applied sciences

MDPI

Article

Structured Urban Airspace Capacity Analysis: Four Drone Delivery Cases

Sangjun Bae [1,2], Hyo-Sang Shin [1,*] and Antonios Tsourdos [1]

[1] School of Aerospace, Tranport, and Manufacturing, Cranfield University, Cranfield MK43 0AL, UK
[2] AI/BigData Optimization Team, LG Display, Paju-Si 10845, Republic of Korea
[*] Correspondence: h.shin@cranfield.ac.uk

Abstract: A route network-based urban airspace is one of the initial operational concepts of managing the high-density very low-level (VLL) urban airspace for unmanned aircraft system (UAS) traffic management (UTM). For the conceptual urban airspace, it is necessary to perform a quantitative analysis of urban airspace to stakeholders for designing rules and regulations. This study aims to discuss the urban airspace capacity for four different operation types by applying different sequencing algorithms and comparing its results to provide insight and suggestions for different operation cases to assist airspace designers, regulators, and policymakers. Four drone delivery operation types that can be applied in the high-density VLL urban airspace are analysed using the suggested four metrics: total flight time; total flight distance; mission completion time; the number of conflicts. The metrics can be calculated from a flight planning algorithm that we proposed in our previous studies. The algorithm for multiple agents flight planning problems consists of an inner loop algorithm, which calculates each agent's flight plan, and an outer loop algorithm, which determines the arrival and departure sequences. For each operation type, we apply two different outer loops with the same inner loop to suggest an appropriate sequencing algorithm. Numerical simulation results show tendencies for each type of operation with regard to the outer loop algorithms and the number of agents, and we analyse the results in terms of airspace capacity, which could be utilised for designing structures depending on urban airspace situations and environments. We expect that this study could give some intuition and support to policymakers, urban airspace designers, and regulators.

Keywords: drone delivery; flight planning algorithm; unmanned aircraft system (UAS) traffic management (UTM); graph theory; capacity analysis

Citation: Bae, S.; Shin, H.-S.; Tsourdos, A. Structured Urban Airspace Capacity Analysis: Four Drone Delivery Cases. *Appl. Sci.* **2023**, *13*, 3833. https://doi.org/10.3390/app13063833

Academic Editor: Augusto Ferrante

Received: 10 January 2023
Revised: 24 February 2023
Accepted: 6 March 2023
Published: 17 March 2023

1. Introduction

Many civilian applications of small unmanned aircraft systems (sUASs) have been envisioned to be operated in congested urban airspace for various purposes such as the delivery of medical supplies and the delivery of packages to rural areas [1–7]. Such applications are expected to significantly increase the quality of these services. However, there is no infrastructure to enable and safely manage the use of urban airspace and sUAS operations yet. Learning from the history of the air traffic management (ATM) systems, numerous organisations have investigated concepts, functional designs, and prototypes of unmanned aircraft system (UAS) traffic management (UTM) systems to support safe and efficient sUAS operations for many applications in urban airspace [8–14].

One of the most fundamental tasks facing the stakeholders of the UTM system over the world involves defining, measuring, and predicting of the capacity of urban airspace [2,15]. The capacity can be viewed from various perspectives such as safety, performance efficiency, conflicts, noise, communication spectrum, etc. The difficulty of the task is due to the numerous factors that affect the urban airspace capacity. In the ATM system most similar to the UTM system, the general notion of the airspace capacity is the number of flights that can be accommodated in a given airspace within a given time. In [16–20], the authors proposed capacity estimation approaches, which are evaluated from the human workload of air traffic controllers and pilots. These approaches are highly dependent on the subjective assessment or judgment of the air traffic controllers. However, such human-dependent airspace capacity estimation approaches are evaded and unsuitable for the UTM system which is expected to operate a large number of sUASs by a small number of people or by itself.

In [21], authors presented a method to calculate air traffic conflict probabilities. The method is more scalable and robust than the previous approaches. Furthermore, the method can be applied for any arbitrary stochastic distribution under the condition that the method requires constant communication between the vehicles and cooperative operations. In [22], the authors showed that the threshold-based mathematical definition to estimate the capacity for free flight-based low-altitude airspace. The authors used two metrics to estimate the capacity: safety (total loss of flight per flight hour) and performance (percentage extension of travel distance). In [23], the authors investigated new deconfliction schemes for unmanned aerial vehicles in high-density very low-level (VLL) uncontrolled airspace, and assessed single-layered and multi-layered airspace designs. Their aim was to provide a framework for choosing resolution strategies for regulators and policymakers for the UTM system. In [13], the authors determined the situation-appropriate route network for unmanned aircraft in urban airspace. The authors suggested three different types of route networks (AirMatrix, Over Buildings, Over Roads), and showed the results of the capacity and throughput for each route network. The results were obtained by generating five two-way routes for aerial delivery missions from the supply point to five service points where the routing problem was solved with an assumption that once a sUAS occupies a segment, then any other sUASs cannot traverse the route.

However, the approach caused very conservative analysis results and might disturb the optimality of solutions for the following problems: scheduling and separation management, although these have a computational advantage. As presented in the literature [11,12,22–24], on the development of advanced technology, the free flight-based operation in congested urban airspace can be an option for the UTM system. One of the few options is route network-based operations as described in [13,25]. Additionally, in the ATM context, opinions on the types of airspace operations diverge into two groups, a reduction in the constraints research [26,27] and more structured operations-preferred research [28,29].

Inspired by [13,25,30,31], we investigate strategies for analysing high-density VLL structured and controlled urban airspace capacity, with the aim to support airspace designers, regulators, and policymakers for determining tailored strategies in the consideration of regional characteristics and environment. The benchmark scenarios considered to perform capacity analysis are four different drone delivery operation types (1-to-$\mathcal{M}$, $\mathcal{M}$-to-1, $\mathcal{N}$-to-$\mathcal{M}$, and $\mathcal{M}$-to-$\mathcal{N}$, where $\mathcal{M}, \mathcal{N}$ is a number of arrival or departure points and $\mathcal{M} > \mathcal{N}$) in a route network over the roads in a given urban airspace. To overcome the disadvantages of the five two-way route planning used in [13] and perform quantitative capacity analysis, we develop a flight planning algorithm which is an extended version of our previous algorithm [32,33].

The proposed algorithm that iteratively solves multiple sUASs' flight planning problems consists of an inner loop algorithm and an outer loop algorithm. At each iteration, the inner loop algorithm generates each sUAS' flight route and speed profile by solving the shortest path problem, and the outer loop algorithm determines the departure and arrival sequences. To suggest an arrival and departure algorithm for each operation type, we utilise two different outer loop algorithms, which are the first-come-first-served (FCFS) algorithm and the last-come-first-served (LCFS) algorithm, for allocating the sequences. We utilise the obtained flight plans to analyse the urban airspace by defining four metrics as follows: total flight time; mission completion time; total flight distance; and normalised number of conflicts. For each operation type, we conduct 100 Monte Carlo simulations for both the FCFS algorithm and the LCFS algorithm. From the results, we can either find an appropriate outer loop algorithm for regional characteristics and the cases or reconfigure route networks as is required. We expect that the simulation results and its analysis could provide insight for policymakers, urban airspace designers, and regulators from the perspective of safety, capacity, etc.

The structure of this paper is as follows. We first focus on a design of the route network over roads in Section 2. In the following section, we provide details of the flight planning algorithms for the different cases, and describe the metrics to analyse the urban airspace capacity. Note that, in order to avoid extraneous examples to explain the flight planning algorithm in Section 3, we adopt one simulation case in Section 4 to explain it. In Section 4, we perform numerical simulations for each operation type and analyse the results. Finally, the conclusion and discussion are presented in Section 5.

2. Urban Airspace Concept: Route Network over Roads

Many studies have focused on the sUAS path planning problem [24,34–46]. The majority of them solve the problem with the unstructured airspace assumption. For urban airspace users in high-density VLL urban airspaces, however, many difficulties are expected with sUAS studies, which are based on free routing and free flight operations, such as the risk of unforeseen crashes, the invasion of privacy, etc. On the other hand, structured and controlled airspace can predict crashes more easily than the unstructured airspace, and privacy can be considered in the airspace structural design stage.

In this section, we suggest a route network-based urban airspace structure, which can be a potential candidate for the high-density VLL airspace structure. The proposed urban airspace structure concept resembles the airspace structure used in en-route phases of the ATM system today that can be considered as a combination of the Layers and Tubes concepts amongst four concepts Full Mix, Layers, Zones, and Tubes presented in [31]. The characteristics of the suggested airspace structure concept are as follows: consist of waypoints (nodes) and segments (edges) connecting some pair of waypoints; flight direction is strictly defined for all segments; at least two layers are stacked; separate traffic based on a time-based separation concept within each layer. As shown in Figure 1, for example, there is an eastbound layer and a westbound layer, and each layer consists of parallel-aligned northbound segments and southbound segments. Such a route network over roads with directions in urban airspace will increase the predictability of traffic by using flight plans. In this concept, the urban airspace structure can be thought of as a directed graph (or digraph) $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, called an airspace graph to highlight the aerospace context. In the airspace graph $\mathcal{G}$, node $v \in \mathcal{V}$ is a point in Euclidean space of dimension two or three representing a waypoint candidate to be traversed. Edge $e \in \mathcal{E}$, corresponding to a rectifiable curve, is a segment between some pair of waypoints in the airspace graph $\mathcal{G}$.

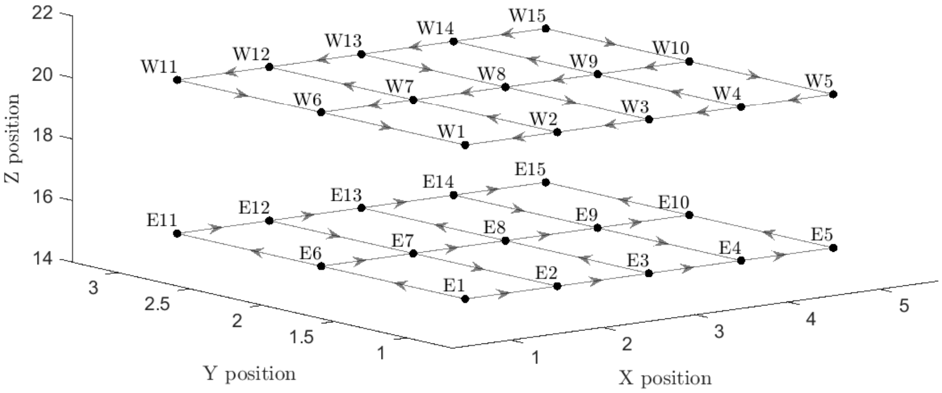


Figure 1. An example of the proposed route network-based urban airspace structure concept.

The main assumptions we make for the route network-based operations in this study are as follows: each flight route is composed of a series of linear segments as shown in Figure 2; each sUAS flies along each segment at a constant speed while it changes speed in between segments; each sUAS is able to exactly follow the flight route without deviation; a time-based separation concept is utilised for collision avoidance; uncertainties produced by external sources are neglected such as adverse weather (i.e., true airspeed is considered); a control tower exists (i.e., flight plans are shared between sUASs and a control tower); the hovering and vertical manoeuvres are not considered; sUASs have limited payload and flight range; we only consider flights within each layer (e.g., no transitions between layers), but the transitions could be adopted by establishing the vertical segments in reality; homogeneous sUASs.

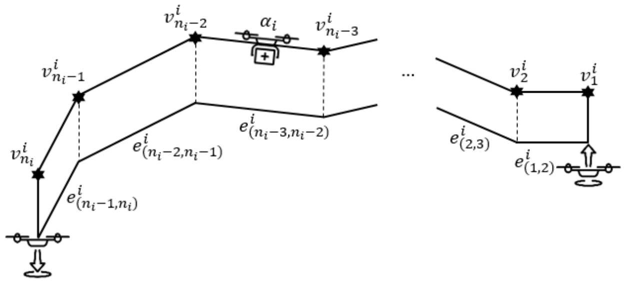


Figure 2. A series of linear segments (sUAS α_i's flight route from its origin v^i_1 to its destination $v^i_{n_i}$).

3. Flight Planning Algorithm

In this section, we describe the flight planning algorithm that is used to analyse the route network-based urban airspace. The algorithm that solves multiple sUASs flight planning problems consists of the inner loop algorithm and the outer loop algorithm. The inner loop algorithm optimises each flight plan of an sUAS that minimises the flight time while satisfying the time-based minimum separation requirement from other planned sUASs. Then, the outer loop algorithm allocates arrival and departure sequences of the sUASs.

Definition 1. *We define the flight planning problem as the following: (a) the flight route of each sUAS, (b) its separation-compliant speed profile including arrival times at each waypoint, and (c) the arrival and departure sequences (if it is a multiple sUASs flight planning problem).*

This section consists of five parts: (a) a description of the time-based separation concept; (b) flight planning problem formulation for each sUAS (inner loop); (c) the entire flight planning algorithm with two different outer loop algorithms (FCFS and LCFS); (d) a flight planning example using the proposed algorithm for a last-mile delivery case (1-to-$\mathcal{M}$); and (e) a description of four metrics to quantitatively analyse the urban airspace capacity.

3.1. Separation Concept in Route Network-Based Unban Airspace

Conflict detection is activated when the separation of two sUASs is less than the minimum separation criterion. In this study, time-based separation is applied instead of the widely used distance-based separation to stabilise the spacing between all the sUASs. In this concept, a sUAS will occupy a waypoint when the sUAS passes the waypoint for a predetermined time interval. Within this time interval, no sUAS is allowed to traverse through the waypoint to satisfy the separation requirement. For each waypoint, a time interval list is maintained to keep track of the times at which the waypoint is expected to be occupied. These lists, called the flight schedule in this study, are shared between all sUASs during operation. By satisfying the separation assurance flight time at each waypoint rather than adjusting a distance between the sUASs, we can obtain results that satisfy the minimum separation along routes as well as at waypoints at all times. Note that, even if we consider homogeneous sUASs in this study, the separation concept can be applied to both heterogeneous and homogeneous cases.

The separation concept allows time-based separation to be satisfied at merging points and crossing points as shown in Figures 3 and 4, respectively. At the time t_0 of Figure 3, sUAS α_i and sUAS $\alpha_{i'}$ fly towards the v_C through the same merging point v_Y at speeds

s_1^i and $s_1^{i'}$, respectively. The time t_0 is stored at each start point $T(v_\mathrm{A})$ and $T(v_\mathrm{B})$, and no sUAS can pass through both of the waypoints for separation time $t_{SP}^{(i'',i)}$ seconds and $t_{SP}^{(i'',i')}$ seconds before and after t_0, respectively. Note that, where $t_{SP}^{(i',i)}$ is dependent on the types of α_i and $\alpha_{i'}$. The time t_1 of Figure 3, when $\alpha_{i'}$ just passes through v_Y, is stored at the merging point $T(v_\mathrm{Y})$, and no sUAS can pass through this waypoint for $t_{SP}^{(i',i)}$ seconds before or after t_1. The time t_2 of Figure 3 is stored in $T(v_\mathrm{Y})$ in the same way. Then, α_i and $\alpha_{i'}$ traverse towards v_C at speeds of s_2^i and $s_2^{i'}$, respectively. If α_i and $\alpha_{i'}$ fly from v_M to v_C at speeds of s_2^i and $s_2^{i'}$, respectively, the separation between the two sUASs will always meet the separation requirement or be greater than that on the segment between v_Y and v_C. At the time t_3 of Figure 3, α_i and $\alpha_{i'}$ fly at the same speed while maintaining the minimum separation. The flight schedule from t_0 to t_3 at each waypoint is as follows:

$$\begin{aligned}
T(v_\mathrm{A}) &= \{0s\}, \\
T(v_\mathrm{B}) &= \{0s\}, \\
T(v_\mathrm{Y}) &= \{0s, 120s, 210s\}, \\
T(v_\mathrm{C}) &= \emptyset.
\end{aligned}$$

If no sUAS has passed waypoint v_C, then $T(v_\mathrm{C}) = \emptyset$. Such a flight schedule at each waypoint is used to formulate a flight planning problem that will be described in the following section. Note that, in this study we assume that all sUASs are of the same type to analyse the capacity with less decision parameters.

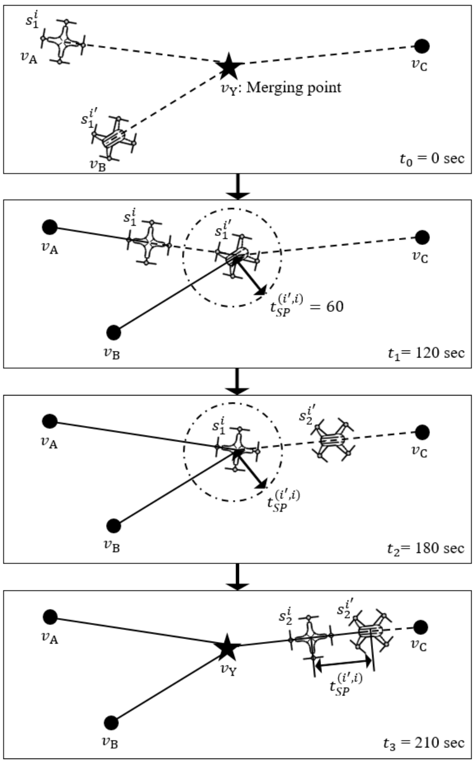


Figure 3. Time-based separation concept for merging points. α_i and $\alpha_{i'}$ are represented by superscript i and i' in the figure, fly from their origin v_A and v_B to v_C through the merging point v_Y at speeds $\{s_1^i, s_2^i\}$ and $\{s_1^{i'}, s_2^{i'}\}$, respectively. The superscript and subscript of s are the sUAS index and segment index, respectively.

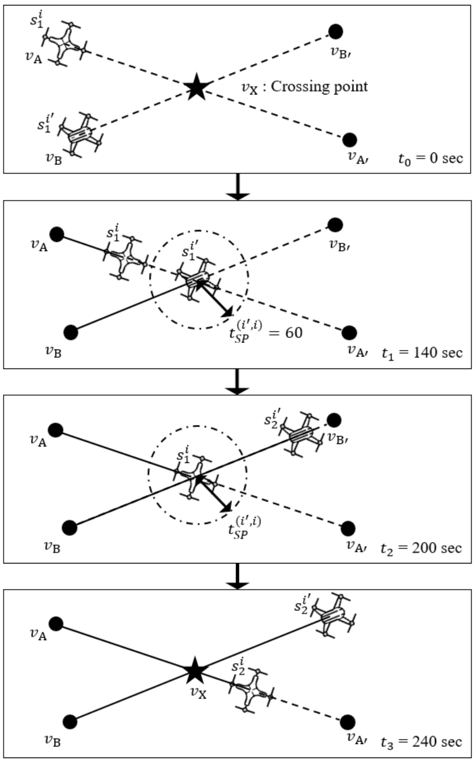


Figure 4. Time-based separation concept for crossing points. α_i and $\alpha_{i'}$ are represented by superscript i and i' in figure, fly from their origin point v_A and v_B to their final point $v_{A'}$ and $v_{B'}$ through the crossing point v_X at speeds $\{s_1^i, s_2^i\}$ and $\{s_1^{i'}, s_2^{i'}\}$, respectively. The superscript and subscript of s are the sUAS index and segment index, respectively.

3.2. Inner Loop: Single sUAS Flight Planning Problem

In this section, our focus is on flight planning a finite set of sUASs in a given route network-based urban airspace.

Definition 2. *Let $\mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_i\}$ be a finite set of sUASs. The set is partitioned into two sets: a set of sUASs to be planned $\mathcal{A}^+$, and a set of planned sUASs $\mathcal{A}^-$ such that $\mathcal{A}^+ \cup \mathcal{A}^- = \mathcal{A}$, $\mathcal{A}^+ \cap \mathcal{A}^- = \emptyset$.*

In the urban airspace, each sUAS flies from its origin to its destination. A feasible flight route for each sUAS in the urban airspace is defined as follows:

Definition 3. *Is given in the airspace graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, sUAS $\alpha_i \in \mathcal{A}^+$ to be routed in $\mathcal{G}$, the initial waypoint $v_1^i \in \mathcal{V}$ and the final waypoint $v_{n_i}^i \in \mathcal{V}$ to construct a flight route denoted by $p(\alpha_i)$ in $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ defined by a sequence of waypoints.*

Thus, in the airspace graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, there can be an abundance of flight route candidates denoted by $\mathcal{C}$ that satisfy the conditions as described in Definition 3. Through Definition 3, each flight route candidate $p(\alpha_i) \in \mathcal{C}$ can be given a corresponding flight route (a set of waypoints) as follows:

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

where n_i is the number of waypoints sUAS α_i traverses as shown in Figure 2. For each sUAS' flight route as given in above equation, there is a set of segments connecting the waypoints, which is as follows:

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

For each flight route candidate, it is assumed that a performance index for a set of segments $\mathcal{E}(p(\alpha_i))$ can be quantified as a set of positive numeric weighting values, as follows:

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

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 segment for all flight routes. In the airspace graph $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$, each flight route can be estimated by summing all weights of $\mathcal{W}(p(\alpha_i))$, as follows:

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

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

Definition 4. *Given an airspace graph $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$ and corresponding flight route candidates $\mathcal{C}$, the routing problem is defined as finding a flight route (or a sequence of waypoints) such that*

$$p^*(\alpha_i) = \underset{p(\alpha_i)\in\mathcal{C}}{\operatorname{argmin}}\, \mathcal{T}(p(\alpha_i)). \tag{2}$$

The optimal flight route $p^*(\alpha_i)$ can be found by using the well-known shortest path algorithms such as Dijkstra's algorithm, A* algorithm or exhaustive search algorithm [47]. Although the optimal flight route $p^*(\alpha_i)$ can be obtained according to Definition 4, if the airspace users want to manage the separation between each sUAS in sequential approaches, the optimality of the obtained flight route might be disturbed in the following separation management stages. One possible option is to solve the multiple sUASs flight planning problem using numerical trajectory optimisation approaches, which is not scalable although the approaches provide high-fidelity flight plans to each sUAS. Such approaches can also require additional processes for managing separation, which might cause optimality or scalability issues.

Our objective of formulating the flight planning problem is to find a flight route and speed profile that minimises the flight time of each sUAS while satisfying the separation requirement. The main idea for achieving the objective is to assign a weight to each segment (edge) of the airspace graph where the weight is expressed as a flight time. Then, each sUAS' speed profile also can be obtained by finding its flight route in the airspace graph as defined in Definition 4. We set a flight time of each sUAS as weight $w \in \mathcal{W}$ to each segment $e \in \mathcal{E}$ of the airspace graph $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W})$. Note that, a flight time and speed of each sUAS are mutually interchangeable using geographic data included in the airspace graph. Where each speed profile of sUAS is determined within a feasible speed range of each sUAS. Another issue we have pursued is to satisfy the separation requirement between every pair of sUASs. To assign a flight time that satisfies the separation requirement to each edge of $\mathcal{G}$, we need the flight schedule of the set of planned sUASs including the arrival times of each sUAS at each waypoint as discussed in Section 3.1. The flight schedule T will be included in the airspace graph $\mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W}, T)$, and used to calculate the weights of the airspace graph. By finding a solution of the airspace graph $\mathcal{G}$, then, we can obtain a flight route and speed profile of the sUASs while satisfying the separation requirement simultaneously. Each sUAS has different weights $\mathcal{W}_i$ because of different specifications such as the feasible speed range or the separation requirement depending on nearby sUASs. Therefore, sUAS $\alpha_i \in \mathcal{A}^+$ has its own airspace graph $\mathcal{G}_i$, as shown in Figure 5.

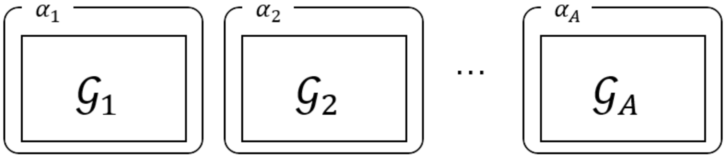


Figure 5. Independent airspace graphs for each sUAS.

Flight Time Weights $\mathcal{W}_{i,t}$ on $\mathcal{G}_i$

We propose a weighting scheme to be applied to the airspace graphs $\mathcal{G}_i$ to fulfil separation-compliant speed profiles. First of all, we introduce a set of constant speeds for each sUAS as follows:

$$s(\alpha_i) : s^i_1, s^i_2, \ldots, s^i_{n_i-1}$$

where each element of the set $s(\alpha_i)$ must be within a feasible speed range $[s^i_{\min}\ s^i_{\max}]$, $\forall \alpha_i \in \mathcal{A}$. Furthermore, we define a set of flight distances using geographical data allowing to compute the set $s(\alpha_i), \forall \alpha_i \in \mathcal{A}^+$

$$d(\alpha_i) : d^i_1, d^i_2, \ldots, d^i_{n_i-1}.$$

Let $u^i_j = 1/s^i_j, \forall j \in \{1, 2, \ldots, n_i - 1\}, \forall \alpha_i \in \mathcal{A}^+, u^i_{\max} = 1/s^i_{\min}$, and $u^i_{\min} = 1/s^i_{\max}$, we formulate a linear programming problem and solve the problem to obtain the flight time weight $\mathcal{W}_{i,t}$, $\forall \alpha_i \in \mathcal{A}^+$ as follows:

$$\min \quad \sum_{j=1}^{n_i-1} u^i_j d^i_j, \tag{3}$$

$$\text{s.t.} \quad d^i_1 u^i_1 + d^i_2 u^i_2 + \cdots + d^i_{n_i-1} u^i_{n_i-1} \geq \max(T(v^i_{n_i})) + t^{(i',i)}_{SP}, \tag{4}$$

$$d^i_1 u^i_1 + d^i_2 u^i_2 + \cdots + d^i_{n_i-2} u^i_{n_i-2} \geq \max(T(v^i_{n_i-1})) + t^{(i',i)}_{SP}, \tag{5}$$

$$\vdots$$

$$d^i_1 u^i_1 + d^i_2 u^i_2 \geq \max(T(v^i_3)) + t^{(i',i)}_{SP}, \tag{6}$$

$$d^i_1 u^i_1 \geq \max(T(v^i_2)) + t^{(i',i)}_{SP}, \tag{7}$$

$$u^i_{\min} \leq u^i_1, u^i_2, \cdots, u^i_{n_i} \leq u^i_{\max} \tag{8}$$

where $t^{(i',i)}_{SP}$ is the minimum time-based separation requirement between preceding sUAS $\alpha_{i'}$ and following sUAS α_i at waypoints as shown in Figures 3 and 4. The objective function of Equation (3) is to minimise the sum of flight time of each segment. The constraints of Equations (4)–(7) are for satisfying the minimum separation requirement at each waypoint. In Figure 2, for example, if $\mathcal{E}(p(\alpha_i))$ consists of $n_i - 1$ segments for sUAS α_i, each segment requires a flight time that satisfies the separation requirement. The left-hand side of each constraint of Equations (4)–(7) is the flight time for the segment of each constraint, which must be greater than and equal to the time that satisfies the separation requirement. For the constraints, flight distances between waypoints and the flight schedule are required. The flight schedule is stored in T for each waypoint $v \in \mathcal{V}$, and T is updated every time a sUAS is allocated and shared with $\forall \alpha_i \in \mathcal{A}^+$.

$$T(v) = \{t^v_1, t^v_2, \ldots, t^v_A\}. \tag{9}$$

A is the number of elements in $\mathcal{A}$. Then, the formulated optimisation problem can be solved by any well-known linear programming algorithm. In this study, we utilise the interior-point algorithm. In Equation (8), decision variables, $s^i_1, s^i_2, \cdots, s^i_{n_i-1}$, are constant speeds for the flight segment, which are transformed into flight times, and assigned into weights $\mathcal{W}_{i,t}$ for each flight route candidate of sUAS $\alpha_i \in \mathcal{A}^+$.

$$\mathcal{W}_{i,t} : w^i_{1,t}, w^i_{2,t}, \cdots, w^i_{n_i-1,t}.$$

Therefore, a solution to the airspace graph $\mathcal{G}_i = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}, T)$ can simultaneously provide a flight route and its speed profile for sUAS $\alpha_i, \forall \alpha_i \in \mathcal{A}^+$ while satisfying minimum separation at all times.

We also construct flight distance weights $\mathcal{W}_d$ on $\mathcal{G}_i$ to utilise the flight distance d as a second criterion. The weights $\mathcal{W}_d$ are the same for all sUASs, which are necessary to prioritise for the multiple sUASs' flight planning problem when two or more sUASs arrive at the same destination at the same time, more details about the priority are discussed in Section 3.3. The single sUAS flight planning problem that motivated this study can be formulated as follows:

Problem 1. *Given an unique airspace graph $\mathcal{G}_i = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}, \mathcal{W}_d, T)$, sUAS α_i in $\mathcal{G}_i$, and its origin $v^i_1 \in \mathcal{V}$ and destination $v^i_{n_i} \in \mathcal{V}$ reachable from the origin, construct a flight 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 sUAS $\forall \alpha_i \in \mathcal{A}^-$;*
- *the speed profile of each sUAS must be within its feasible speed range;*
- *and the airspace graphs are updated every time when a sUAS $\alpha_i \in \mathcal{A}^+$ is planned.*

In this study, we solve Problem 1 using Dijkstra's algorithm [47].

3.3. Outer Loop: Multiple sUASs Flight Planning

The main idea of this study is that the single sUAS flight planning problem can be formulated by assigning the separation-satisfied flight time weights in the airspace graph as described in Section 3.2. By doing so, the problem is able to not only manage separation between every pair of sUASs, but also consider operational factors such as the feasible speed ranges, available routes, the separation requirements. However, the separation of each sUAS only satisfies the separation requirement from a set of allocated sUASs. In this section, we address the multiple sUASs flight planning problem by solving each single sUAS flight planning problem in an iterative way. At each iteration, each sUAS finds the optimal flight route including its speed profile by solving Problem 1 (inner loop), and one of the sUASs is allocated using a criterion-based algorithm (outer loop). In this study, we utilise the FCFS and LCFS algorithms for the outer loop algorithm.

Once inputs of the multiple sUASs flight planning problem are given, Algorithm 1 starts the flight planning process until $\mathcal{A}^+$ is empty (Line 3). The algorithm first generates airspace graphs $\mathcal{G}^k_i = (\mathcal{E}, \mathcal{V}, \mathcal{W}_d, T^k), \forall \alpha_i \in \mathcal{A}^+$ (Line 2). Assigning separation-satisfied flight time weights to all possible flight paths in the airspace graph is a time-consuming task. In order to reduce the complexity of the problem caused by the high degree of the airspace graphs, we select the h-shortest routes for $\mathcal{G}^k_i = (\mathcal{E}, \mathcal{V}, \mathcal{W}_d, T^k), \forall \alpha_i \in \mathcal{A}^+$ by using Yen's algorithm, and generate $\bar{\mathcal{G}}^k_i$ only considering the h-shortest routes (Line 4–5) [48]. Then, each sUAS has its unique airspace graph that contains flight time weights $\mathcal{W}^k_{i,t}$ determining the optimal flight route with its speed profile. The optimal flight route $p^*(\alpha_i)$ and its speed profile $s(\alpha_i)$ are obtained by using Dijkstra's algorithm, $\forall \alpha_i \in \mathcal{A}^+$ (Line 6) [49]. sUAS α^*_i that arrives first to its destination is obtained by using the FCFS algorithm (Line 7), and allocated (Line 15). The flight plan of sUAS α^*_i is transferred from $\mathcal{A}^+$ to $\mathcal{A}^-$ (Line 12). Note that, in this study, α_i refers to a flight as well as a sUAS. In the case that more than two sUASs arrive at the same destination at the same time (Line 8), the flight distance weights $\mathcal{W}_d$ are utilised to allocate sUAS $\alpha^\dagger_i$ amongst them (Line 9–13). In either case, a sUAS is allocated and its flight schedule is shared with sUAS $\alpha_i, \forall \alpha_i \in \mathcal{A}^+$. Based on the flight schedule each airspace graph $\mathcal{G}^k_i$ is updated, $\forall \alpha_i \in \mathcal{A}^+$ (Line 20). Note that, to change the outer loop algorithm to other criterion-based algorithms, such as the LCFS algorithm, users are only required to change the outer loop algorithm Line 7 and Line 10 to what the users want.

Algorithm 1: The multiple sUASs flight planning algorithm

```
   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 $\mathcal{G}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_d, T^k)$, $\forall \alpha_i \in \mathcal{A}^+$
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 $\bar{\mathcal{G}}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}^k, \mathcal{W}_d, T^k)$ that only considers the best $h$ routes, $\forall \alpha_i \in \mathcal{A}^+$;
6      find $p^*(\alpha_i)$ of $\bar{\mathcal{G}}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}^k, T^k)$, $\forall \alpha_i \in \mathcal{A}^+$ (using Dijkstra's algorithm);
7      $\alpha_i^* \leftarrow \text{argmin}_{\forall \alpha_i \in \mathcal{A}^+} p^*(\alpha_i)$ (using FCFS 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         $\alpha_i^\dagger \leftarrow \text{argmin}_{\forall \alpha_i^*} p^*(\alpha_i)$ (using FCFS or LCFS algorithm);
11         allocate $\alpha_i^\dagger$;
12         transfer $\alpha_i^\dagger$ from $\mathcal{A}^+$ to $\mathcal{A}^-$;
13         share $\mathcal{A}^-$ with $\forall \alpha_i \in \mathcal{A}^+$;
14     else
15         allocate $\alpha_i^*$;
16         transfer $\alpha_i^*$ from $\mathcal{A}^+$ to $\mathcal{A}^-$;
17         share $\mathcal{A}^-$ with $\forall \alpha_i \in \mathcal{A}^+$;
18     end
19     $k = k + 1$;
20     update $\mathcal{G}_i^k = (\mathcal{E}, \mathcal{V}, \mathcal{W}_{i,t}^k, \mathcal{W}_d, T^{\mathrm{k}})$, $\forall \alpha_i \in \mathcal{A}^+$ ;
21 end
```

We illustrate an example in Figure 6 for a better understanding of Algorithm 1 with an case that A sUASs fly in the urban airspace. In the first iteration, each sUAS generates its airspace graph $\mathcal{G}_i^1$, $\forall \alpha_i \in \mathcal{A}^+ = \{\alpha_1, \alpha_2, \ldots \alpha_A\}$. Through the inner loop algorithm, each sUAS's $p^*(\alpha_i)$ is determined, $\forall \alpha_i \in \mathcal{A}^+$, and the most flight-time-efficient sUAS, which is α_2, is allocated using the FCFS algorithm. In the second iteration, sUAS α_A is allocated through the same process. In the $(A-1)$th iteration, sUAS α_3 is allocated, and sUAS α_1 is allocated last.

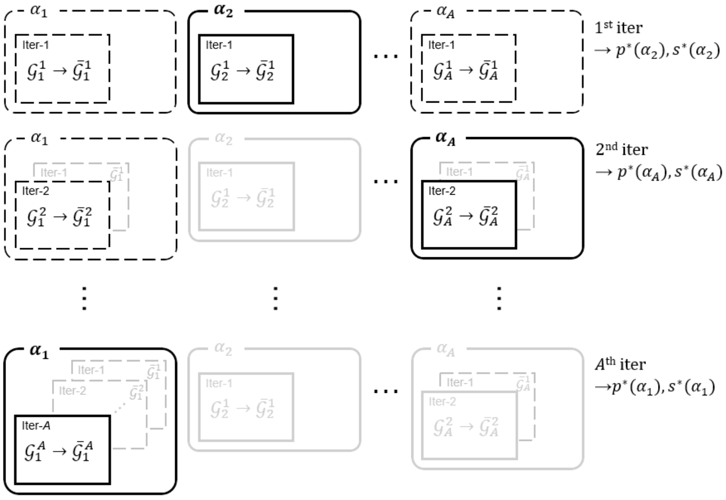

α_1
Iter-1
$\mathcal{G}_1^1 \to \bar{\mathcal{G}}_1^1$
α_2
$\mathcal{G}_2^1 \to \bar{\mathcal{G}}_2^1$
α_A
$\mathcal{G}_A^1 \to \bar{\mathcal{G}}_A^1$
1st iter
$\to p^*(\alpha_2), s^*(\alpha_2)$
Iter-2
$\mathcal{G}_1^2 \to \bar{\mathcal{G}}_1^2$
$\mathcal{G}_A^2 \to \bar{\mathcal{G}}_A^2$
2nd iter
$\to p^*(\alpha_A), s^*(\alpha_A)$
Iter-A
$\mathcal{G}_1^A \to \bar{\mathcal{G}}_1^A$
Ath iter
$\to p^*(\alpha_1), s^*(\alpha_1)$

Figure 6. Iteratively generated airspace graphs.

3.4. Flight Planning Example for 1-to-$\mathcal{M}$ Case (Last-Mile Delivery with a Single Retail Point)

We used a simulation case in Section 4 to explain the flight planning algorithm and did not include extraneous examples to avoid confusion. We appreciate the readers' understanding in this matter. This example is a sample of the Monte Carlo simulations for the 1-to-$\mathcal{M}$ case, and its simulation settings are described at the beginning of Section 4. Put briefly, thirty sUASs traverse from the retail point (E68) to each predetermined sUAS' service points (E1∼E67) in Figure 8. We apply both the FCFS and LCFS algorithms to this example, and thirty flight plans for each case are shown in Tables 1 and 2, respectively. The results show each sUAS' separation requirement-satisfied flight route and speed profiles for each outer loop algorithm. Due to a 5 s departure separation, 5 s horizontal separation as well as the single retail point of this case, all sUASs find their flight routes with the maximum speed without any conflicts for both cases. For the same reasons, most of the flight routes are the same as the shortest path of each sUAS. Table 3 shows a comparison of the results between the FCFS and LCFS algorithms for this example. The total flight time and the total flight distance of thirty sUASs for the FCFS algorithm case is almost same as the LCFS algorithm case. However, the mission completion time for the LCFS algorithm case is 34.5% more efficient than the FCFS algorithm case. Therefore, although the total flight time and the total flight distance are very similar for both cases, the LCFS algorithm case is more efficient with regard to the urban airspace operating time. Both airspace users and airspace service providers prefer to operate as many sUASs as possible in a given time within a given urban airspace. However, if there are more than two urban airspace users with more than two retail points, the urban airspace usage priority will be an essential factor to be considered for the flight planning problems, which is a similar situation reported for today's airspace for commercial aircraft.

Advantages of the proposed algorithm are as follows: able to adopt different sequencing algorithms; providing separation-compliant speed profiles that satisfy each sUAS's performance; applicable to various types of route networks; enlarge a solution-searching space by finding a flight route and speed profile of each sUAS simultaneously; and fast computational time. Therefore, we expect that the advantages of Algorithm 1, which provides not only the flight route but also the detailed schedule, will be appropriate to analyse the capacity of the high-density VLL urban airspace populated by many sUASs with short flight times, and many service providers.

Table 1. Flight planning results of 30 sUASs for the 1-to-$\mathcal{M}$ case (outer loop: FCFS algorithm).

sUAS ID	Retail Point	Service Point	Waypoint Sequence (Arrival Time at Each Waypoint (s)) Speed Profile (km/h)
⋮	⋮	⋮	⋮
a_4	E68	E4	E68-E24-E17-E1-E2-E3-E4 (180)-(197)-(217)-(244)-(268)-(280)-(293) 25-25-25-25-25-25
a_5	E68	E5	E68-E24-E17-E1-E2-E3-E4-E5 (220)-(237)-(257)-(284)-(308)-(320)-(333)-(345) 25-25-25-25-25-25-25
⋮	⋮	⋮	⋮
a_{22}	E68	E36	E68-E32-E33-E34-E35-E36 (130)-(147)-(169)-(181)-(193)-(206) 25-25-25-25-25
a_{23}	E68	E38	E68-E32-E33-E34-E35-E36-E37-E38 (170)-(188)-(209)-(221)-(234)-(246)-(258)-(271) 25-25-25-25-25-25-25
⋮	⋮	⋮	⋮

Table 2. Flight planning results of 30 sUASs for the 1-to-$\mathcal{M}$ case (outer loop: LCFS algorithm).

sUAS ID	Retail Point	Service Point	Waypoint Sequence (Arrival Time Each Waypoint (s)) Speed Profile (km/h)
⋮	⋮	⋮	⋮
a_4	E68	E4	E68-E24-E17-E1-E2-E3-E4 (110)-(127)-(147)-(173)-(197)-(210)-(223) 25-25-25-25-25-25
a_5	E68	E5	E68-E24-E17-E1-E2-E3-E4-E5 (70)-(87)-(107)-(133)-(157)-(170)-(183)-(196) 25-25-25-25-25-25-25
⋮	⋮	⋮	⋮
a_{22}	E68	E36	E68-E32-E33-E34-E35-E36 (160)-(177)-(198)-(211)-(223)-(236) 25-25-25-25-25
a_{23}	E68	E38	E68- E32-E33-E34-E35-E36-E37-E38 (120)-(137)-(158)-(171)-(183)-(196)-(208)-(221) 25-25-25-25-25-25-25
⋮	⋮	⋮	⋮

Table 3. Comparison between the FCFS and LCFS algorithms for the 1-to-$\mathcal{M}$ case.

	FCFS Algorithm	**LCFS Algorithm**
Total flight time (s)	7035.1	7031.1
Total flight distance (km)	19.3	19.3
Mission completion time (s)	468.7	307.0
Number of conflict	0	0

3.5. *Four Metrics for the Capacity Estimation*

In the ATM system, typically, the airspace capacity is defined as the maximum number of aircraft that can be accommodated by the air traffic controllers in a given airspace at any point of time, while throughput is defined as the number of aircraft that land at an airport over a specific time window. In the sUASs' flights in high-density VLL urban airspace, however, it is expected that it might be difficult to construct and negotiate multiple sUASs' flight plans a few days in advance, such as the flight plans in the ATM system, and it might be complicated to conduct separation management through conventional human air traffic controllers due to the large number of sUASs. In addition, compared with commercial aircraft, very short flight times, many landing points, many flights, and fewer operators are expected. Thus, we can expect that one option is to analyse the urban airspace capacity based on the detailed flight plans with regard to the flight time to complete the given mission, the total flight time, the total flight distance, and the number of conflicts. We utilise such detailed flight plans to define four metrics to analyse the urban airspace capacity.

3.5.1. Total Flight Time (M_{tft})

Intuitively, one metric to analyse the urban airspace capacity is the total flight time for a given mission. The total flight time can be calculated as follows:

$$M_{tft} = \sum_{i=1}^{A} \sum_{j=1}^{n_i-1} w^i_{j,t} \tag{10}$$

where A is the number of sUASs to be planned. M_{tft} can also be utilised as a metric to measure the operating costs for the sUAS operators.

3.5.2. Mission Completion Time (M_{mct})

Once missions are given to a set of sUASs, each sUAS generates its separation-compliant flight plan. Then, the mission completion time is the latest arrival time at a service point or a retail point of the last sUAS, which can be found as follows:

$$M_{mct} = \underset{\forall vs. \in \mathcal{V}}{\operatorname{argmax}}\, T^A(v). \tag{11}$$

The meaning of this metric can be used as an indicator to determine which algorithms are effective to operate the route network-based urban airspace under the same conditions. This will be one of the most important metrics for the urban airspace operators.

3.5.3. Total Flight Distance (M_{tfd})

The flight distance is a commonly used as a criterion for analysing the flight performance to find the optimal path in the urban airspace. Each sUAS' flight distance, which is used for the second criterion in Algorithm 1, can be obtained from the geometry information. The total flight distance can be calculated as follows:

$$M_{tfd} = \sum_{i=1}^{A} \sum_{j=1}^{n_i-1} d^i_j. \tag{12}$$

3.5.4. Normalised Number of Conflicts (M_{cnf})

Algorithm 1 generates a flight plan that only fulfils the destination separation requirement within the permissible speed range when the algorithm could not find a flight route that satisfies the separation requirement during the whole flight as well as the destination separation requirement and the permissible speed range. Namely, the sUAS could conflict with other sUASs during the flight based on the obtained flight plans. Although this paper does not provide overtaking manoeuvres, it is assumed that overtaking manoeuvres via vertical manoeuvring is sufficiently possible using current CD and R (conflict detection and resolution) technologies. Then, the number of conflicts is counted whenever a sUAS overtakes a preceding sUAS. Naturally, the number of conflicts increases with the number of sUASs in the urban airspace. We divide the number of conflicts by the total number of possible conflicts to normalise, and use it as a metric:

$$M_{cnf} = \frac{n_{\text{conf}}}{n_{\text{tot}}} \tag{13}$$

where n_{tot} is the total number of possible conflicts, which is $\frac{A\times(A+1)}{2}$, and n_{conf} is the number of conflicts.

4. Monte Carlo Simulations

In this section, we conduct Monte Carlo (MC) simulations with two different outer loop algorithms (the FCFS and LCFS algorithms) that are based on the proposed flight planning algorithm for four drone delivery operation types. We investigate how the different operation types' performance and safety behave with regard to the different flight planning algorithms and the number of sUASs in the high-density VLL urban airspace using the suggested four metrics. It aims to provide an approach for choosing between resolution strategies for urban airspace management by regulators, policymakers, and urban airspace designers.

The representative operation being considered in this study is drone delivery, especially last-mile delivery and first-mile delivery. Last-mile delivery is defined as the movement of items from transportation hubs (which are retailers in this study) to final delivery destinations, typically personal residences. First-mile delivery refers to the movement of goods from sellers to courier services who will take these goods to their final users.

A scenario for the simulations is based on the town 'Oldbrook', which is one of the well-planned towns in Milton Keynes in the United Kingdom. Two-dimensional infrastructure data of the town is obtained from Google Earth Pro. The scenario has an area size of 0.98 km^2 as shown in Figure 7. Similar to the en-route airspace for commercial aircraft, in this route network, we construct two layers of nodes that are set above the roads at heights of 15 m and 25 m as shown in Figures 8 and 9 for eastbound and westbound routes, respectively. We assume that each layer consists of 70 points (3 retail points (red pins) and 67 service points (green pins)) and 107 directed routes as shown in Figures 8 and 9. All sUASs are of the same type as shown in Table 4. The sUAS type can be later changed and varied based on the missions and operators. For the separation between sUASs, it is assumed that the time-based minimum vertical and horizontal separation requirement during the flight is 5 s. No hovering manoeuvres are allowed for any sUASs.

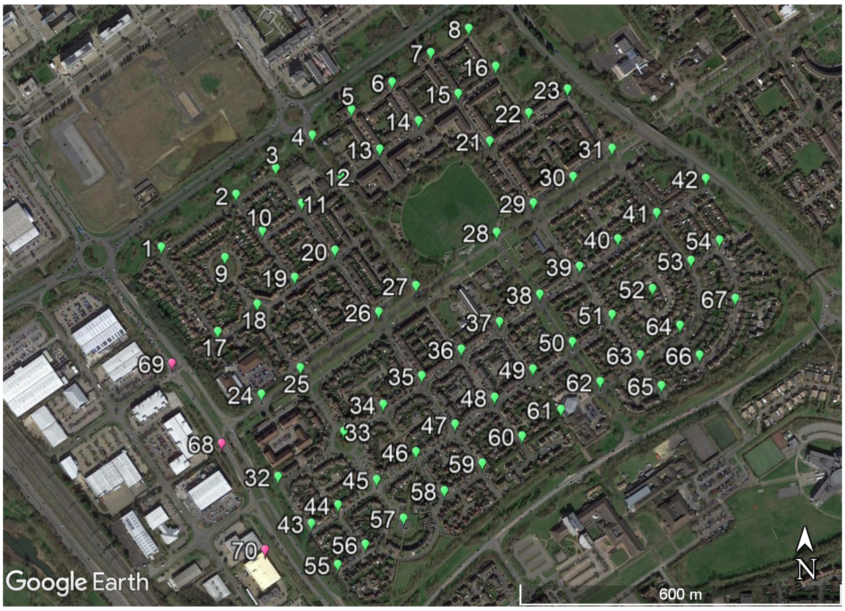


Figure 7. Oldbrook, Milton Keynes, United Kingdom (from Google Earth Pro). Green pins (1∼67) and red pins (67∼70) represents retail points and service points respectively.

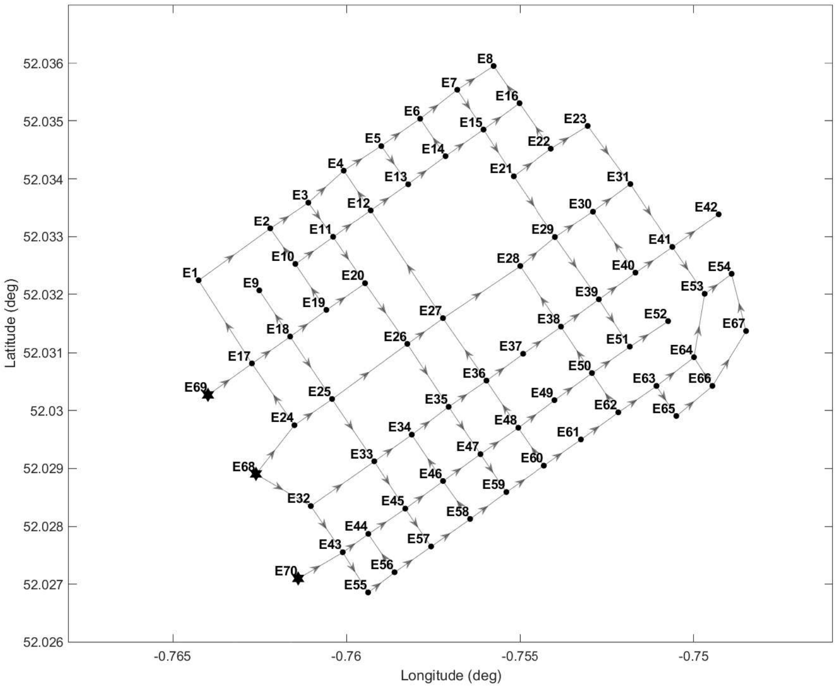


Figure 8. Eastbound at 15 m, 3 retail points (E68, E69, E70) and 67 service points (E1∼E67) and 107 directed routes, (Oldbrook, Milton Keynes, United Kingdom).

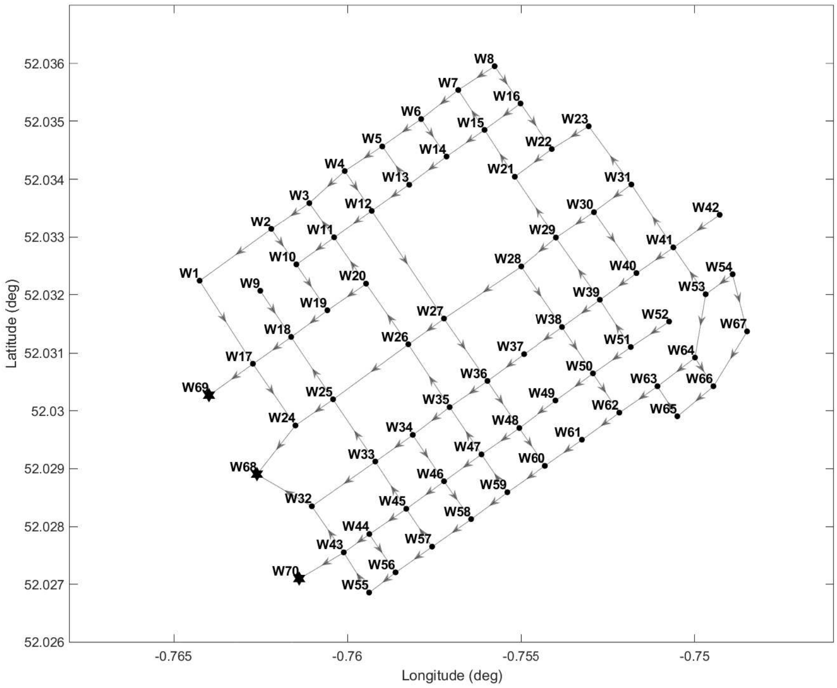


Figure 9. Westbound at 20 m, 3 retail points (W68, W69, W70) and 67 service points (W1∼W67) and 107 directed routes, (Oldbrook, Milton Keynes, United Kingdom).

Table 4. sUAS specification.

	Small UAS
Speed Range (km/h)	[5–25]
Endurance (s)	900
Separation (s)	5
Max Weight (kg)	25 (FAA Part 107)

We conduct 100 MC simulations for each outer loop algorithm (FCFS and LCFS algorithms) by increasing the number of sUASs from 5 to 50 with 5 s interval for the four drone delivery operation types (1-to-$\mathcal{M}$, $\mathcal{M}$-to-1, $\mathcal{N}$-to-$\mathcal{M}$, $\mathcal{M}$-to-$\mathcal{N}$) as shown in Table 5. At each simulation a service point and a retail point for each sUAS are randomly chosen except for the 1-to-$\mathcal{M}$ and $\mathcal{M}$-to-1 cases in which there is the single retail point E68 and W68, respectively. Once the route network and each sUAS' retail point and service point are given, which are the input of Algorithm 1, the algorithm generates each sUAS' flight plan as described in Section 3. From the output of the algorithm, we obtain M_{tft} (total flight time), M_{mct} (mission completion time), M_{tfd} (total flight distance), and M_{cnf} (normalised number of conflicts).

Table 5. Eight Monte Carlo simulation cases.

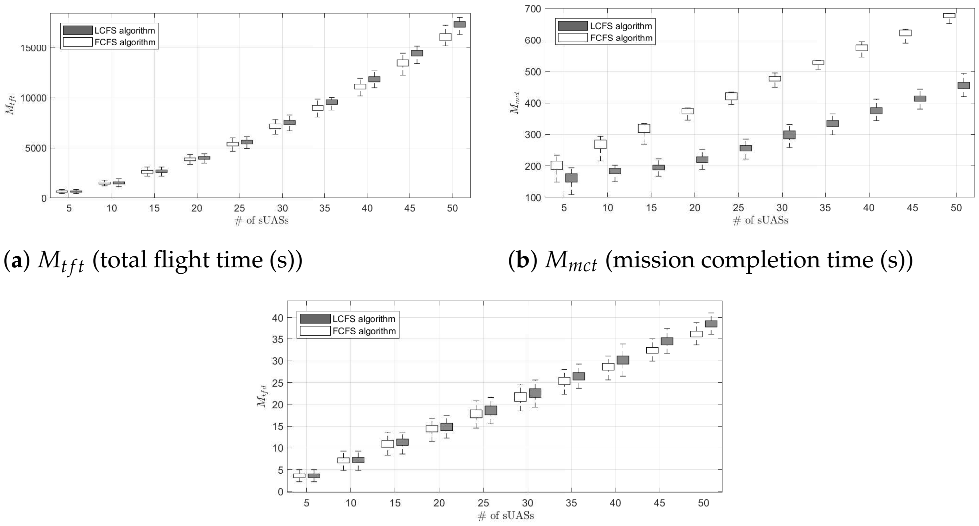
	FCFS Algorithm	**LCFS Algorithm**
1-to-$\mathcal{M}$ case	✓	✓
$\mathcal{M}$-to-1 case	✓	✓
$\mathcal{N}$-to-$\mathcal{M}$ case	✓	✓
$\mathcal{M}$-to-$\mathcal{N}$ case	✓	✓

Both the FCFS and LCFS algorithm results quantify the route network-based urban airspace based on the four metrics. For each operation type, we analyse both results, and suggest one of the two outer loop algorithms that is more appropriate for the operation type.

4.1. 1-to-$\mathcal{M}$ Last-Mile Delivery Case (Eastbound)

In the 1-to-$\mathcal{M}$ operation type, each sUAS traverses with an item from the single retail point (E68) to each sUAS' predetermined service point between E1 and E67 in Figure 8, where we disregard two retail points (E69 and E70) and two routes (E69→E17 and E70→E43).

In both of the outer loop algorithms, multiple sUASs traverse at maximum speed due to the five second time interval between each sUAS departure at the single retail point (E68). Thus, there is no significant difference between the two algorithms in M_{tft} and M_{tfd}, as shown in Figure 10a,c, respectively. Furthermore, the interval allows them to satisfy the separation requirement at all times, so there is no conflict between any pair of sUASs. However, Figure 10b shows that the LCFS algorithm results in more efficient mission completion times compared to the FCFS algorithm, which means the LCFS algorithm allows more sUASs to fly at a given time within the urban airspace. Therefore, in the operation type of 1-to-$\mathcal{M}$, the LCFS algorithm is more efficient than the FCFS algorithm in terms of total flight distance M_{mct}.



(**a**) M_{tft} (total flight time (s))

(**b**) M_{mct} (mission completion time (s))

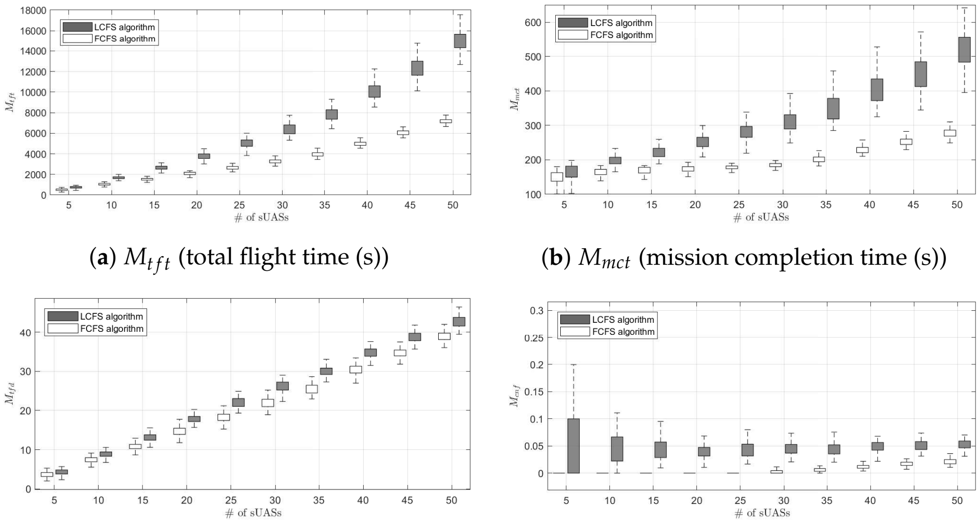
(**c**) M_{mct} (total flight distance (km))

Figure 10. 1-to-$\mathcal{M}$ Case: Standard deviation comparisons between the FCFS algorithm and the LCFS algorithm for M_{tft}, M_{mct}, and M_{tfd}.

4.2. $\mathcal{M}$-to-1 First-Mile Delivery Case (Westbound)

In the $\mathcal{M}$-to-1 operation type, multiple sUASs fly from their predetermined service points between W1 and W67 to the single retail point (W68) in Figure 9, where we ignore two retail points (W69 and W70) and two routes (W17→W69 and W43→W70). We assume that all sUASs depart from each sUAS' service point at the same time to consider the worst case scenario.

Figure 11c shows that M_{tfd} of the LCFS algorithm is slightly higher than the FCFS algorithm, and Figure 11a shows that M_{tft} is rapidly raised as the number of sUASs increases compared to M_{tfd}. As a result, it is seen that the LCFS algorithm causes not only detours but also low flight speeds. These results are inextricably linked to M_{cnf} shown in Figure 11d, resulting in inefficient and unpredictable results for M_{cnf}. In all respects, therefore, the FCFS algorithm is better than the LCFS algorithm for the $\mathcal{M}$-to-1 operation type for the urban airspace.



(**a**) M_{tft} (total flight time (s))

(**b**) M_{mct} (mission completion time (s))

(**c**) M_{tfd} (total flight distance (km))

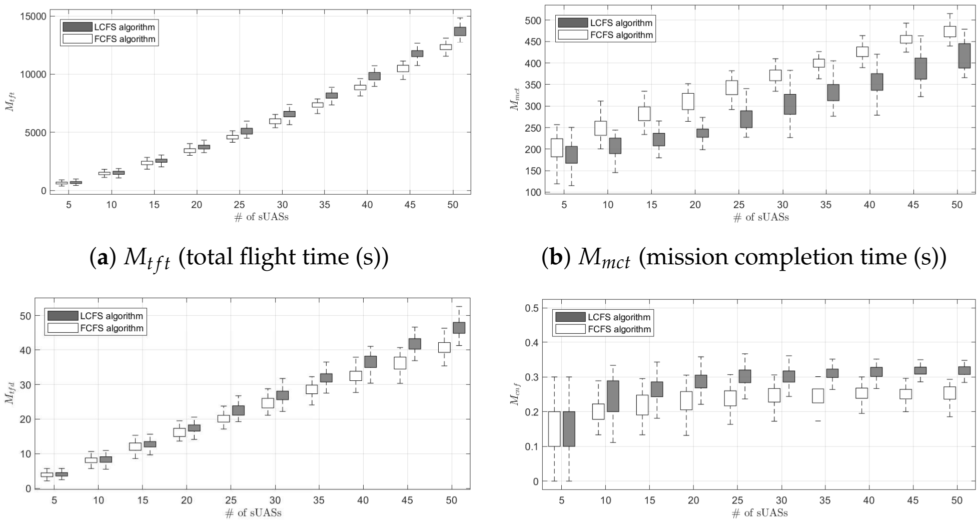
(**d**) M_{cnf}

Figure 11. $\mathcal{M}$-to-1 Case: Standard deviation comparisons between the FCFS algorithm and the LCFS algorithm for M_{tft}, M_{mct}, M_{tfd}, and M_{cnf}.

4.3. $\mathcal{N}$-to-$\mathcal{M}$ Last-Mile Delivery Case (Eastbound)

In the $\mathcal{N}$-to-$\mathcal{M}$ operation type, each sUAS traverses with an item from the predetermined retail points (E68, E69, and E70) to the each sUAS' predetermined service point from E1 to E67 in Figure 8. It is assumed that each retail point has five second departure intervals.

From the safety and flight performance perspective, the FCFS algorithm results in more efficient results for M_{tft}, M_{tfd}, and M_{cnf}, as shown in Figure 12. However, M_{mct} shows that the LCFS algorithm can allow to accommodate more sUASs in the given space within a given time. Therefore, in the case of $\mathcal{N}$-to-$\mathcal{M}$ for the urban airspace, the algorithm is dependent on the situation as shown in Figure 12. For example, it is appropriate to use the LCFS algorithm when a large number of items must be delivered in the given urban airspace, while the FCFS algorithm could be used for safety and individual performance when there is more room in the given urban airspace.



(**a**) M_{tft} (total flight time (s))

(**b**) M_{mct} (mission completion time (s))

(**c**) M_{tfd} (total flight distance (km))

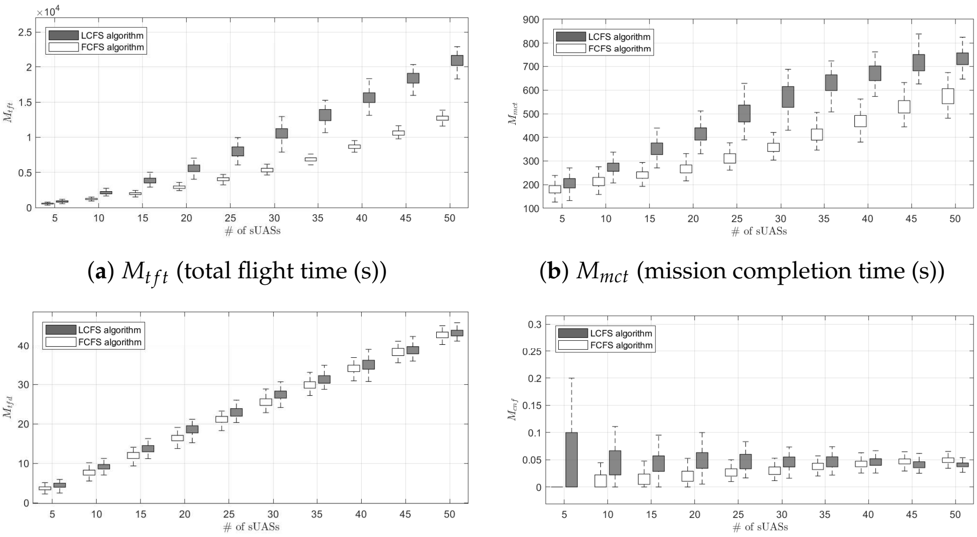
(**d**) M_{cnf}

Figure 12. $\mathcal{N}$-to-$\mathcal{M}$ Case: Standard deviation comparisons between the FCFS algorithm and the LCFS algorithm for M_{tft}, M_{mct}, M_{tfd}, and M_{cnf}.

4.4. $\mathcal{M}$-to-$\mathcal{N}$ First-Mile Delivery Case (Westbound)

In the $\mathcal{M}$-to-$\mathcal{N}$ operation type, each sUAS traverses with an item from each sUAS' predetermined service point between W1∼W67 to the predetermined retail points (W68, W69, W70) in Figure 9. We assume that all sUASs depart from each sUAS' service points at the same time to consider the worst case scenario.

Figure 13 shows that the results of M_{tft}, M_{tfd}, and M_{mct} are very similar to the results of the $\mathcal{M}$-to-1 operation type. Namely, the FCFS algorithm is appropriate for this case with regard to the three metrics. However, the M_{cnf} result of the LCFS algorithm approaches to 0.05, while in the FCFS algorithm results steadily increases. This is because of the overtaking manoeuvres caused by the outer loop algorithm. The FCFS algorithm finds flight plans while satisfying the minimum separation requirement when there are a small number of sUASs. However, as the number of sUASs increases, M_{cnf} is increasing because of a bottleneck phenomenon around the retail points. In the case of the LCFS algorithm, even when a small number of sUASs are flying, the M_{cnf} always shows a similar tendency because the following sUASs may outstrip the preceding sUASs. Because of the same reason, when there are a small number of sUASs in the urban airspace, the variation is large as shown in Figure 13d. Due to the tendency, the result of the FCFS algorithm shows a higher M_{cnf} than the LCFS algorithm's one from when the number of sUASs is 45. From this result, we suggest applying either the FCFS algorithm or LCFS algorithm to the airspace users according to the number of sUASs in the urban airspace.



(**a**) M_{tft} (total flight time (s))

(**b**) M_{mct} (mission completion time (s))

(**c**) M_{tfd} (total flight distance (km))

(**d**) M_{cnf}

Figure 13. $\mathcal{M}$-to-$\mathcal{N}$ Case: Standard deviation comparisons between the FCFS algorithm and the LCFS algorithm for M_{tft}, M_{mct}, M_{tfd}, and M_{cnf}.

4.5. The Improvement in M_{mct} and M_{cnf} between the FCFS and LCFS Algorithms

In order to complement Figures 10–13, we provide Table 6 to show the improvement in M_{mct} and M_{cnf} between the FCFS and LCFS algorithms with the following equations:

$$\text{Improvement in } M_{mct} = \frac{M_{mct}^{\text{FCFS}} - M_{mct}^{\text{LCFS}}}{M_{mct}^{\text{FCFS}}} \times 100. \tag{14}$$

$$\text{Improvement in } M_{cnf} = \frac{M_{cnf}^{\text{FCFS}} - M_{cnf}^{\text{LCFS}}}{M_{cnf}^{\text{FCFS}}} \times 100. \tag{15}$$

The metric, M_{cnf}, is the cause of increasing M_{tft} and M_{tfd}. Namely, M_{tft} and M_{tfd}, which also indicate sUASs' performance, are directly related to M_{cnf}. Mission completion time, M_{mct}, is an indicator of the performance of the urban airspace within a given period of time, although its relationship with M_{tft} and M_{tfd} is not prominent. From the results, the following analysis can be made: (1) For the four operation types, the FCFS algorithm generates more efficient results than the LCFS algorithm for M_{tft} and M_{tfd}. This is because sUASs with short flight times complete the flight first, and then the other sUASs can fly in relaxed urban airspace at high speed; (2) for the 1-to-$\mathcal{M}$ operation type, the LCFS algorithm shows very efficient results for M_{mct}. This indicates that the LCFS algorithm efficiently uses the urban airspace for a given time period; and (3) from the standard deviation of the results except for the 1-to-$\mathcal{M}$ operation type, the FCFS algorithm is easier to predict each metric as the number of sUASs increases than the LCFS algorithm. The main reason why the standard deviation of the LCFS algorithm results are high is that the variation of each sUAS' speed profile dependent on the randomly selected starting point and destination point is very large.

Table 6. The improvement in M_{mct} and M_{cnf} between the FCFS and LCFS algorithms (%).

# of sUASs		**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**	**50**
1-to-$\mathcal{M}$	M_{mct}	19.3	32.0	38.8	40.3	39.6	37.6	36.3	34.7	33.4	32.2
	M_{cnf}	0	0	0	0	0	0	0	0	0	0
$\mathcal{M}$-to-1	M_{mct}	−10.2	−21.1	−32.6	−45.2	−59.5	−68.9	−73.4	−76.4	−79.7	−86.0
	M_{cnf}	$-\infty$	$-\infty$	−45,300.0	−9475.0	−3600.0	−1281.8	−623.0	−297.5	−200.0	−146.5
$\mathcal{N}$-to-$\mathcal{M}$	M_{mct}	6.1	16.9	21.5	23.1	20.0	18.6	16.8	16.5	15.0	12.4
	M_{cnf}	−36.3	−20.6	−21.3	−22.8	−26.3	−23.3	−27.5	−24.4	−27.4	−27.3
$\mathcal{M}$-to-$\mathcal{N}$	M_{mct}	−15.6	−28.5	−46.4	−54.1	−59.0	−58.9	−51.9	−43.1	−34.9	−28.4
	M_{cnf}	−178.6	−181.9	−181.5	−131.1	−70.8	−51.0	−22.8	−7.7	11.7	18.2

5. Conclusions

This work investigated the capacity analysis for the route network-based very low-level urban airspace with regard to the four cases of drone delivery operations. Two different flight planning algorithms were compared for each operation type using the Monte Carlo simulations. From the analysis, the following implications can be made. Firstly, for efficient operation on the same operation type, flight planning should be carefully determined depending on operation factors such as the number of retail points, number of service points, number of sUASs, etc. Secondly, in this study, the FCFS and LCFS flight planning algorithms were used for homogeneous sUASs. For practical operations, however, it is required to consider the airspace users' priorities, i.e., non-cooperative operation cases with heterogeneous sUASs. Further simulation and verification methodologies will be necessary for future studies because decentralised cases are more complex than the centralised instances studied in this research. Finally, such analyses and results could be used for designing airspace structures depending on urban airspace situations and environments.

We expect that the results could give the following support to policymakers, urban airspace designers, and regulators: (1) When the stakeholders configure a new structure of the urban airspace, they can utilise the metrics to estimate the capacity according to the situation and environment; (2) it is easy to analyse the entry of new stakeholders by increasing the number of retail points; and (3) it is also available as a test-bed for new algorithms.

Author Contributions: Conceptualization, S.B., H.-S.S. and A.T.; methodology, S.B., H.-S.S. and A.T.; software, S.B.; validation, H.-S.S. and A.T.; formal analysis, H.-S.S. and A.T.; investigation, S.B., H.-S.S. and A.T.; resources, S.B., H.-S.S. and A.T.; data curation, S.B.; writing—original draft preparation, S.B. and H.-S.S.; writing—review and editing, H.-S.S.; visualization, S.B.; supervision, H.-S.S. and A.T.; project administration, H.-S.S.; funding acquisition, H.-S.S. and A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Amazon to begin drone deliveries in Lockeford, California this year. *BBC*, 14 June 2022.
2. Garrow, L.A.; German, B.J.; Leonard, C.E. Urban air mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research. *Transp. Res. Part C Emerg. Technol.* **2021**, *132*, 103377. [CrossRef]
3. Lavars, N. Amazon to begin testing new delivery drones in the US. *New Atlas*, 13 April 2015.
4. Amazon.com Inc. Determining Safe Access with a Best- Equipped, Best-Served Model for Small Unmanned Aircraft Systems. In *NASA UTM 2015: The Next Era of Aviation*; NASA: Washington, DC, USA, 2015.
5. Simmons, D. Rwanda begins Zipline commercial drone deliveries. *BBC World News*, 14 October 2016.
6. Thiels, C.A.; Aho, J.M.; Zietlow, S.P.; Jenkins, D.H. Use of unmanned aerial vehicles for medical product transport. *Air Med. J.* **2015**, *34*, 104–108. [CrossRef] [PubMed]
7. Haidari, L.A.; Brown, S.T.; Ferguson, M.; Bancroft, E.; Spiker, M.; Wilcox, A.; Ambikapathi, R.; Sampath, V.; Connor, D.L.; Lee, B.Y. The economic and operational value of using drones to transport vaccines. *Vaccine* **2016**, *34*, 4062–4067. [CrossRef]
8. Kopardekar, P. *Unmanned Aerial System (UAS) Traffic Management (UTM): Enabling Low-Altitude Airspace and UAS Operations*; NASA: Washington, DC, USA, 2014.
9. Hoekstra, J.; Kern, S.; Schneider, O.; Knabe, F.; Lamiscarre, B. *Metropolis—Urban Airspace Design*. Technical Report; TU Delft: Delft, The Netherlands, 2015.
10. Prevot, T.; Rios, J.; Kopardekar, P.; Robinson, J.E., III; Johnson, M.; Jung, J. UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations. In Proceedings of the 16th AIAA Aviation Technology, Integration, and Operations Conference, Washington, DC, USA, 13–15 June 2016; pp. 1–16. [CrossRef]
11. Bulusu, V.; Sengupta, R.; Polishchuk, V.; Sedov, L. Cooperative and non-cooperative UAS traffic volumes. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017, Miami, FL, USA, 13–16 June 2017; pp. 1673–1681. [CrossRef]
12. Sedov, L.; Polishchuk, V.; Bulusu, V. Sampling-based capacity estimation for unmanned traffic management. In Proceedings of the AIAA/IEEE Digital Avionics Systems Conference—Proceedings, St. Petersburg, FL, USA, 17–21 September 2017. [CrossRef]
13. Mohamed Salleh, M.F.B.; Wanchao, C.; Wang, Z.; Huang, S.; Tan, D.Y.; Huang, T.; Low, K.H. Preliminary Concept of Adaptive Urban Airspace Management for Unmanned Aircraft Operations. In Proceedings of the 2018 AIAA Information Systems-AIAA Infotech @ Aerospace, Kissimmee, FL, USA, 8–12 January 2018; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2018; pp. 1–12. [CrossRef]
14. Öreg, Z.; Shin, H.S.; Tsourdos, A. Traffic conflict reduction based on distributed stochastic task allocation. *Aeronaut. J.* **2022**, *126*, 993–1025. [CrossRef]
15. Rimjha, M.; Hotle, S.; Trani, A.; Hinze, N. Commuter demand estimation and feasibility assessment for Urban Air Mobility in Northern California. *Transp. Res. Part A Policy Pract.* **2021**, *148*, 506–524. [CrossRef]
16. Laudeman, I.V.; Shelden, S.G.; Branstrom, R.; Brasil, C.L. *Dynamic Density: An Air Traffic Management Metric*; NASA: Washington, DC, USA, 1998.
17. Majumdar, A.; Ochieng, W.; Polak, J. Estimation of European Airspace Capacity from a Model of Controller Workload. *J. Navig.* **2002**, *55*, 381–403. [CrossRef]
18. Majumdar, A.; Ochieng, W.Y.; Bentham, J.; Richards, M. En-route sector capacity estimation methodologies: An international survey. *J. Air Transp. Manag.* **2005**, *11*, 375–387. [CrossRef]
19. Klein, A.; Cook, L.; Wood, B.; Simenauer, D. Airspace capacity estimation using flows and weather-impacted traffic index. In Proceedings of the 2008 Integrated Communications, Navigation and Surveillance Conference, ICNS Conference 2008, Bethesda, MD, USA, 5–7 May 2008. [CrossRef]
20. Qinetiq, A.J.H.; Care, E.; Metrics, I. Assessing the Capacity of Novel ATM Systems. In Proceedings of the 4th USA/Europe Air Traffic Management R&D Seminar, Santa Fe, NM, USA, 4–7 December 2001.

21. Öreg, Z.; Shin, H.S.; Tsourdos, A. On the underlying dynamics of traffic conflicts related to stochastic behaviour. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2022**, *237*.
22. Bulusu, V.; Polishchuk, V.; Sengupta, R.; Sedov, L. Capacity Estimation for Low Altitude Airspace. In Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference, Denver, CO, USA, 5–9 June 2017; pp. 1–15. [CrossRef]
23. Sedov, L.; Polishchuk, V. Centralized and Distributed UTM in Layered Airspace. In Proceedings of the 8th International Conference on Research in Air Transportation, ICRAT 2018, Catalonia, Spain, 25–29 June 2018; pp. 1–8.
24. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle Routing Problems for Drone Delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 70–85.
25. Mohamed Salleh, M.F.B.; Low, K.H. Concept of Operations (ConOps) for Traffic Management of Unmanned Aircraft Systems (TM-UAS) in Urban Environment. In Proceedings of the AIAA Information Systems-AIAA Infotech @ Aerospace, Grapevine, TX, USA, 9–13 January 2017; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2017; pp. 1–13. [CrossRef]
26. Krozel, J.; Peters, M.; Bilimoria, K.D.; Lee, C.; Mitchell, J.S. System Performance Characteristics of Centralized and Decentralized Air Traffic Separation Strategies. *Air Traffic Control Q.* **2001**, *9*, 311–332. [CrossRef]
27. Ballin, M.; Hoekstra, J.; Wing, D.; Lohr, G. NASA Langley and NLR Research of Distributed Air/Ground Traffic Management. In Proceedings of the AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum, Los Angeles, CA, USA, 1–3 October 2002; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2002. [CrossRef]
28. Klooster, J.; Torres, S.; Earman, D.; Castillo-Effen, M.; Subbu, R.; Kammer, L.; Chan, D.; Tomlinson, T. Trajectory synchronization and negotiation in Trajectory Based Operations. In Proceedings of the 29th Digital Avionics Systems Conference, Salt Lake City, UT, USA, 3–7 October 2010; pp. 1.A.3-1–1.A.3-11. [CrossRef]
29. Wichman, K.; Lindberg, L.; Kilchert, L.; Bleeker, O. Four-Dimensional Trajectory Based Air Traffic Management. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2004. [CrossRef]
30. Jang, D.S.; Ippolito, C.A.; Sankararaman, S.; Stepanyan, V. Concepts of Airspace Structures and System Analysis for UAS Traffic flows for Urban Areas. In Proceedings of the AIAA Information Systems-AIAA Infotech @ Aerospace, Grapevine, TX, USA, 9–13 January 2017. [CrossRef]
31. Sunil, E.; Ellerbroek, J.; Hoekstra, J.; Vidosavljevic, A.; Arntzen, M.; Bussink, F.; Nieuwenhuisen, D. Analysis of Airspace Structure and Capacity for Decentralized Separation Using Fast-Time Simulations. *J. Guid. Control Dyn.* **2017**, *40*, 38–51. [CrossRef]
32. Bae, S.; Shin, H.S.; Antonios, T. A New Graph-Based Flight Planning Algorithm for Unmanned Aircraft System Traffic Management. In Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, UK, 23–27 September 2018.
33. Bae, S.; Shin, H.S.; Lee, C.H.; Antonios, T. A New Multiple Commercial Aircraft Routing and Scheduling Algorithm in Terminal Manoeuvring Area. In Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, UK, 23–27 September 2018.
34. Bortoff, S. Path planning for UAVs. In Proceedings of the 2000 American Control Conference, ACC (IEEE Cat. No.00CH36334), Chicago, IL, USA, 28–30 June 2000; Volume 1, pp. 364–368. [CrossRef]
35. Enright, J.; Frazzoli, E.; Savla, K.; Bullo, F. On Multiple UAV Routing with Stochastic Targets: Performance Bounds and Algorithms. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA, USA, 15–18 August 2005. [CrossRef]
36. Lamont, G.B.; Slear, J.N.; Melendez, K. UAV swarm mission planning and routing using multi-objective evolutionary algorithms. In Proceedings of the IEEE Symposium Computational Intelligence in Multicriteria Decision Making, Honolulu, HI, USA, 1–5 April 2007; pp. 10–20. [CrossRef]
37. Hall, J.; Anderson, D. Reactive route selection from pre-calculated trajectories—Application to micro-UAV path planning. *Aeronaut. J.* **2011**, *115*, 635–640. [CrossRef]
38. De Filippis, L.; Guglieri, G.; Quagliotti, F. Path Planning Strategies for UAVS in 3D Environments. *J. Intell. Robot. Syst.* **2012**, *65*, 247–264. [CrossRef]
39. Liu, W.; Zheng, Z.; Cai, K.Y. Bi-level programming based real-time path planning for unmanned aerial vehicles. *Knowl.-Based Syst.* **2013**, *44*, 34–47. [CrossRef]
40. Altmann, A.; Niendorf, M.; Bednar, M.; Reichel, R. Improved 3D Interpolation-Based Path Planning for a Fixed-Wing Unmanned Aircraft. *J. Intell. Robot. Syst.* **2014**, *76*, 185–197. [CrossRef]
41. Radmanesh, M.; Kumar, M. Flight formation of UAVs in presence of moving obstacles using fast-dynamic mixed integer linear programming. *Aerosp. Sci. Technol.* **2016**, *50*, 149–160. [CrossRef]
42. Manyam, S.G.; Rasmussen, S.; Casbeer, D.W.; Kalyanam, K.; Manickam, S. Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017, Miami, FL, USA, 13–16 June 2017; pp. 573–580.
43. Yu, K.; Budhiraja, A.K.; Tokekar, P. Algorithms for Routing of Unmanned Aerial Vehicles with Mobile Recharging Stations. *arXiv* **2017**, arXiv:1704.00079.
44. Aggarwal, R.; Soderlund, A.A.; Kumar, M. Multi-UAV Path Planning in a Spreading Wildfire. In Proceedings of the AIAA Scitech 2021 Forum, Virtual, 11–15 & 9–21 January 2021; p. 0866.

45. Zhou, X.; Gao, F.; Fang, X.; Lan, Z. Improved Bat Algorithm for UAV Path Planning in Three-Dimensional Space. *IEEE Access* **2021**, *9*, 20100–20116. [CrossRef]
46. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [CrossRef]
47. Wong, R.T. Combinatorial Optimization: Algorithms and Complexity (Christos H. Papadimitriou and Kenneth Steiglitz). *SIAM Rev.* **1983**, *25*, 424–425. [CrossRef]
48. Yen, J.Y. Finding the K Shortest Loopless Paths in a Network. *Manag. Sci.* **1971**, *17*, 712–716. [CrossRef]
49. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.