Abstract—We address multiple flights planning problems from its initial waypoint to its destination while satisfying the minimum separation requirement between each aircraft at all times in a Terminal Manoeuvring Area (TMA) to maintain or increase runway throughput. Due to operational constraints for safety, most of the current aircraft fly over or by waypoints, and along nominal routes in the airspace. Where the waypoints and routes in the airspace can be modelled as a weighted digraph, called airspace graph. We propose a problem that consists of determining a flight path (routing problem) and its speed profile (scheduling problem) in a given airspace graph in which a time-based weighting scheme of the airspace graph is proposed to reflect a speed-limitation-compliant schedule that satisfy the minimum separation requirement. For multiple flights cases, the flight paths and schedules are obtained by iteratively solving the problem for each flight by applying the First Come First Served (FCFS) algorithm to determine an arrival sequence. The main contributions of this paper are increasing a solution search space by solving two problems simultaneously, efficient computational time, and providing the separation-compliant flight path and speed profile within the speed limitation for each flight. We demonstrate the advantages of the proposed approach through a case study in which multiple flights arrive at a single airport, and we compare the results with Regulated Tactical Flight Model (RTFM) obtained from EUROCONTROL Demand Data Repository 2 (DDR2). Although, we consider only a single airport and make an assumption to simplify flight routes from holding stacks to a Final Approach Fix (FAF), the results show the potential usage of the proposed algorithm as a Decision Support Tool (DST) for Air Traffic Controllers (ATCOs) if the potential usage of the proposed algorithm as a Decision Support Tool (DST) for Air Traffic Controllers (ATCOs) if the

Keywords—routing and scheduling; graph theory; time-based separation; terminal manoeuvring area

I. INTRODUCTION

Since air traffic demand increases continuously, modernisation projects SESAR, and NextGen [1], [2], and a considerable amount of research [3]–[16] for the future Air Traffic Management (ATM) system have been conducted to improve safety and efficiency of flights. In future ATM environment, the operation of air traffic consisted of route networks requires more detailed information such as separation-compliant speed profile for each flight than the current ATM system because the flights will follow aRea NAVigation (RNAV) and Required Navigation Performance (RNP) procedures in the networks [10]. Typically, scheduling means determining an arrival sequence with time slots in the flight planning stage, but the terminology of scheduling for this study is for determining a separation-compliant speed profile, and an arrival time at each waypoint (i.e., an arrival sequence) in the flight planning stages. Thus, the schedule helps Air Traffic Controllers (ATCOs) manage safe and efficient operations in the tactical planning stage. The terminology of routing for this study is for determining a flight path which consisted of a set of linear segments (nominal routes).

The requirements lead to an entire class of routing and scheduling problem, which depends on airspace configuration, aircraft performance constraints, and operational constraints. For example, a Terminal Manoeuvring Area (TMA), where is a dense and complex route network compared to an en-route area, requires heavy workload of ATCOs in charge of routing and scheduling for each flight in their control area and assuring separation between each pair of aircraft. ATCOs resolve these problems based on their experience, intuition and some scheduling rules without using formally defined performance indices [16]. Without mathematical formulations or quantitative metrics, however, it is difficult to obtain meaningful results in post-analyses to improve the airspace efficiency and safety, although the system works well.

A great deal of previous research into the issue has focused on resolving the routing and scheduling problem using optimisation techniques. Determining a flight path consisting of a finite set of waypoints in the airspace network is often modelled as a directed graph, and a schedule as a chart
that optimises the estimated time of arrival at each of the waypoints [3]–[8]. Such approaches only consider the flight schedule at a runway, not entire speed profiles. One limitation of the model is that the scheduling model does not take into account the minimum separation during the flight, which is one of the most important operational requirements for safety without adding artificial constraints. Recent literature [6], [17] has attempted to model aircraft scheduling problems using a Mixed-Integer Linear Programming (MILP) and a Dynamic Programming (DP) to maximise runway throughput while enforcing minimum separation in time. However, the drawback of such approaches is that it is highly computational demanding [18], [19], although these are scalable in the number of aircraft. More recent works [9], [11]–[15] have attempted to address the routing and scheduling problems separately or sequentially. These approaches have advantages such as fast computational time, proposing a mathematical model, proving the existence, and determination of separation-compliant speed. However, we expect that if two problems are solved separately or sequentially, the scheduling problem satisfying the separation may degrade the optimality of routing results.

Our main idea is that if the Estimated Time of Arrival (ETA) at each waypoint for each flight can be shared with each other, routing and scheduling solutions satisfying the minimum separation can be formulated using a weighted directed graph and may show computational performance suitable for real-time applications in the flight planning stage. Then, we would obtain a route and a schedule of each flight simultaneously by solving the shortest path of the airspace graph by means of any shortest path algorithms. Thus, we could find solutions in a larger solution space than the existing approaches.

In this paper, we propose a novel algorithm based on the weighted directed graph to solve a routing and scheduling problem for a single flight to be planned. Data of planned flights are utilised to calculate the weight expressed in flight time for each edge of the graph to satisfy the time-based separation requirement in which the objective function of the weights is to minimise the total flight time. Thus, a flight path found using the shortest path algorithm includes a speed profile that minimises flight time. The speed profile sought here is a separation-compliant speed advisory for ATCOs as the speed profiles assure the minimum separation at all times. Furthermore, we can obtain the multiple flight’s flight paths and speed profiles by iteratively solving the routing and scheduling problem for each flight with the FCFS algorithm to determine an arrival sequence.

We expect the following merits of the proposed algorithm; 1) by simultaneously addressing the routing and scheduling problem while satisfying the minimum separation requirement, the algorithm could mitigate the disadvantage of sequential approaches (i.e., latter stage (scheduling and/or separation) may degrade the optimality of the former results (routing and/or scheduling)); 2) as the algorithm can significantly improve computational performance compared to the previous research, we highly expect that it can be utilised as an automated real-time DST that helps ATCOs’ duties; and 3) it is capable of capturing realistic aspects of the ATM system such as time-based separation, speed restrictions, nominal routes and waypoints, etc.

The rest of this paper is organised as follows. Section II describes the problem formulation for single flight routing and scheduling. Section III proposes the algorithm for multiple flights routing and scheduling. A case study is given in Section IV. Section V discusses the effectiveness of the proposed algorithm and further works. Section VI provides the conclusions of this study.

II. PROBLEM STATEMENT FOR SINGLE FLIGHT ROUTING AND SCHEDULING

This section consists of five subsections: (a) a description of airspace networks for operational purposes; (b) a description of a time-based separation concept; (c) assumptions considered in this study; (d) mathematical modelling for the routing and scheduling problem; (e) a description of the flight-time-based weighting scheme.

A. Airspace network

For ATM operational purposes, airspace is often considered as a region with a set of waypoints. Some pairs of waypoints are connected, and these are called routes. STandard Arrival Routes (STARs) provide waypoints and routes with other information that airspace users should follow such as altitude constraints, track angle, etc. For the routing problem, we use the routes published in STARs instead of using new potential routes such as those proposed in [20].

B. Time-Based Separation

Conflict detection is activated when the separation (time-based separation or distance-based separation) of two aircraft is less than a minimum separation criterion. In this study, we utilise the time-based separation that is determined by the International Civil Aviation Organisation (ICAO) Wake Turbulence Category (WTC) of the leading aircraft and the following aircraft as shown in Table I instead of the widely used distance-based separation (e.g., 5 NM pre-decided by International Civil Aviation Organisation Doc 4444) to stabilise the time spacing between arrival pairs of aircraft across headwind conditions to recover the lost landing rate currently experienced [21], [22]. By satisfying the separation assurance adjusting the Estimated Time to Arrival (ETA), we define it as the Separation Assured Estimated Time of Arrival (SAETA), at each waypoint rather than adjusting the speed on the routes, we obtain results that satisfy the minimum separation at each waypoints as well as the separation at the routes. The concept allows the time-based separation to be satisfied at merging points as shown in Fig. 1. In the problem formulation, we optimise an aircraft speed profile where time data containing the SAETA at each waypoint, $T(v)$, must be known for this concept. Here we assume that the speed profile is a set of constant values, where the form of the speed profile is the same as one in flight plans.
Figure. 1: Time-based separation concept (aircraft 1, \( \alpha_1 \), and aircraft 2, \( \alpha_2 \), fly from its origin \( A \) and \( B \) to the same destination \( C \) via a merging point \( M \) at the speed \( \{ s_{11}, s_{12} \} \) and \( \{ s_{21}, s_{22} \} \), respectively. The superscript and subscript of \( s \) are aircraft index and segment index, respectively)

At the time \( t_0 \) of Fig. 1, aircraft 1, \( \alpha_1 \), and aircraft 2, \( \alpha_2 \), fly toward the waypoint \( C \) through the same merging point, \( M \), at the speeds \( s_{11} \) and \( s_{12} \), respectively. The time \( t_1 \) when \( \alpha_2 \) just passes through \( M \) at the speed of \( s_{21} \) is stored at the merging point \( T(M) \), and no aircraft can pass through this point for \( t_{SP} \) seconds before and after \( t_1 \). The time \( t_2 \) is stored in \( T(M) \) in the same way. Then, \( \alpha_1 \) and \( \alpha_2 \) traverse toward \( C \) at the speeds of \( s_{12} \) and \( s_{22} \), respectively. If \( \alpha_1 \) and \( \alpha_2 \) fly from \( M \) to \( C \) at the speeds of \( s_{12} \) and \( s_{22} \), respectively, the separation between two aircraft will always be met the minimum separation requirement or greater than that on the route between \( M \) and \( C \).

\[
T(M) = \{0s, 120s, 210s\} \tag{1}
\]

The generic reference time-based separation depending on the following and leading aircraft is used for setting up \( t_{SP} \) as shown in Table I. Time data at each waypoint obtained such as (1) is used to calculate (8)-(13) that will be described in Section.III.

### Table I

<table>
<thead>
<tr>
<th>Leader</th>
<th>A380</th>
<th>Heavy</th>
<th>Medium</th>
<th>Light</th>
</tr>
</thead>
<tbody>
<tr>
<td>A380</td>
<td>60s</td>
<td>145s</td>
<td>167s</td>
<td>189s</td>
</tr>
<tr>
<td>Heavy</td>
<td>60s</td>
<td>98s</td>
<td>122s</td>
<td>145s</td>
</tr>
<tr>
<td>Medium</td>
<td>60s</td>
<td>60s</td>
<td>60s</td>
<td>122s</td>
</tr>
<tr>
<td>Light</td>
<td>60s</td>
<td>60s</td>
<td>60s</td>
<td>60s</td>
</tr>
</tbody>
</table>

### Assumptions

The following assumptions are made for this study: (a) each flight path is represented by a series of linear segments as shown in Fig. 2; (b) the speed of each segment is considered as a constant value; (c) aircraft speed applied in this paper is True AirSpeed (TAS); (d) arrival sequence is undetermined; and (e) uncertainties produced by external sources are neglected.

### Mathematical modelling

Our focus is on routing and scheduling a finite set

\[
\mathcal{A}^+ = \{ \alpha_1, \alpha_2, \ldots, \alpha_i \} \tag{2}
\]

of \( i \) flights in a TMA, each aircraft \( \alpha_i \in \mathcal{A}^+ \) to fly from its initial waypoint to its destination (runway), both specified as an input to the problem within feasible speed ranges and nominal routes. The route network in the TAM is modelled as a directed graph \( \mathcal{G} = (\mathcal{E}, \mathcal{V}) \), called 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 aircraft \( \alpha_i \) in the airspace graph \( \mathcal{G} = (\mathcal{E}, \mathcal{V}) \) is defined as follows:

**Definition 1.** (D1) As given in the airspace graph \( \mathcal{G} = (\mathcal{E}, \mathcal{V}) \), an aircraft \( \alpha_i \) to be routed in \( \mathcal{G} \), and the initial waypoint \( v_{i1} \in \mathcal{V} \) and the final waypoint \( v_{in} \in \mathcal{V} \), a flight path denoted by \( p(\alpha_i) \) in \( \mathcal{G} = (\mathcal{E}, \mathcal{V}) \) is defined by a sequence of waypoints.

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

\[
p(\alpha_i) : v_1^i, v_2^i, \ldots, v_{n_i}^i
\]  

(3)

For each flight’s path as given in (4), there is a set of routes

\[
\mathcal{E}(p(\alpha_i)) : e_{(1,2)}^i, e_{(2,3)}^i, \ldots, e_{(n_i-1,n_i)}^i
\]  

(4)

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

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

(5)

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

\[
T(p(\alpha_i)) = \sum_{j=1}^{n_i-1} w_j^i
\]  

(6)

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

\[
p^*(\alpha_i) = \arg\min_{p(\alpha_i) \in \mathcal{C}} T(p(\alpha_i))
\]  

(7)

The optimal path \( p^*(\alpha) \) can be found by using the well-known shortest path algorithms such as Dijkstra’s algorithm or the exhaustive search algorithm [18]. Although the optimal path \( p^*(\alpha_i) \) can be obtained according to D2, the optimality of the flight path might be disturbed in the following scheduling and conflict resolution stages to satisfy the minimum separation. Such a sequential approach can cause not only optimality issues but also high levels of workload to ATCOs.

Under the considered assumptions in the study, maximising runway throughput is equivalent to minimise flight times. Our main idea for achieving the objective is to assign a weight to each route (edge) of the airspace graph that expresses the flight time. Then, aircraft’s schedule can be obtained by finding the optimal path in the airspace graph of which weights are the flight time. In this study, we set a flight time of the aircraft as weight \( w \in \mathcal{W} \) to route \( e \in \mathcal{E} \) of the airspace graph \( \mathcal{G} = (\mathcal{E}, \mathcal{V}, \mathcal{W}) \). Note that, the flight time and the speed of aircraft are mutually interchangeable since we have geographic data of the airspace graph from STARs.

Another issue we have pursued is to satisfy the minimum separation requirement between each pair of aircraft. To assign a flight time that satisfies the minimum separation to each edge of \( \mathcal{G} \), we need time data including the SAETA for each waypoint discussed in Section II-B. 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 a solution of the airspace graph \( \mathcal{G} \), then, we can obtain the optimal flight path and its schedule while satisfying the minimum separation simultaneously. Each flight \( \alpha_i \in \mathcal{A}^+ \) might have different weights \( \mathcal{W}_{\alpha_i} \) because of the different start point, and different specifications of each aircraft. Thus, aircraft \( \alpha_i \) might have its unique airspace graph \( \mathcal{G}_{\alpha_i} \) as shown in Fig.3. In the following Section II-E, we describe the weight scheme to be applied to airspace graphs \( \mathcal{G}_{\alpha_i} \).

**E. Calculating weights on \( \mathcal{G}_{\alpha_i} \) suitable for separation-compliant speed profiles**

This subsection describes a separation assured flight-time-based weight scheme for the airspace graph. The weights can be easily converted to separation-compliant speed profiles to support the ATCOs’ decision making. An Non-Linear Programming (NLP) problem for calculating the weights is then defined follows:

\[
\min \sum_{j=1}^{n_i-1} \frac{d_j^i}{s_j^i}
\]  

(8)

s.t. \[
d_1^i + d_2^i + \ldots + d_{n_i-1}^i \geq \max T(v_{n_i}^i) + t_{SP} \]  

(9)

\[
d_1^i + d_2^i + \ldots + d_{n_i-2}^i \geq \max T(v_{n_i-1}^i) + t_{SP} \]  

(10)

\[ \vdots \]  

\[
d_1^i \geq \max T(v_3^i) + t_{SP} \]  

(11)

\[
d_1^i \geq \max T(v_2^i) + t_{SP} \]  

(12)

\[
s_{\min} \leq s_1^i, s_2^i, \ldots, s_{n_i-1}^i \leq s_{\max} \]  

(13)

where \( s_{\min}, s_{\max} \), and \( t_{SP} \) are the aircraft minimum speed, the aircraft maximum speed, and the minimum separation time between aircraft at waypoints as shown in Fig. 1.

The objective function of the NLP problem is to minimise the flight time (8). The \( n_i - 1 \) constraints (9)-(12) are for satisfying the minimum separation at each waypoint. As shown in Fig. 2, for example, if \( \mathcal{E}(p(\alpha_i)) \) consists of \( n_i - 1 \) segments
for aircraft $\alpha_i$, each segment requires a flight time that satisfies the time-based separation. Namely, the number of constraints for the separation is equal to the number of segments. The left hand side of the constraints is the total flight time to the end point of the final segment of each constraint, which must be greater than and equal to the time that satisfies the minimum separation.

For the constraints, route distance $d$ between waypoints, and the time data $T$ stored in $G_i$ are required. Flight distances $d(e)$ for route $e \in E$ in STARS are calculated by using the Harversine formula [24]. The time $T(v_{n_i})$, which indicates the SAETA that aircraft $\alpha_i$ fly over a waypoint $v_{n_i}$. The SAETA is stored in $T$ for waypoint $v \in V$, and $T$ is updated every time a flight is routed and scheduled, and $T$ is shared with aircraft $\alpha_i \in A^+$. In this study, we use a Sequential Quadratic Programming (SQP) method to solve the NLP problem.

Decision variables, $s_1, s_2, \ldots, s_{n_i-1}$, are constant aircraft speeds for a set of segments. The aircraft speeds can be converted into the flight time using the following:

$$\text{flight time} = \frac{d(\alpha_i)}{s(\alpha_i)} \quad (14)$$

where $s(\alpha_i)$ and $d(\alpha_i)$ are a set of constant speeds and a set of flight distances, respectively.

$$s(\alpha_i) : s_1, s_2, \ldots, s_{n_i-1} \quad (15)$$

$$d(\alpha_i) : d_1, d_2, \ldots, d_{n_i-1} \quad (16)$$

and the flight times are assigned into weights $W_{i,t}$. Therefore, a solution of the airspace graph $G_i = (E, V, W_{i,t}, W_{d}, T)$ can simultaneously provide the optimal route and its schedule satisfying the minimum separation requirement at all times. A speed profile shown in (15) derived from the solution can be used as a separation-compliant speed profile. The permissible speed range of each aircraft is only a function of the aircraft type (13) [25].

We also construct flight distance weights $W_d$ on $G_i$ to reflect the flight distance as a second criterion. The weights are necessary to prioritise for the multiple flights routing and scheduling problem when two or more aircraft arrive at the runway at the same time, more details in Section III. In summary, the problem that motivated this study can be formulated as follows:

**Problem 1. (P1)** Given an airspace graph $G_i = (E, V, W_{i,t}, W_d, T)$, flight $\alpha_i$ to be routed and scheduled in $G_i$, and its initial waypoint $v_i^1 \in V$ and its destination $v_{n_i}^t \in V$, reachable from the origin, of each flight $\alpha_i \in A^+$, constructs a flight path $p(\alpha_i)$ and a speed profile $s(\alpha_i)$ of $A^+$ in $G_i$ such that

- the separation requirement is satisfied from the planned flights,
- the speed profile of each flight must be within its feasible speed range,
- the airspace graphs are updated every time when flight $\alpha_i \in A^+$ finds its plan.

In this paper, we solve the P1 using Dijkstra’s algorithm [26].

**III. PROPOSED ALGORITHM FOR MULTIPLE FLIGHTS ROUTING AND SCHEDULING PROBLEMS**

Scheduling of arrival aircraft can be divided into three stages: initial sequencing stage, modifying schedule stage, and freezing stage [27]. Generally, for the Aircraft Landing Problem (ALP), the initial sequencing stage is based on the FCFS algorithm, which is the landing order that would be resulted, if each flight proceeded to the runway and landed without due consideration of other flights. This approach, however, causes many modifications and an increased workload to ATCOs in the next modifying schedule stage. The algorithm considering separation between every pair of aircraft is proposed based on the FCFS algorithm to reduce the workload of ATCOs by reducing the difference between the stages.

We propose A1 to solve the multiple flights routing and scheduling problem while satisfying the minimum separation requirement by iteratively solving P1 with the FCFS algorithm. The proposed algorithm proceeds as follows: (Line 4) given an instance of the problem, our approach first computes the separation-compliant flight path and schedule for each flight in a set of flights to be planned $A^+$ using Dijkstra’s algorithm (in this stage of every iteration, the meaning of separation means the minimum separation requirement from the planned flights $A^-$); (Line 5) the FCFS algorithm is used to find a flight that can maximise the runway throughput; (Line 7-8) if the performance index which is the runway arrival time of two or more flights are the same time, the second criterion, which is flight distance, is applied to determine the flight to arrive; (Line 10-11 or Line 14-15) the planned flight is removed from $A^+$ and stored in a set of planned flight $A^-$, and the route and schedule of the planned flight are shared for aircraft in $A^+$; (Line 18) based on the shared data, our approach updates P1 for unplanned flights $A^+$. Once the inputs of the algorithm are given, the algorithm runs the routing and scheduling process of the multiple flights in the TMA until $A^+$ is empty and, therefore, $A^-$ is full.

We illustrate an example in Fig. 4 for a better understanding of the iteration concept of the algorithm. In the first iteration, each flight $\alpha_1 \in A^+ = \{\alpha_1,\alpha_1,\ldots,\alpha_A\}$ has its airspace graph, and flight $\alpha_2$ is planned as the first flight. In the second iteration, the airspace graph of each flight $\alpha_1 \in A^+$ is updated, and flight $\alpha_A$ is planned as the second flight. In the $(A-1)^{th}$ iteration, flight $\alpha_1$ is planned, then flight $\alpha_2$ is planned in the same way.

**IV. CASE STUDY**

To validate the proposed algorithm, we conduct a case study of multiple flights routing and scheduling in the London TMA (LTMA). As a preliminary case study, we make the following operational assumptions: (1) only London Heathrow airport (LHR) of the five airports in the LTMA is considered; (2) only medium type of aircraft is considered; (3) routes from
Algorithm 1: (A1) Multiple flights routing and scheduling algorithm in a TMA

Input: \( v_1^i, v_d^i \in \mathcal{V}, \forall \alpha_i \in \mathcal{A}^+ \), airspace information
Output: \( p(\alpha_i), s(\alpha_i), \forall \alpha_i \in \mathcal{A}^- \)

1. \( k = 1 \)
2. generate \( G_i^k = (\mathcal{E}, \mathcal{V}, W_d^i, W_i, T^K) \), \( \forall \alpha_i \in \mathcal{A}^+ \)
3. while \( \mathcal{A} \neq \emptyset \) do
   4. find \( p^*(\alpha_i) \) of \( G_i^k = (\mathcal{E}, \mathcal{V}, W_d^i, W_i, T^K) \), \( \forall \alpha_i \in \mathcal{A}^+ \)
   5. \( \alpha_i^* \leftarrow \arg\min p^*(\alpha_i) \) (using the FCFS algorithm)
   6. if There are more than two \( p^*(\alpha_i) \) exist then
      7. find \( p^*(\alpha_i) \) of \( G_i^k = (\mathcal{E}, \mathcal{V}, W_d^i, W_i, T^K) \) amongst them (using Dijkstra's algorithm)
      8. \( \alpha_i^* \leftarrow \arg\min p^*(\alpha_i) \) (using the FCFS algorithm)
      9. allocate \( \alpha_i^* \); 
     10. remove \( \alpha_i^* \) from \( \mathcal{A}^+ \) and store \( \alpha_i^* \) in \( \mathcal{A}^+ \) with \( p^*(\alpha_i^*) \) and \( s(p^*(\alpha_i^*)) \); 
     11. share \( \mathcal{A}^+ \) with \( \forall \alpha_i \in \mathcal{A}^+ \);
   else
     12. allocate \( \alpha_i^* \);
     13. remove \( \alpha_i^* \) from \( \mathcal{A}^+ \) and store \( \alpha_i^* \) in \( \mathcal{A}^+ \) with \( p^*(\alpha_i^*) \) and \( s(p^*(\alpha_i^*)) \);
     14. share \( \mathcal{A}^- \) with \( \forall \alpha_i \in \mathcal{A}^- \);
   end
   15. \( k = k + 1 \);
   16. update \( G_i^k = (\mathcal{E}, \mathcal{V}, W_d^i, W_i, T^K) \), \( \forall \alpha_i \in \mathcal{A}^+ \);
end

the holding stacks to LHR are straight lines (namely, there are no holding manoeuvres); and (4) single runway is considered. Results will show the efficiency of our approach and the possibility of ATCOs' DST by providing a flight path and a schedule for each flight. Also, the results will be converted to the number of aircraft that can arrive at LHR by the medium type of aircraft per hour, and compared with Regulated Tactical Flight Model (RTFM) obtained from EUROCONTROL DDR2 [28].

London Heathrow (EGLL) STARs in the United Kingdom Aeronautical Information Publication (AIP) [29] are applied to construct an airspace graph, as the flights obtained using the algorithm only follow the nominal routes. We construct the airspace graph as shown in Fig. 5, where we simplify the routes from the four holding stacks (BNN, LAM, OCK, BIG) including the Initial Approach Fix (IAF) to the Final Approach Fix (FAF) because of the following practical issues: most of the arrival flights to LHR in peak time holds at the four holding stacks until pilots receive an ATC clearance to maintain the runway throughput.

<table>
<thead>
<tr>
<th>Airspace graph information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport</td>
</tr>
<tr>
<td>STARS</td>
</tr>
<tr>
<td># of runways</td>
</tr>
<tr>
<td># of fixes (waypoints)</td>
</tr>
<tr>
<td># of airways</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Aircraft information</th>
</tr>
</thead>
<tbody>
<tr>
<td># of aircraft</td>
</tr>
<tr>
<td>Minimum speed (TAS)</td>
</tr>
<tr>
<td>Maximum speed (TAS)</td>
</tr>
</tbody>
</table>

To construct a realistic scenario, we deal with a routing and a scheduling problem of 23 medium type of aircraft in the ICAO WTC. More details are given in the Table II. There are 23 aircraft (\( \alpha_1 \sim \alpha_{23} \)) on the airspace graph as shown in Fig. 5. Each aircraft should arrive London Heathrow Airport via one of the holding stack.
Table III and Table IV show the results, also graphically depicted in Fig. 6 ~ Fig. 9. Table III shows the arrival sequence, the arrival time at the airport, starting and the holding stacks. The route from the start waypoint to the end waypoint of each flight and its separation-compliant speed profile are shown in Table IV, which would help ATCOs’ decision by reducing the gap between each stages.

For quantitative comparison relying on the assumptions we made, actual flight data “21:00~21:20 Thursday 14 September 2017”, is used, as shown in Table V. In this data, 15 flights landed in 20 minutes, on the other hand, in the proposed algorithm, 23 flights could land in 23 minutes. Namely, the proposed algorithm could improve the runway throughput by 23% compared to the actual data at London Heathrow Airport, and might become as a new DST that enables a reduction of the ATCOs’ workload. The computation time for calculating the case study is 10.65 seconds on a Windows 10 OS 3.4 GHz desktop computer with 16 GB RAM.

<table>
<thead>
<tr>
<th>Flight ID</th>
<th>Arr. Seq.</th>
<th>Start WP</th>
<th>Merging WP</th>
<th>Arr. Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>α20</td>
<td>1</td>
<td>BNN</td>
<td>BNN</td>
<td>226</td>
</tr>
<tr>
<td>α23</td>
<td>2</td>
<td>DORKI</td>
<td>OCK</td>
<td>286</td>
</tr>
<tr>
<td>α22</td>
<td>3</td>
<td>LAM</td>
<td>LAM</td>
<td>363</td>
</tr>
<tr>
<td>α21</td>
<td>4</td>
<td>WOD</td>
<td>OCK</td>
<td>423</td>
</tr>
<tr>
<td>α19</td>
<td>5</td>
<td>WCO</td>
<td>BNN</td>
<td>483</td>
</tr>
<tr>
<td>α18</td>
<td>6</td>
<td>NIGHT</td>
<td>OCK</td>
<td>543</td>
</tr>
<tr>
<td>α13</td>
<td>7</td>
<td>GWC</td>
<td>OCK</td>
<td>603</td>
</tr>
<tr>
<td>α16</td>
<td>8</td>
<td>TIGER</td>
<td>BIG</td>
<td>663</td>
</tr>
<tr>
<td>α11</td>
<td>9</td>
<td>BRASO</td>
<td>LAM</td>
<td>723</td>
</tr>
<tr>
<td>α7</td>
<td>10</td>
<td>KENET</td>
<td>OCK</td>
<td>783</td>
</tr>
<tr>
<td>α5</td>
<td>11</td>
<td>DTY</td>
<td>BNN</td>
<td>843</td>
</tr>
<tr>
<td>α12</td>
<td>12</td>
<td>DET</td>
<td>BIG</td>
<td>903</td>
</tr>
<tr>
<td>α14</td>
<td>13</td>
<td>BEGTO</td>
<td>OCK</td>
<td>963</td>
</tr>
<tr>
<td>α15</td>
<td>14</td>
<td>LYD</td>
<td>BIG</td>
<td>1023</td>
</tr>
<tr>
<td>α6</td>
<td>15</td>
<td>CLN</td>
<td>LAM</td>
<td>1083</td>
</tr>
<tr>
<td>α19</td>
<td>16</td>
<td>ROTNO</td>
<td>BIG</td>
<td>1143</td>
</tr>
<tr>
<td>α3</td>
<td>17</td>
<td>HON</td>
<td>BNN</td>
<td>1203</td>
</tr>
<tr>
<td>α4</td>
<td>18</td>
<td>DVR</td>
<td>BIG</td>
<td>1263</td>
</tr>
<tr>
<td>α2</td>
<td>19</td>
<td>LOGAN</td>
<td>LAM</td>
<td>1323</td>
</tr>
<tr>
<td>α1</td>
<td>20</td>
<td>ALESO</td>
<td>BIG</td>
<td>1383</td>
</tr>
<tr>
<td>α17</td>
<td>21</td>
<td>KOPUL</td>
<td>BIG</td>
<td>1443</td>
</tr>
<tr>
<td>α9</td>
<td>22</td>
<td>DOMUT</td>
<td>OCK</td>
<td>1503</td>
</tr>
<tr>
<td>α8</td>
<td>23</td>
<td>BILNI</td>
<td>OCK</td>
<td>1563</td>
</tr>
</tbody>
</table>

V. DISCUSSION

To deal with routing and scheduling problems, which are one of the most challenging problems in a TMA, many researches have been conducted. However, these approaches find a solution in a limited solution search space because these approaches solve the routing and scheduling problem separately or sequentially. At the later stages (modifying schedule stage or freezing stage), therefore, optimality of the solution might be degraded. In contrast, our approach finds separation-compliant flight and schedules simultaneously. Namely, we could find solutions in a larger solution search space compared with the sequential approaches. The following additional studies in terms of the proposed algorithm are recommended to develop a full picture of a new advanced DST.

- **A better sequencing algorithm**: More studies are needed to develop better algorithms for determining the sequence of arrival aircraft than the FCFS algorithm to improve optimality. However, other sequencing algorithms with the single flight routing and scheduling algorithm we proposed might require high computational cost if the algorithms require more computational cost than the FCFS algorithm.
TABLE IV
SIMULATION RESULTS: FLIGHT PLANS (ROUTE AND SCHEDULE OF 23 FLIGHT)

<table>
<thead>
<tr>
<th>Flight ID</th>
<th>Arrival Sequence</th>
<th>Waypoint sequence (arrival time at each waypoint [sec]) speed profile [km/h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>α1</td>
<td>18th</td>
<td>ALESO(0) - ROTNO(345) - ETVAX(483) - TIGER(604) - BIG(936) - LON(1263) 250 - 250 - 250 - 238 - 224</td>
</tr>
<tr>
<td>α2</td>
<td>17th</td>
<td>LOGAN(0) - TRIPO(285) - SABER(372) - BRASO(553) - LAM(840) - LON(1203) 250 - 207 - 198 - 250 - 250</td>
</tr>
<tr>
<td>α3</td>
<td>15th</td>
<td>HON(0) - TOBID(160) - SOPIT(454) - WCO(595) - BNN(841) - LON(1083) 250 - 250 - 212 - 250 - 233</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>α21</td>
<td>4th</td>
<td>WOD(0) - OCK(266) - LON(423) 250 - 228</td>
</tr>
<tr>
<td>α22</td>
<td>3rd</td>
<td>LAM(0) - LON(363) 250</td>
</tr>
<tr>
<td>α23</td>
<td>2nd</td>
<td>DORKI(0) - OCK(110) - LON(286) 231 - 204</td>
</tr>
</tbody>
</table>

Figure. 9: Case study results of 23 flights in the LTMA for landing to London Heathrow: positions of aircraft at t = 936 sec

TABLE V
ACTUAL FLIGHT DATA FOR LANDING AT LONDON HEATHROW AIRPORT (21:00 ~ 21:20, THURSDAY 14 SEPTEMBER 2017)

<table>
<thead>
<tr>
<th>Flight ID</th>
<th>Origin</th>
<th>Destination</th>
<th>Aircraft name (ICAO WTC)</th>
<th>Arr. Time (h:m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>211116707</td>
<td>EGNT</td>
<td>EGLL</td>
<td>A319 (M)</td>
<td>21:00</td>
</tr>
<tr>
<td>211116002</td>
<td>EDDF</td>
<td>EGLL</td>
<td>A320 (M)</td>
<td>21:01</td>
</tr>
<tr>
<td>211114496</td>
<td>LEBL</td>
<td>EGLL</td>
<td>A320 (M)</td>
<td>21:02</td>
</tr>
<tr>
<td>211115949</td>
<td>EHAM</td>
<td>EGLL</td>
<td>A321 (M)</td>
<td>21:03</td>
</tr>
<tr>
<td>211114186</td>
<td>LPPT</td>
<td>EGLL</td>
<td>A320 (M)</td>
<td>21:05</td>
</tr>
<tr>
<td>211116372</td>
<td>EGPD</td>
<td>EGLL</td>
<td>A321 (M)</td>
<td>21:06</td>
</tr>
<tr>
<td>211116495</td>
<td>EGAC</td>
<td>EGLL</td>
<td>A319 (M)</td>
<td>21:09</td>
</tr>
<tr>
<td>211115728</td>
<td>LFML</td>
<td>EGLL</td>
<td>A320 (M)</td>
<td>21:10</td>
</tr>
<tr>
<td>211116249</td>
<td>EIDW</td>
<td>EGLL</td>
<td>A319 (M)</td>
<td>21:11</td>
</tr>
<tr>
<td>211114766</td>
<td>LIRF</td>
<td>EGLL</td>
<td>A321 (M)</td>
<td>21:12</td>
</tr>
<tr>
<td>211114520</td>
<td>EPWA</td>
<td>EGLL</td>
<td>B738 (M)</td>
<td>21:13</td>
</tr>
<tr>
<td>211114740</td>
<td>LEPA</td>
<td>EGLL</td>
<td>A321 (M)</td>
<td>21:15</td>
</tr>
<tr>
<td>211116925</td>
<td>EGFF</td>
<td>EGLL</td>
<td>A321 (M)</td>
<td>21:16</td>
</tr>
<tr>
<td>211116568</td>
<td>LFSB</td>
<td>EGLL</td>
<td>A319 (M)</td>
<td>21:17</td>
</tr>
<tr>
<td>211109711</td>
<td>OLBA</td>
<td>EGLL</td>
<td>A320 (M)</td>
<td>21:18</td>
</tr>
</tbody>
</table>

- Considering multiple airports: For more realistic scenarios in the TMAs with more than one airports such as the LTMA, an extended study should be conducted. In a multiple airport region, we should consider crossing points as well as merging points.
- Considering departures: Although the departure problem is usually solved separately from the landing problem, additional research is needed for simultaneously take-off and landing procedures at the same runway.
- Considering various types of aircraft: Since aircraft have different speed limitations and time-based separation requirements, it is necessary to consider all possible categories of aircraft types.
- Considering new aircraft entering the TMA: In a real system, the traffic flow entering the TMA is continuous. In order to take this into account in all possible phases to support ATCOs, the algorithm should be modified to allow for new aircraft entering the TMA.
- Considering detailed flight from the holding stacks: The considered assumptions should be reviewed to cope with more detailed description of approaching procedures or separately tackled in order to make the algorithm more practical.

VI. CONCLUSION

A new routing and scheduling approach for multiple flights has been presented that ensure minimum separation amongst inbound traffic. The approach seeks the optimal route and its schedule of each aircraft to maximise the runway throughput, given an airspace structure. The resultant outputs provide separation-compliant and speed-limitations-compliant routes and speed profiles. The main advantages of the proposed approach are three: solving the problem in a larger search space compared with the sequential approaches; efficient computational time; and providing separation-compliant segments and schedules. These advantages are relevant for the development of an new advance DST that helps ATCOs to manage traffic in a more efficient manner.
ACKNOWLEDGMENT

Authors would like to thanks Prof. Francisco Javier Sáez Nieto, Christian Verdonk Gallego and Manuel Ángel Amaro Carmona in Air Traffic Management Laboratory for the fruitful discussions.

REFERENCES


