A}^+$  (using Dijkstra's algorithm);
- 7      $\alpha_i^* \leftarrow \operatorname{argmin}_{\alpha_i \in \mathcal{A}^+} p^*(\alpha_i)$  (using the FCFS algorithm or the LCFS algorithm);
- 8     **if** *There are more than two  $\alpha_i^*$  exist* **then**
- 9         find  $p^*(\alpha_i)$  of  $\bar{\mathcal{G}}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_d, T^k)$  amongst them (using Dijkstra's algorithm);
- 10          $\alpha_i^\dagger \leftarrow \operatorname{argmin}_{\alpha_i^*} p^*(\alpha_i)$  (using the FCFS algorithm or the LCFS algorithm) ;
- 11         allocate  $\alpha_i^\dagger$ ;
- 12         remove  $\alpha_i^\dagger$  from  $\mathcal{A}^+$  and store  $\alpha_i^\dagger$  in  $\mathcal{A}^-$  with  $p^*(\alpha_i)$  and  $s(p^*(\alpha_i))$ ;
- 13         share  $\mathcal{A}^-$  with  $\forall \alpha_i \in \mathcal{A}^+$ ;
- 14     **else**
- 15         allocate  $\alpha_i^*$ ;
- 16         remove  $\alpha_i^*$  from  $\mathcal{A}^+$  and store  $\alpha_i^*$  in  $\mathcal{A}^-$  with  $p^*(\alpha_i)$  and  $s(p^*(\alpha_i))$ ;
- 17         share  $\mathcal{A}^-$  with  $\forall \alpha_i \in \mathcal{A}^+$ ;
- 18     **end**
- 19      $k = k + 1$ ;
- 20     update  $\mathcal{G}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}^k, \mathcal{W}_d, T^k), \forall \alpha_i \in \mathcal{A}^+$  ;
- 21 **end**

---

Section II focuses on the single sUAS flight planning problem (inner loop) which can be formulated by assigning the proposed separation satisfied flight time-based weights to the

airspace graph. By doing so, we can consider practical factors such as feasible speed ranges of the sUASs, and different separation requirements. In this study, we solve the **P1** using Dijkstra's algorithm. However, the flight time-based weights of each sUAS only satisfy the separation requirement from already allocated sUASs.

In this section, we describe the outer loop of our algorithm which determines the order of arrival and departure of multiple sUASs in an iterative way. At each iteration, each unallocated sUAS finds its flight plan by solving the **P1** (inner loop) and one of the unallocated sUASs is allocated using the First Come First Served (FCFS) algorithm or the Last Come First Served (LCFS) algorithm (outer loop). The outer loop algorithm plays a important role to increase the airspace capacity. Through the case studies, we will show the difference between the FCFS algorithm and the LCFS algorithm according to the type of case.

Once inputs of the algorithm are given, the algorithm runs the flight planning process of the sUASs until  $\mathcal{A}^+$  is empty (Line 3). The algorithm generates first the airspace graphs  $\mathcal{G}_i^k, \forall \alpha_i \in \mathcal{A}^+$  (Line 2). Then, each sUAS has its unique airspace graph that contains flight time-based weights  $\mathcal{W}_{i,t}^k$  determining the optimal route and its flight time. In order to reduce the complexity of the process that calculates the weights, the algorithm selects the  $h$ -shortest routes for  $\mathcal{G}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_d, T^k), \forall \alpha_i \in \mathcal{A}^+$  using Yen's algorithm, and generates  $\bar{\mathcal{G}}_i^k$  only considering the  $h$ -shortest routes (Line 4-5) [20]. The optimal route  $p^*(\alpha_i)$  is obtained by using Dijkstra's algorithm,  $\forall \alpha_i \in \mathcal{A}^+$  (Line 6) [17]. The sUAS  $\alpha_i^*$  is obtained by using the FCFS algorithm or the LCFS algorithm (Line 7), and allocated (Line 15). The sUAS  $\alpha_i^*$  is removed from  $\mathcal{A}^+$ , and stored in  $\mathcal{A}^-$  with its route  $p^*(\alpha_i)$  and speed profile  $s(p^*(\alpha_i))$  (Line 16). In the case of that more than two sUASs are arriving to the same destination (Line 8), the flight distance-based weights  $\mathcal{W}_d$  are utilised to allocate a sUAS amongst them (Line 9-13). In any case, a sUAS is allocated and its flight data is shared with sUAS  $\alpha_i, \forall \alpha_i \in \mathcal{A}^+$ . Based on the data each airspace graph  $\mathcal{G}_i^k$  is updated,  $\forall \alpha_i \in \mathcal{A}$  (Line 20).

We illustrate an example in Fig. 5 for a better understanding of the outer loop concept of the proposed algorithm with the FCFS algorithm as the outer loop algorithm. In the first iteration, each sUAS generates its airspace graph  $\mathcal{G}_i^1, \forall \alpha_i \in \mathcal{A}^+$ . Through the inner loop process, each sUAS's  $p^*(\alpha_i)$  is determined. The first come sUAS  $\alpha_2$  is allocated using the FCFS algorithm. In the second iteration, sUAS  $\alpha_A$  is allocated through the same process as the first. In the  $(A)^{th}$  iteration, sUAS  $\alpha_1$  is allocated finally. Although in the example above we assume that the first come sUAS has the priority to be allocated, the criteria or the outer loop algorithm to determine the priority can be changed.

Advantages of the proposed algorithm are as follows: (a) enlarging the solution searching space by searching a flight path and schedule of each sUAS simultaneously; (b) providing separation-compliant speed profiles that satisfy each sUAS's performance; (c) applicable for various types of route net-

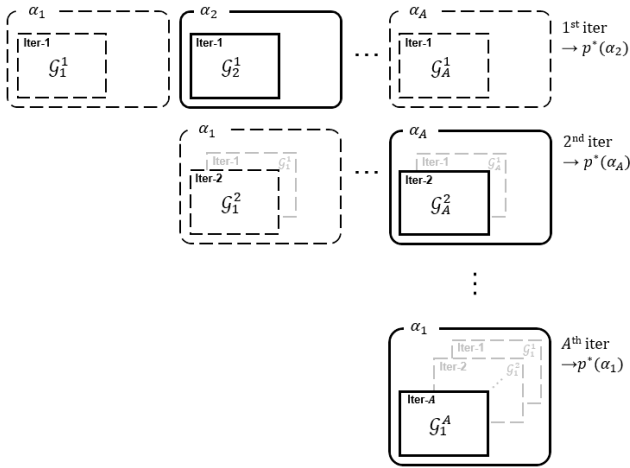


Figure 5: Iteratively generated urban airspace graph concept of **A1**



Figure 6: Oldbrook, Milton Keynes, United Kingdom (need Google earth reference)

works; (d) applicable for airspace capacity estimation.

#### IV. CASE STUDY

Last-mile delivery and first-mile delivery have been chosen to demonstrate the proposed algorithm. Scenarios are based on a town ‘Oldbrook’ in Milton Keynes, which is one of the well-planned cities in United Kingdom as shown in Fig.6. Two-dimensional infrastructure data of the town is obtained from Google Earth Pro. Both scenarios have an area size of 0.98 km<sup>2</sup>. For this two cases, we construct two route networks over roads which are similar to the en-route airspace for commercial aircraft. In this route networks, two layers of nodes are set above the roads at the height of 20 meters and 30 meters for eastbound and westbound as shown in Fig.7 and Fig.8, respectively. Each route network consists of 68 points (1 retail point (red pin) and 67 service points (green pins)) as shown in Fig.6, and 103 directed airways as shown in Fig.7 and Fig.8. Thirty sUASs traverse from each sUAS’ start point to each sUAS’ final point. Both cases consider only one type of the sUAS. The sUAS type can be changed based on their missions and operators. Also, it is assumed that the separation requirement including landing and take-off during the flight between sUASs is 5 seconds as shown in Table I.

TABLE I  
sUAS SPECIFICATION & SEPARATION REQUIREMENTS

Permissible Speed Range [km/h]	[5 - 25]
Horizontal Separation [sec]	5
Departure Separation [sec]	5
Arrival Separation [sec]	5

The outer loop of **A1** to allocate sUASs’ departure and arrival order is one of the factors that determine airspace capacity, throughput, etc. For the outer loop we present two simulations with both the FCFS algorithm and the LCFS algorithm for both the last mile delivery case (1-to- $\mathcal{M}$ ) and the first mile delivery ( $\mathcal{N}$ -to-1) case to demonstrate the effect of the outer loop algorithm as well as the efficiency of **A1**.

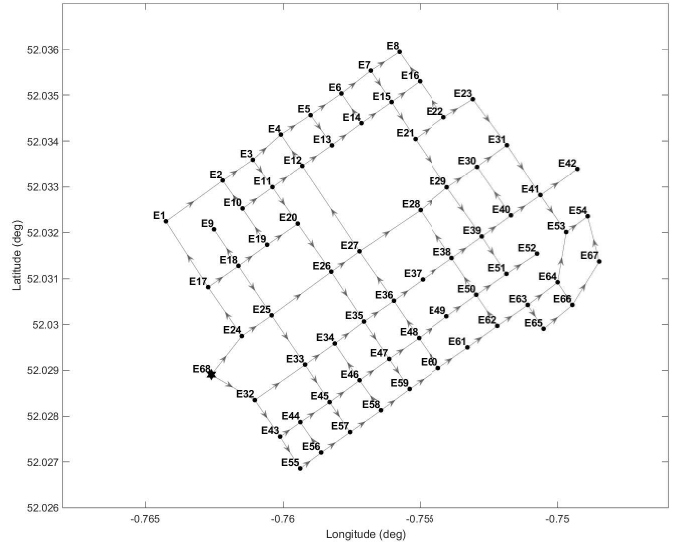


Figure 7: Eastbound, 1 retail point (E68) and 67 service points (E1 ~ E67) and 103 directed airways

##### A. Last Mile Delivery (1-to- $\mathcal{M}$ )

For the last mile delivery both the FCFS algorithm and the LCFS algorithm to allocate sUASs’ departure sequence are applied (for the latter case Line 7 and Line 10 of **A1** are replaced to the LCFS algorithm). Table II and III show each sUAS’ flight details when the FCFS algorithm and the LCFS algorithm are applied, respectively. Both of the outer loop algorithms generate 30 sUASs’ flight routes and its schedules while satisfying the minimum separation requirements. Additionally, from comparison results between both of the algorithms as shown in Table IV and Fig.9, we can see that the mission completion time of the LCFS algorithm case is 34.5% more faster than the FCFS algorithm case. In the latter case, also, each sUASs’ mission completion time is more evenly distributed than the FCFS algorithm.

TABLE II  
FLIGHT PLANNING RESULTS OF 30 sUASs FOR THE LAST MILE DELIVERY (OUTER LOOP: FCFS ALGORITHM)

sUAS ID	Start point	Final point	Waypoint sequence (arrival time at each waypoint [sec]) speed profile [km/h]
⋮	⋮	⋮	⋮
$\alpha_4$	E68	E4	E68(180) - E24(197) - E17(217) - E1(244) - E2(268) - E3(280) - E4(293) 25 - 25 - 25 - 25 - 25 - 25
$\alpha_5$	E68	E5	E68(220) - E24(237) - E17(257) - E1(284) - E2(308) - E3(320) - E4(333) - E5(345) 25 - 25 - 25 - 25 - 25 - 25
⋮	⋮	⋮	⋮
$\alpha_{22}$	E68	E36	E68(130) - E32(147) - E33(169) - E34(181) - E35(193) - E36(206) 25 - 25 - 25 - 25 - 25
$\alpha_{23}$	E68	E38	E68(170) - E32(188) - E33(209) - E34(221) - E35(234) - E36(246) - E37(258) - E38(271) 25 - 25 - 25 - 25 - 25 - 25
⋮	⋮	⋮	⋮

TABLE III  
FLIGHT PLANNING RESULTS OF 30 sUASs FOR THE LAST MILE DELIVERY (OUTER LOOP: LCFS ALGORITHM)

sUAS ID	Start point	Final point	Waypoint sequence (arrival time at each waypoint [sec]) speed profile [km/h]
⋮	⋮	⋮	⋮
$\alpha_4$	E68	E4	E68(110) - E24(127) - E17(147) - E1(173) - E2(197) - E3(210) - E4(223) 25 - 25 - 25 - 25 - 25 - 25
$\alpha_5$	E68	E5	E68(70) - E24(87) - E17(107) - E1(133) - E2(157) - E3(170) - E4(183) - E5(195) 25 - 25 - 25 - 25 - 25 - 25
⋮	⋮	⋮	⋮
$\alpha_{22}$	E68	E36	E68(160) - E32(177) - E33(198) - E34(211) - E35(223) - E36(236) 25 - 25 - 25 - 25 - 25
$\alpha_{23}$	E68	E38	E68(120) - E32(137) - E33(158) - E34(171) - E35(183) - E36(196) - E37(208) - E38(221) 25 - 25 - 25 - 25 - 25 - 25
⋮	⋮	⋮	⋮

TABLE IV  
COMPARISON BETWEEN THE FCFS ALGORITHM AND THE LCFS ALGORITHM FOR THE LAST MILE DELIVERY

	FCFS algorithm	LCFS algorithm
Total flight time [sec]	7035.1	7031.1
Total flight distance [km]	19.3	19.3
Mission completion time [sec]	468.7	307.0

### B. First Mile Delivery ( $\mathcal{N}$ -to-1)

We consider the worse case of the first mile delivery, which is that every sUAS departs at the same time. For this case, both the FCFS algorithm and the LCFS algorithm to allocates sUASs' departure sequence are applied (for the latter case Line 7 and Line 10 of **A1** are replaced to the LCFS algorithm). Note that, due to various factors such as the number of sUASs, feasible speed ranges, and minimum separation requirements, **A1** may not be able to generate flight plans for the sUAVs satisfying the minimum separation requirements for the  $\mathcal{N}$ -to-

1 case. In this worst case of the first mile delivery, especially, **A1** with the LCFS algorithm for the outer loop cannot generate separation assured flight plans while the FCFS algorithm generates each sUAS' flight plan that satisfies the minimum separation requirement. Table V shows each sUAS' flight details when the FCFS outer loop algorithm is applied for the given start point and final point of each sUAS.

The computation time of the algorithm for this example on a Windows 10 OS 3.4 GHz desktop computer with 16 GB RAM is shown in Table VI. The computation time will be significantly decreased if we solve each sUAS' inner loop in a decentralised way, which is assumed as a centralised manner in this study.

## V. CONCLUSIONS

In this paper, the flight planning algorithm for multiple sUASs is proposed with the route network over roads. The algorithm generates routes and schedules of the sUASs to minimise each sUAS' flight time in the given route network.

TABLE V  
FLIGHT PLANNING RESULTS OF 30 sUASs FOR THE FIRST MILE DELIVERY (OUTER LOOP: FCFS ALGORITHM)

UAS ID	Start point	Final point	Waypoint sequence (arrival time at each waypoint [sec]) speed profile [km/h]
⋮	⋮	⋮	⋮
$\alpha_4$	W4	W68	W4(0) - W3(13) - W2(25) - W1(49) - W17(76) - W24(96) - W68(113) 25 - 25 - 25 - 25 - 25 - 25
$\alpha_5$	W5	W68	W5(0) - W4(13) - W3(26) - W2(39) - W10(50) - W19(65) - W18(77) - W17(91) - W24(111) - W68(128) 24 - 25 - 25 - 25 - 25 - 25 - 24 - 25 - 25
⋮	⋮	⋮	⋮
$\alpha_{22}$	W36	W68	W36(0) - W35(13) - W34(25) - E33(39) - W32(60) - W68(77) 25 - 25 - 24 - 25 - 25
$\alpha_{23}$	W38	W68	W38(0) - W37(15) - W36(28) - E35(41) - W34(56) - E33(69) - W32(90) - W68(107) 22 - 24 - 24 - 22 - 25 - 25 - 25
⋮	⋮	⋮	⋮

TABLE VI  
COMPUTATIONAL TIME FOR THE CASES [SEC]

<b>Last mile delivery with FCFS algorithm</b>	19.8
<b>Last mile delivery with LCFS algorithm</b>	12.1
<b>First mile delivery with FCFS algorithm</b>	21.8

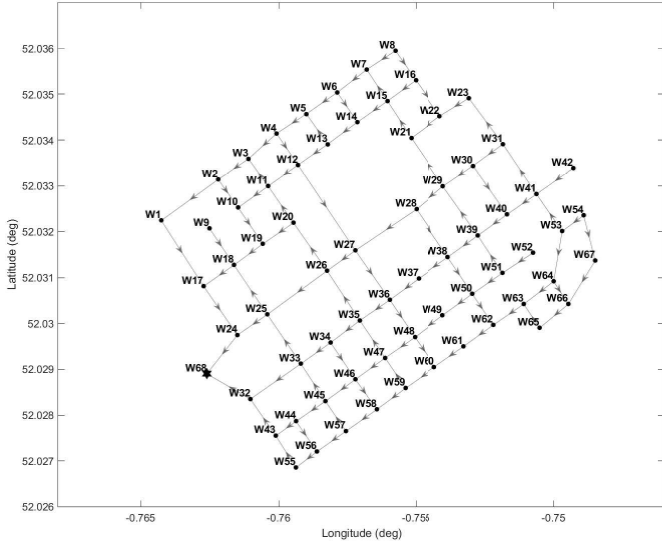


Figure 8: Westbound, 1 retail point (W68) and 67 service points (W1 ~ W67) and 103 directed airways

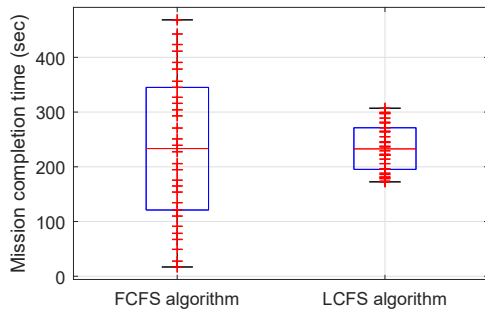


Figure 9: Standard deviation (mission completion time) comparison between the FCFS algorithm and the LCFS algorithm for the outer loop of A1

The resulting flight plans satisfy the minimum separation requirement between the sUASs at all times. The algorithm searches each sUAS' route and its schedule simultaneously, which means the flight plans are found in a large solution space. Also, each sUAS' flight planning can be done in parallel, which significantly may reduce the computation time. Through the last mile delivery case and the first mile delivery case, we show that the practical and detailed flight plans that can support to analyse the route network capacity and to establish requirements for safe operation. This algorithm positively will benefit the backbone of designing the concept of the UTM system to be integrated into the ATM system. However, additional dynamic constraints are required to improve the algorithm.

## VI. DISCUSSION

To make the proposed algorithm a practical tool for flight planning for multiple sUASs, more research is required to considering factors such as flight order determining methods, hovering manoeuvre in a highly congested airspace, uncertainties such as wind effect, dynamical behaviour of the sUAS, etc.

*Outer loop algorithm* - Determining departure and arrival sequences which is the outer loop of our algorithm is an important factor for both flight planning and airspace capacity. Although we applied the centralised FCFS algorithm and the LCFS algorithm to decide the sequence of arrival and departure, there are many different algorithms that use many different approaches and techniques. We expect that the algorithm to be changed according to the airspace structure and the use of airspace.



*N-to-M case* - There could be abundant of airspace users who have different retail points with different priorities. Therefore, the algorithm should consider multiple start points, and each airspace user's flight plans also should be calculated separately. Here, determining the multiple airspace users' arrival and departure order is also a key issue to be addressed to efficiently manage and operate the airspace. There can be several approaches to tackle the issue such as intention-based approaches, criteria-based approaches, etc.

*Uncertainty* - One possible approach to include wind effect is to increase the minimum separation requirement, although the approach will negatively affect the airspace capacity and throughput. If the static wind effect could be calculated as a vector, the algorithm can directly consider the effect. Also, contingency planning will be available with our algorithm if well-structured emergency infrastructures exist such as emergency landing points.

## REFERENCES

- [1] Amazon.com Inc., "Determining Safe Access with a Best-Equipped, Best-Served Model for Small Unmanned Aircraft Systems," NASA UTM 2015: The Next Era of Aviation, 2015.
- [2] C. A. Thiels, J. M. Aho, S. P. Zietlow, and D. H. Jenkins, "Use of unmanned aerial vehicles for medical product transport," *Air Medical Journal*, vol. 34, no. 2, pp. 104–108, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.amj.2014.10.011>
- [3] Federal Aviation Administration, "Operation and Certification of Small Unmanned Aircraft Systems," Federal Aviation Administration, Tech. Rep. 124, 2016. [Online]. Available: <https://federalregister.gov/a/2016-15079>
- [4] Geister and Dagi, "Concept for Urban Airspace Integration DLR U-Space Blueprint," Institute of Flight Guidance, Tech. Rep. December, 2017. [Online]. Available: [http://www.dlr.de/fl/desktopdefault.aspx/tabid-11763/20624\\_read-48305/](http://www.dlr.de/fl/desktopdefault.aspx/tabid-11763/20624_read-48305/)
- [5] J. Hoekstra, S. Kern, O. Schneider, F. Knabe, and B. Lamiscarre, "Metropolis – Urban Airspace Design," Technical University of Delft National, Tech. Rep., 2015.
- [6] J. Rios, D. Mulfinger, J. Homola, and P. Venkatesan, "NASA UAS traffic management national campaign: Operations across Six UAS Test Sites," in 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC). IEEE, sep 2016, pp. 1–6. [Online]. Available: <http://ieeexplore.ieee.org/document/7778080/>
- [7] M. F. B. Mohamed Salleh and K. H. Low, "Concept of Operations (ConOps) for Traffic Management of Unmanned Aircraft Systems (TM-UAS) in Urban Environment," in AIAA Information Systems-AIAA Infotech @ Aerospace, no. January. Reston, Virginia: American Institute of Aeronautics and Astronautics, jan 2017, pp. 1–13. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/6.2017-0223>
- [8] S. Bortoff, "Path planning for UAVs," Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334), no. June, pp. 364–368 vol.1, 2000. [Online]. Available: <http://ieeexplore.ieee.org/document/878915/>
- [9] J. Enright, E. Frazzoli, K. Savla, and F. Bullo, "On Multiple UAV Routing with Stochastic Targets: Performance Bounds and Algorithms," AIAA Guidance, Navigation, and Control Conference and Exhibit, no. August, 2005. [Online]. Available: <http://arc.aiaa.org/doi/10.2514/6.2005-5830>
- [10] G. B. Lamont, J. N. Slear, and K. Melendez, "UAV swarm mission planning and routing using multi-objective evolutionary algorithms," IEEE Symposium Computational Intelligence in Multicriteria Decision Making, no. Mcdm, pp. 10–20, 2007.
- [11] S. G. Manyam, S. Rasmussen, D. W. Casbeer, K. Kalyanam, and S. Manickam, "Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions," 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017, pp. 573–580, 2017.
- [12] K. Yu, A. K. Budhiraja, and P. Tokekar, "Algorithms for Routing of Unmanned Aerial Vehicles with Mobile Recharging Stations," in IEEE International Conference on Robotics and Automation 2018, mar 2018. [Online]. Available: <http://arxiv.org/abs/1704.00079>
- [13] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle Routing Problems for Drone Delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2017.
- [14] M. Samà, A. D'Ariano, P. D'Ariano, and D. Pacciarelli, "Scheduling models for optimal aircraft traffic control at busy airports: Tardiness, priorities, equity and violations considerations," *Omega (United Kingdom)*, vol. 67, pp. 81–98, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.omega.2016.04.003>
- [15] S. Bae, H.-S. Shin, C.-H. Lee, and T. Antonios, "A New Multiple Flights Routing and Scheduling Algorithm in Terminal Manoeuvring Area," in 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC) (in press), 2018.
- [16] M. F. B. Mohamed Salleh, C. Wanchao, Z. Wang, S. Huang, D. Y. Tan, T. Huang, and K. H. Low, "Preliminary Concept of Adaptive Urban Airspace Management for Unmanned Aircraft Operations," in 2018 AIAA Information Systems-AIAA Infotech @ Aerospace, no. January. Reston, Virginia: American Institute of Aeronautics and Astronautics, jan 2018, pp. 1–12. [Online]. Available: <https://arc.aiaa.org/doi/10.2514/6.2018-2260>
- [17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, dec 1959. [Online]. Available: <http://link.springer.com/10.1007/BF01386390>
- [18] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. [Online]. Available: <http://ieeexplore.ieee.org/document/4082128/>
- [19] R. T. Wong, "Combinatorial Optimization: Algorithms and Complexity (Christos H. Papadimitriou and Kenneth Steiglitz)," *SIAM Review*, vol. 25, no. 3, pp. 424–425, jul 1983. [Online]. Available: <http://epubs.siam.org/doi/abs/10.1137/1025101>
- [20] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Science*, vol. 17, no. 11, pp. 712–716, jul 1971. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.17.11.712>

# A new graph-based flight planning algorithm for unmanned aircraft system traffic management

Bae, Sangjun

2018-12-10

Attribution-NonCommercial 4.0 International

---

Sangjun Bae , Hyo–Sang Shin and Antonios Tsourdos. A new graph-based flight planning algorithm for unmanned aircraft system traffic management. Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), 23-27 September 2018, London UK  
<https://doi.org/10.1109/DASC.2018.8569609>

*Downloaded from CERES Research Repository, Cranfield University*